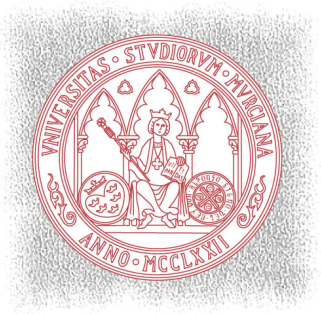


UNIVERSIDAD DE MURCIA
FACULTAD DE INFORMATICA
DEPARTAMENTO DE INFORMATICA Y SISTEMAS

TESIS DOCTORAL

PROCESAMIENTO DE CARAS HUMANAS MEDIANTE INTEGRALES PROYECTIVAS

PRESENTADA POR
GINÉS GARCÍA MATEOS

PARA LA OBTENCIÓN DEL GRADO DE
DOCTOR EN INFORMÁTICA

Director:
Alberto Ruiz García

MURCIA, 2007

Resumen

Uno de los dominios de investigación más activos en la última década, dentro de la disciplina de la visión por computador, es el que se ha dado en llamar el área "mirando a la gente". De forma genérica, se enmarcan en ella todos los sistemas de procesamiento de imágenes, reconocimiento de patrones y percepción artificial, cuyo objeto de estudio son los seres humanos. El análisis de caras constituye una de sus ramas más importantes, entre cuyas aplicaciones se encuentran: el reconocimiento automático de personas; el desarrollo de nuevos métodos de interacción hombre/máquina; la codificación y etiquetado de vídeo para videoconferencia o para sistemas de indexación multimedia; la seguridad, videovigilancia y control de accesos; los sistemas de captura de información gestual y ayuda a minusválidos, etc.

Esta tesis aborda los grandes problemas del procesamiento visual de caras humanas desde el punto de vista de las integrales proyectivas. Intuitivamente, una integral proyectiva –o, simplemente, una proyección– no es más que la media de los valores de gris de una imagen a lo largo de las filas o columnas de píxeles. Si bien las proyecciones constituyen una de las técnicas clásicas del análisis de imágenes, su utilización en el contexto que nos ocupa ha estado marcada por el diseño de métodos heurísticos y *ad hoc*. Nosotros planteamos la necesidad de sustentar su uso en mecanismos más formalizados, como los modelos de proyección y el alineamiento entre señales unidimensionales.

Estas dos cuestiones son estudiadas en detalle, y se proponen varias formas posibles de modelar las proyecciones y un algoritmo eficiente para el alineamiento de una señal respecto de un modelo. Apoyándonos en ambos elementos, se han diseñado métodos para la detección de caras en imágenes estáticas, la localización de componentes faciales, el seguimiento de los rostros en secuencias de vídeo, el reconocimiento facial de personas, el análisis de expresiones faciales, y la estimación de la posición y orientación 3D de la cara.

Los extensos experimentos llevados a cabo demuestran las ventajas de utilizar proyecciones frente a otros tipos de mecanismos: mayor capacidad de generalización; alta inmunidad frente al ruido; e invarianza frente a factores individuales y expresiones faciales. Los algoritmos propuestos para los diferentes problemas alcanzan siempre resultados iguales o superiores a los de otros métodos alternativos, comparables con los que constituyen el estado del arte, pero con una mejora muy significativa de la eficiencia computacional.

Abstract

In the last decade, one of the most active research areas in computer vision has been devoted to the so-called "looking at people" domain. In general, it includes all the techniques in image processing, pattern recognition, and artificial perception, which deal with human beings. Face analysis constitutes a primary field in this area, whose applications include: facial biometric recognition; perceptually driven man-machine interaction; video coding and labeling for teleconferencing and multimedia indexing; video-surveillance, access control and security; gesture recognition and aid for the disabled, etc.

This thesis tackles the main problems in face processing from the point of view of integral projections. Intuitively, an integral projection –or, in short, a projection– is the average of gray levels of an image along rows or columns of pixels. Whilst projections are a classical and well-known technique in image analysis, in the face domain they have been used through heuristic and *ad hoc* methods. We discuss the necessity to formalize their application, by means of projection models and alignment processes.

Both aspects have been thoroughly studied. We propose several techniques for modeling projections and a fast alignment algorithm. Using them, we have designed methods to solve the problems of face detection on still images, facial features location, face tracking in video sequences, person recognition, facial expression analysis, and 3D pose estimation.

Our extensive experiments have proved that using projections has a number of advantages with respect to other techniques: improved generalization capability, immunity to white noise, and robustness against facial expressions and individual factors. The proposed algorithms for the different problems achieve similar or better results than other available and more complex methods, and are comparable to most state-of-the-art systems, but with a considerable reduction on the computational cost.

Para Rocío,
cariño infinito,
faro de mis noches.

"Son tus labios dos senderos
Para que un Dios los andara.
Son tus dientes jazmineros,
Y tus ojos dos luceros
En el cielo de tu cara."
José Traver "Repuntín".

Agradecimientos

Mi primer agradecimiento, cómo no, es para mi director de tesis, Alberto Ruiz García, que allá por la primavera de 1997 supo introducirme en el apasionante mundo de la visión artificial y el reconocimiento de patrones. Sin sus enseñanzas, su apoyo y su absoluta confianza en mi capacidad investigadora, esta tesis nunca habría sido posible.

Tampoco habría sido posible sin el apoyo de mi familia, y especialmente de mis padres, que esperan con ilusión el momento de la investidura con el birrete laureado. A veces la ayuda del más pequeño es la más grande de las ayudas.

Gracias también a los compañeros del grupo de "Percepción Artificial y Reconocimiento de Patrones" –los que lo son y los que lo fueron en su día–, y al resto de colegas investigadores de la Facultad de Informática de la Universidad de Murcia. Un recuerdo muy especial para los que aún luchan por *sacar* su tesis, sin dejarse vencer por el desánimo, el abandono o el conformismo. ¡Ánimo, la tesis está en vosotros!

Y, finalmente, gracias a ti querido lector. Espero que disfrutes leyendo esta humilde tesis siquiera una pequeña parte de lo que yo he disfrutado escribiéndola.

Ginés G.M.
Murcia, diciembre de 2006

Índice general

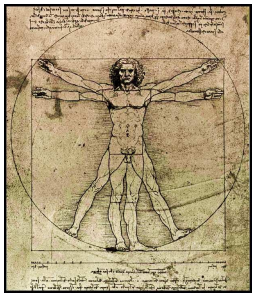
Índice de figuras

Índice de tablas

Índice de algoritmos

.

CAPÍTULO 1



"El hombre de Vitruvio",
Leonardo da Vinci, c. 1490

Procesamiento de Caras Humanas

> *"Pinta las caras de manera que no todas tengan la misma expresión, como hacen muchos pintores, sino dales diferentes expresiones, de acuerdo con su edad, complexión, y su buen o mal carácter. [...] Toma fragmentos de muchas caras bellas, de las cuales la belleza es establecida por la reputación general más que por tu propio juicio, puesto que puedes engañarte seleccionando rostros parecidos al tuyo mismo."*
>
> LEONARDO DA VINCI

En enero de 2000, Alex Pentland, director del influyente "Media Lab" del Instituto Tecnológico de Massachusetts, publica un artículo en el que saluda el nacimiento de una nueva y emergente rama de la visión artificial, a la que califica como el área "mirando a la gente" (en inglés, *looking-at-people*) [136]. El documento hace un repaso de la corta pero intensa trayectoria del área y –con cierta perspectiva de futuro– anticipa las nuevas tecnologías, aplicaciones y los problemas que plantea. En sí mismo, el documento no marca un hito o un punto de partida, pero viene a constatar un hecho que hoy día resulta innegable: la disciplina de la visión artificial, que hasta unos años atrás se había dedicado a *mirar* rectas, planos, curvas, elipses, polígonos, caracteres, símbolos, mundos poliédricos y entornos estructurados, empieza ahora a dirigir su mirada hacia las personas. Como en la magistral interpretación de Leonardo da Vinci de la teoría de las proporciones de Marco Vitruvio, el problema vuelve a girar en torno al ser humano.

En la presente tesis doctoral estudiamos cómo las integrales proyectivas, una de las técnicas clásicas en el procesamiento de imágenes, pueden ayudar de manera decisiva en los diferentes problemas planteados en el dominio de las caras humanas. De esta forma, el trabajo desarrollado presenta una doble motivación: establecer un marco adecuado para el manejo de proyecciones; y aplicarlo en la resolución de algunos de los grandes problemas con caras.

Este capítulo introductorio presenta brevemente el ámbito del procesamiento facial, las técnicas y acercamientos existentes, así como los objetivos y la metodología de la tesis. En primer lugar, la sección 1.1 echa un rápido vistazo a los diversos problemas a los que nos

enfrentamos, y describe los principales acercamientos y las aplicaciones que han sido propuestos. Sólo pretendemos dar aquí una visión global de las tecnologías disponibles; en los capítulos del 3 al 6 se pueden encontrar revisiones más exhaustivas del estado del arte para cada problema concreto. En la sección 1.2 se exponen los motivos por los que las integrales proyectivas se perfilan como una técnica prometedora en el área específica de las caras. Finalmente, se presentan los objetivos de la tesis en la sección 1.3, y se describe la metodología de trabajo seguida.

1.1. El ámbito del procesamiento automático de caras

Es sorprendente la cantidad de información que los humanos podemos extraer mirando a simple vista una imagen donde aparecen rostros de otras personas; y más aún lo que podemos llegar a deducir fijándonos en todos los aspectos y detalles de la imagen. Para demostrarlo, pedimos al lector que realice el pequeño experimento expuesto en la figura 1.1, midiendo el tiempo que tarda entre la lectura de cada pregunta y la obtención de la respuesta.

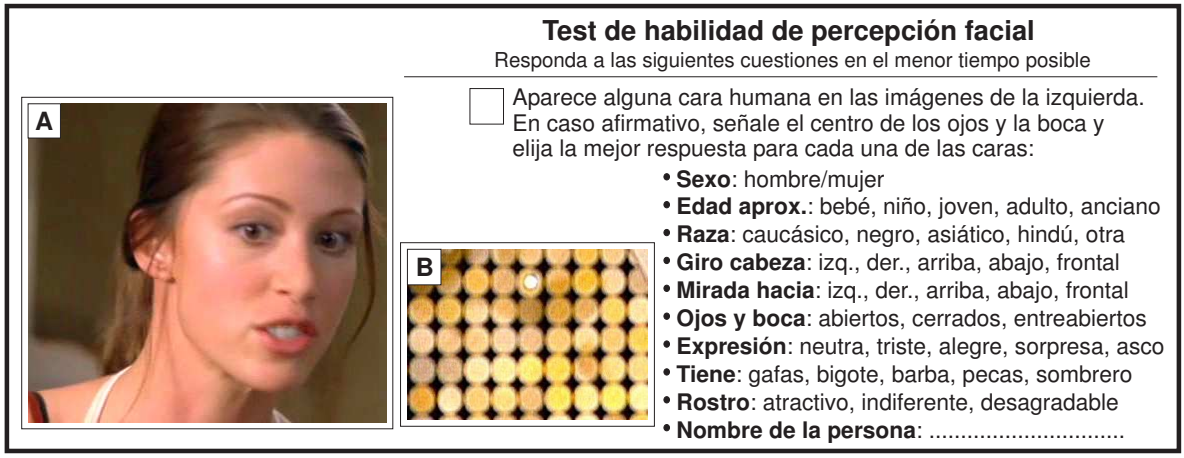


Figura 1.1: *Un pequeño experimento de percepción humana de caras. Observe las imágenes de la izquierda y responda las cuestiones lo más rápidamente posible. En el texto se describen algunos resultados del procesamiento automático sobre estas mismas imágenes. (Continúa en la figura 1.2.)*

Con toda probabilidad, el lector habrá sido incapaz de diferenciar entre el momento en que acaba de leer cada cuestión y el instante en que sabe la respuesta. Nos resulta inmediato: contar el número de individuos presentes y localizarlos por separado; decir, para cada uno, si se trata de un hombre o de una mujer, su raza y, posiblemente, su grupo étnico específico; diferenciar si es una persona anciana, adulta, joven o niño, y afinar en el margen de edades más probable; contestar preguntas de tipo verdadero/falso como: usa gafas, lleva sombrero, tiene barba o bigote, es calvo, rubio o moreno, tiene pecas, lleva tatuajes, etc.; estimar la orientación del rostro respecto de la cámara, decir hacia dónde está mirando y dónde están las fuentes de iluminación. Si la persona es conocida, el lector habrá sido capaz de asignarle una identidad al rostro con total independencia de su expresión, iluminación, y orientación. Y si

no lo es, habrá establecido relaciones de similitud con otras personas conocidas.

En cuanto a las expresiones faciales, nuestra habilidad no se queda atrás. Se ha descrito más de una cuarentena de "unidades de activación" distinguibles por los humanos [48], asociadas a la contracción o relajación de diversos músculos faciales. La lista incluye, por ejemplo, apretar los labios, subir la barbilla, poner boca de embudo, pestañear, y subir o bajar las cejas. El resultado del análisis de la expresión nos permite inferir información sobre el estado de ánimo de la persona, y es uno de los mecanismos básicos de la comunicación no verbal entre humanos. Podemos decir si está feliz, triste, indiferente, si está actuando o la expresión es natural. Y todo esto por no hablar de las sensaciones o impresiones subjetivas que una cara provoca en el observador: simpatía, antipatía, enfado, lástima, atracción, etc.

Resultados del análisis automático de caras

Como sucede en tantos otros ámbitos de la visión artificial –y también de la inteligencia artificial–, cuando tratamos de emular en un ordenador las capacidades instintivas y naturales del ser humano nos topamos con enormes dificultades. Sin ánimo de ser exhaustivos, estos son algunos de los resultados conseguidos por varias técnicas que constituyen el estado del arte del procesamiento facial sobre las imágenes de la figura 1.1:

- Uno de los detectores de caras más populares y exitosos –basado en filtros de Haar, cascadas de clasificadores y el algoritmo AdaBoost [188]– es incapaz de encontrar ninguna cara en la imagen A de la figura 1.1, incluso ajustando el método a un modo tan sensible que "encuentra" 3 caras en la imagen B. La razón parece encontrarse en que el rostro está "demasiado próximo" al lado derecho de la imagen.

- Otro detector clásico –basado en búsqueda exhaustiva multiescala, ecualización de histogramas y redes neuronales [152]– encuentra la cara de la imagen A, pero tarda 3 segundos (en un Pentium IV a 2,6GHz) para llegar a la conclusión de que en la imagen B aparecen 5 caras.

- El potente mecanismo de localización de componentes faciales basado en modelos de apariencia activa (AAM) –que se apoya en procesos iterativos de ajuste del modelo, métodos de gradiente descendente y descomposiciones mediante PCA [32]–, acaba situando la boca de la imagen A en la posición de la mejilla, el ojo derecho en el tabique nasal, y la ceja izquierda en el ojo del mismo lado. Aunque el proceso es inicializado exactamente en la posición ideal, el método no es capaz de generalizar a una cara no usada en el entrenamiento del modelo.

- El proceso de seguimiento por color –en este caso, el célebre algoritmo CamShift [16]– fracasa completamente sobre la secuencia a la que pertenece la imagen A de la figura 1.1, debido a que el cuello, el pelo y el fondo de la escena tienen un tono de color muy parecido al de la piel.

- Una popular herramienta *on-line* de reconocimiento de personas famosas[1] –que usa uno de los tres métodos mejor clasificados en las evaluaciones públicas del consorcio FRVT 2002 [139]– identifica incorrectamente la cara de la figura 1.1 A como la actriz Kate Beckinsale. Sin embargo, su base de imágenes contiene nada menos que 7 muestras de la persona correcta. La identidad real tampoco se encuentra entre las 10 más probables devueltas por el sistema.

Muchos de estos fallos son debidos a los grandes *caballos de batalla* de la percepción artificial y el reconocimiento de patrones: el sobreajuste a los datos de entrenamiento, la escasa capacidad de generalización de los métodos de aprendizaje usados, la introducción de restricciones no justificables, la utilización de invariantes que no son tan invariantes, etc.

En cualquier caso, esperamos no haber dado una visión excesivamente negativa de la investigación en procesamiento facial que, por otro lado, está salpicada de grandes hitos: la construcción del primer sistema *completo* de reconocimiento facial en 1973 [93]; la publicación gratuita y en código abierto de buenos sistemas de detección y seguimiento [35]; la capacidad de reconocimiento de gemelos idénticos mediante técnicas 3D en 2003 [17]; los sistemas de ayuda a minusválidos; la incorporación de tecnologías de seguimiento facial en cámaras fotográficas –las primeras están previstas para 2007–; y las nuevas aplicaciones de generación de *avatares* animados[2], actores virtuales, y alucinación facial, entre otras muchas.

Además, también la percepción humana de las caras se encuentra con grandes dificultades bajo condiciones a las que no está habituada. Si el lector, después de completar el experimento de la figura 1.1, ha acabado excesivamente confiado de su gran habilidad de percepción facial, le pedimos que repita el mismo experimento sobre las imágenes de la figura 1.2.

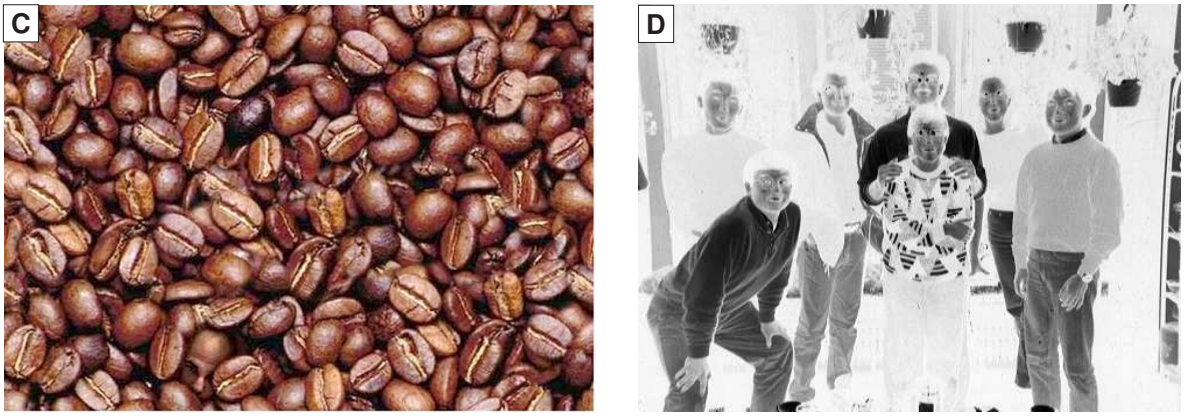


Figura 1.2: *Un pequeño experimento de percepción humana de caras (continuación). Responder lo más rápidamente posible las cuestiones de la figura 1.1. La imagen C es debida a Furitsu (`http://worth1000.com`). La D es una inversión de la imagen "rehg-thanksgiving-1994.gif" de la base de caras CMU/MIT [152].*

Seguramente habrá tardado cerca de medio minuto en encontrar el rostro de la imagen C. Y dudamos que haya sido capaz de deducir que en la D aparecen 3 orientales, 3 caucásicos y

[1] Ver la página: `http://www.myheritage.com`
[2] Entiéndase por "avatar" una representación iconográfica del rostro del usuario.

un hindú, de ellos 5 hombres y 2 mujeres. Se puede comprobar el resultado en la figura 1.3, donde el lector verá que el inconveniente no se encuentra en la resolución de la imagen.

A pesar de todos estos obstáculos, el beneficio que supondría disponer de técnicas automáticas capaces analizar y trabajar con rostros humanos motiva sobradamente la investigación en este ámbito. Es más, el dominio de las caras es un interesante banco de trabajo para la investigación en visión artificial. Muchos avances en los problemas de caras pueden ser extendidos –y de hecho ya lo han sido en el pasado [183, 188]– a otros dominios.

1.1.1. Problemas con caras humanas

Son muchos los problemas específicos que surgen en el procesamiento facial automático. Existe una relación muy estrecha entre todos ellos, pero la mayoría de los investigadores coinciden en clasificarlos según la cuestión que tratan de resolver. Podemos identificar los siguientes grandes problemas:

- **Detección de caras.** La cuestión que resuelve es: cuántas caras aparecen en una imagen y cuál es la extensión espacial de cada una de ellas. Su resolución resulta un requisito preliminar para la gran mayoría de los restantes problemas: antes de poder analizar un rostro, primero hay que saber si existe y dónde está. Algunos autores denominan **localización de caras** al problema de detección suponiendo que existe una –y sólo una– persona en la imagen, y que su rostro está más o menos centrado. En la figura 1.3 se pueden observar algunos ejemplos típicos de posibles entradas para un detector de caras.

Figura 1.3: *Ejemplos de entradas para la detección facial, tomadas de la base de caras CMU/MIT [152], la base propia (UMU), y otras fuentes. De arriba abajo, de izquierda a derecha: "rehg-thanksgiving-1994.gif" (CMU/MIT), "bwolen.gif" (CMU/MIT), "1107.avi.jpg" (UMU), "audrybt1.gif" (CMU/MIT), "mars1976.jpg" (otras), "aeon1a.gif" (CMU/MIT).*

  Varios de los ejemplos, como las imágenes "aeon1a.gif" y "bwolen.gif", son muestras de cómo no existe un consenso unánime en cuanto a qué se considera o no como cara humana (por no hablar de los *ríos de tinta* que han corrido en relación a la detección facial en la escena de la imagen "mars1976.jpg").

- **Localización de componentes faciales.** Una vez con la cara detectada, el objetivo de

la localización de componentes es determinar las posiciones exactas que ocupan los ojos, la boca, y otros elementos que puedan resultar de interés. A diferencia de la detección –donde lo importante es una buena discriminación cara/no-cara–, aquí el objetivo esencial es la precisión de los resultados, que puede verse influida por la resolución de entrada. Por ejemplo, en la imagen "bwolen.gif" de la figura 1.3 se vislumbra un rostro, pero ¿dónde están exactamente los ojos y la boca?

El problema ha sido también denominado *extracción* o *detección de componentes faciales*; la localización del contorno del rostro se llama a veces la *segmentación de la cara*.

- **Estimación de *pose*.** En este caso se trata de resolver los 6 parámetros de posición y orientación 3D –lo que llamamos la *pose*– de la cabeza del individuo en relación a la cámara. Obviamente, es un problema de naturaleza tridimensional. No obstante, ello no implica usar explícitamente modelos 3D. Un problema relacionado es la **estimación del punto de mira** (en inglés, *gaze*) que se centra en el análisis de los ojos.

- **Seguimiento de caras en vídeo.** El propósito del seguimiento es encontrar las variaciones de posición, forma y/o orientación de los rostros a lo largo de una secuencia de vídeo. Algunos autores entienden que el seguimiento debe ser un problema 3D, mientras que otros trabajan en 2D con la suposición de que el individuo mira frontalmente a la cámara, admitiendo pequeños márgenes de giro. La naturaleza *no rígida* e *impredecible* del rostro hace que el problema difiera sustancialmente de otros tipos de seguimiento estudiados en visión artificial. La figura 1.4 contiene varios extractos de una de las secuencias usadas en los experimentos.

Figura 1.4: *Extractos de una secuencia para seguimiento de caras. El vídeo se denomina "ggm4.avi" y está disponible públicamente en:* `http://dis.um.es/profesores/ginesgm/fip`.

- **Reconocimiento de personas.** Existen varios subproblemas en el contexto del reconocimiento facial de personas. Todos ellos parten de una base de imágenes de individuos

conocidos, también llamada *galería*, y una imagen nueva a procesar, conocida como la *prueba*. Los tres escenarios estándar son:

- **Identificación en conjunto cerrado.** Devuelve la identidad más probable para la prueba de entre los individuos de la galería. La pregunta sería, ¿quién es?
- **Verificación.** Dada una prueba y una identidad aducida, se debe decidir si corresponde a la misma o no. La cuestión ahora sería, ¿es quien dice ser?
- **Identificación en conjunto abierto.** En este caso se trata de resolver dos cuestiones: (1) ¿es conocida la persona (es decir, está en la galería)?; (2) en caso afirmativo, ¿quién es, cuál es su identidad?

La capacidad humana de reconocimiento de caras resulta asombrosa incluso en condiciones complejas de iluminación, pose, expresión y elementos faciales, bajo cuya combinación fallan casi todos –por no decir todos– los acercamientos existentes. Sin embargo, también la percepción humana puede ser confundida con simples alteraciones de las imágenes, por ejemplo, la rotación de los rostros que aparece en la figura 1.5.

Figura 1.5: *Otro pequeño experimento de reconocimiento facial humano. Se trata de encontrar las dos imágenes que corresponden a la misma persona. Imágenes tomadas de la base de caras ORL [159].*

- **Análisis de la expresión facial.** El análisis de expresiones ha sido formulado desde dos perspectivas diferentes. Para algunos autores, el problema consiste en clasificar la expresión del sujeto en un número discreto y predefinido de clases, mientras que otros buscan una estimación graduada del estado de activación de los músculos faciales.

  En ambos casos, sería conveniente diferenciar entre el *análisis del gesto* (boca abierta, cerrada, entreabierta, cejas subidas, etc.) y la *interpretación emocional* (triste, contento, enfadado, etc.). El segundo es más complejo, ya que puede requerir un conocimiento e interpretación del contexto y las consideraciones sociológicas.

- **Extracción de información.** Podemos clasificar dentro de este grupo otros muchos problemas con caras que han sido abordados por diferentes autores, y que tratan de responder preguntas sobre propiedades concretas de los rostros ya detectados. Por un lado, se pueden encontrar cuestiones del tipo: lleva gafas, tiene barba, hay oclusión del rostro, etc. El resultado puede estar orientado, por ejemplo, a mejorar las imágenes para una etapa posterior de reconocimiento biométrico [9].

Por otro lado, se ha trabajado también sobre preguntas relacionadas con *categorías demográficas*, como el sexo, la raza y la edad de los individuos. Se da por supuesto que existen características faciales comunes a cada grupo, y que se pueden aprender mediante entrenamiento. En la figura 1.6 mostramos algunos ejemplos de caras medias según categorías de raza y edad, en imágenes alineadas de la base FERET [52]. En término promedio, existe una diferencia entre grupos. No obstante, la diferencia de un individuo a su media suele ser mucho mayor que la de las caras medias entre sí.

Figura 1.6: *Caras medias de la base FERET según diversos factores demográficos. Parte superior, caras medias por raza. De izquierda a derecha: asiático (332 caras), hispano (110), caucásico (1227), hindú (104), negro (152). Parte inferior, caras medias por edad. De izquierda a derecha: 10-19 años (32 caras), 20-29 (887), 30-39 (551), 40-49 (332), 50 o más (204).*

En la figura 1.7 se pueden ver ejemplos de caras medias según el sexo, en este caso de la base propia. Hemos llevado a cabo un pequeño ensayo, consistente en etiquetar los rostros con un *grado subjetivo de belleza* entre 1 y 4; en muchos casos, el valor está relacionado más bien con la calidad y resolución de las imágenes. Las primeras columnas de la figura 1.7 muestran los resultados para cada uno de los cuatro grupos.

Otros problemas sobre caras humanas son la super-resolución y alucinación de rostros [4], la reconstrucción tridimensional, y la captura para animación facial. En cierto sentido, se pueden considerar como aplicaciones de los anteriores problemas.

1.1.2. Investigación y acercamientos al procesamiento de caras

El creciente interés de la comunidad investigadora por los temas relacionados con el análisis facial se puede constatar sobradamente en el elevado número de congresos internacionales

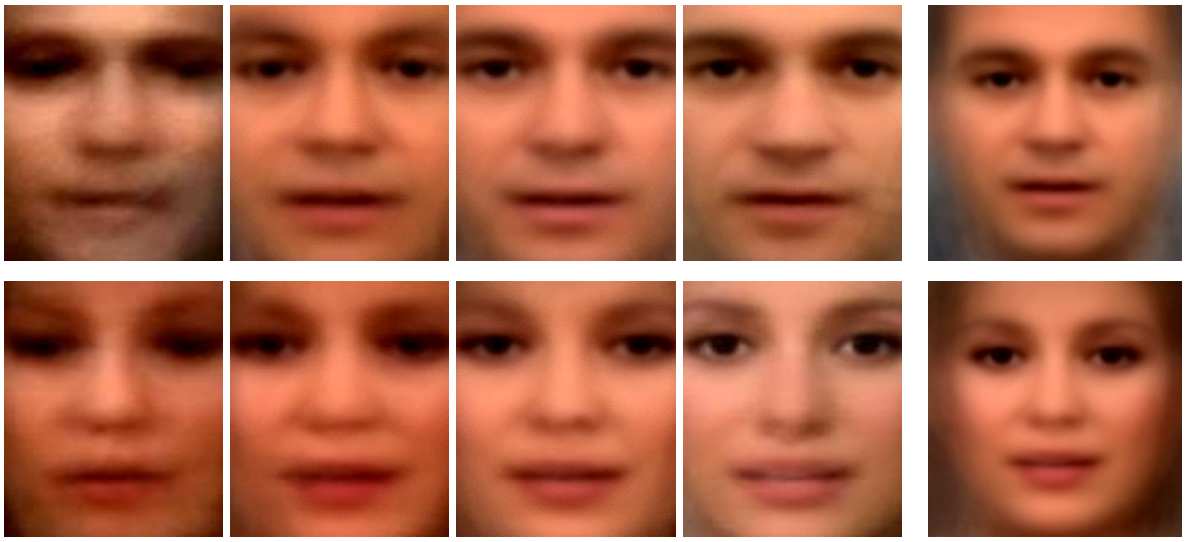


Figura 1.7: *Caras medias de la base UMU según el sexo. Parte superior, caras masculinas. Parte inferior, caras femeninas. Las primeras cuatro columnas son las medias de las 4 categorías de "belleza/resolución", en orden creciente: columna 1 (51 hombres / 53 mujeres), columna 2 (126 / 114), columna 3 (158 / 146), columna 4 (93 / 95). La última columna es la media de los grupos 2, 3 y 4.*

relacionados de forma más o menos directa con alguna de sus diversas ramas. En la tabla 1.1 aparece un listado de las conferencias más importantes, con la información disponible en el momento de la escritura de esta tesis.

Acrónimo	Nombre de la conferencia	Página web oficial
AVBPA 2005	Audio- and Video-based Biometric Person Authentication, Tarrytown, USA	http://biometrics.cse.msu.edu
FRGC 2005	IEEE Workshop on Face Recognition Grand Challenge Experiments, San Diego, USA	http://www.bee-biometrics.org/~frgc05/
FPiV 2005	2nd Workshop on Face Processing in Video, Victoria, Canadá	http://www.visioninterface.net/fpiv05
AMFG 2005	IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures, Pequín, China	http://mmlab.ie.cuhk.edu.hk/iccv05
ICBA 2006	International Conference on Biometric Authentication, Hong Kong	http://www4.comp.polyu.edu.hk/~icba/
FG 2006	7th IEEE Intl. Conference Automatic Face and Gesture Recognition, Southampton, UK	http://www.fg2006.ecs.soton.ac.uk/
ICB 2007	International Conference on Biometrics, Seúl, Corea	http://image.korea.ac.kr/ICB2007/
VidRec 2007	International Workshop on Video Processing and Recognition, Montreal, Canadá	http://computer-vision.org/~VideoRec07

Tabla 1.1: *Algunas de las principales conferencias relacionadas directamente con el procesamiento facial. Se indican las ediciones más recientes de cada una de las conferencias. Información extraída de:* `http://www.face-rec.org/conferences/`

Podemos poner otros ejemplos significativos. En la revisión bibliográfica de M.H. Yang y otros [204] de 2002, se identifican más de 150 acercamientos distintos a los problemas de detección y localización de caras humanas. Por su parte, Ahlberg y Dornaika definen el panorama en seguimiento de rostros como una "plétora" [108], y ya en 1998 Toyama había descrito 35

métodos diferentes [180]. El análisis de expresiones faciales tampoco es menos activo, con más de 80 trabajos revisados por Fasel y Luetin en 2003 [49]. Y, por supuesto, la investigación en biométricas faciales, para las cuales el mercado estimado para 2007 supera los 600 millones de dólares [31]; Zhao y otros [212], mencionan un centenar de grupos trabajando en el reconocimiento de personas, y listan 15 grandes compañías dedicadas al desarrollo y comercialización estos sistemas. Haciendo una búsqueda bibliográfica, el número de publicaciones de "detección de caras" y "reconocimiento de caras" iguala o supera a la de problemas clásicos como "navegación de robots" y "reconocimiento óptico de caracteres"[3].

Ciñéndonos al ámbito puramente académico, podemos señalar una serie de técnicas ubicuas, grandes tópicos y acercamientos comunes a los diferentes problemas del dominio facial. No profundizaremos aquí en exceso, ya que la revisión de métodos y trabajos para problemas concretos se puede encontrar en los sucesivos capítulos.

- **Enfoque holístico vs. basado en características.** La forma de procesar las imágenes de las caras conduce a estos dos grandes enfoques: los que tratan globalmente el rostro como un todo, llamados *holísticos* o *basados en apariencia*; y los que extraen información de las imágenes para luego trabajar con ella, denominados *basados en características*. En cierto sentido, es el reflejo de la disyuntiva local/global, que aparece recurrentemente en detección, localización, seguimiento, reconocimiento, etc.

  - Los métodos del primer grupo suelen confiar en la existencia de ejemplos suficientemente representativos de todas las apariencias faciales, y descansan en técnicas de clasificación potentes. Puesto que manejan directamente las imágenes, se suelen aplicar técnicas de proyección en subespacios como una forma de reducir la dimensionalidad del problema.
  - Los del segundo grupo tratan de apoyarse en propiedades invariantes y fáciles de calcular: color, bordes, textura, simetría, componentes faciales, movimiento, etc. Los procesos suelen ser diseñados *ad hoc*, sustituyendo así el costoso entrenamiento de los métodos basados en apariencia.

  La frontera entre ambos grupos no siempre está clara, y en muchas ocasiones se podría hablar de una tercera alternativa: los *métodos híbridos*. Por ejemplo, un algoritmo puede extraer los bordes para procesarlos después de forma holística; o un seguidor basado en apariencia puede realizar un análisis separado de ojos y boca.

- **Modelado y análisis del color de piel humana.** El color de la piel humana es una de las propiedades que más frecuentemente han sido utilizadas en todos los problemas del procesamiento facial. Es fácil de calcular, invariante frente a muchos factores, y tiene una alta capacidad de discriminación. Las dos grandes cuestiones relacionadas con el color

[3] En concreto, los datos son: "face detection/location" 10800 resultados; "face recognition" 12400; "robot navigation" 14900; "optical character recognition" 12000. Información obtenida de: `http://scholar.google.com`

son: (1) qué espacio de color resulta más adecuado para trabajar con la piel humana; y (2) cómo modelar un tono de color en ese espacio. En el caso del vídeo, surge otra cuestión: cómo actualizar el modelo para afrontar los cambios de iluminación. La figura 1.8 contiene un ejemplo donde se pueden observar los distintos rangos de valores que suelen tomar los canales R, G y B para el color de piel. Es interesante observar que el rojo es el que produce un mayor contraste de tonos en el resultado.

Figura 1.8: *Análisis de color de la piel humana. a) Imagen original en color, en el espacio RGB. b) Canal R. c) Canal G. d) Canal B. e) Histograma conjunto de los canales Cr y Cb del modelo YCrCb.*

Se han realizado muchas y variadas propuestas en relación al color, y diversos estudios comparativos. En [108, capítulo 6] se puede encontrar un resumen de los más interesantes; algunas de las principales conclusiones son: los tonos de piel suelen formar un grupo compacto en la mayoría de los espacios –como sucede en la figura 1.8e)–; los diferentes estudios comparativos no coinciden en cuanto a qué espacio resulta más adecuado; las diferencias entre razas afectan fundamentalmente a la intensidad, más que a la cromacidad; los modelos basados en histogramas suelen conseguir mejor rendimiento que los que usan gaussianas o modelos de mezcla; en la actualidad, ninguna técnica parece capaz de garantizar la constancia de color en todas las condiciones.

- **Autocaras y proyección en subespacios.** El *análisis de componentes principales* (PCA) es otra de las técnicas omnipresentes en los problemas de procesamiento facial, desde que Kirby y Sirovich demostraron por primera vez que las caras podían ser codificadas y reconstruidas de forma fiable usando subespacios lineales de muy reducida dimensionalidad [97]. De hecho, la reducción de dimensiones es uno de sus usos más frecuentes: dada una imagen de una cara, representarla de manera compacta y conservando la mayor parte de la información relevante. Los conceptos de PCA, descomposición en valores y vectores propios, transformada Karhumen-Loève, y transformada Hotelling son esencialmente equivalentes.

  Existen diversas formas de entender la descomposición PCA: como una rotación de los vectores de entrada (en nuestro caso, las imágenes de caras) usando una base *ortonormal*; como una proyección de los ejemplos en las direcciones de máxima varianza; y como una descomposición de una imagen mediante la suma de una base de imágenes (las

autocaras). En la figura 1.9 se representan gráficamente estas interpretaciones.

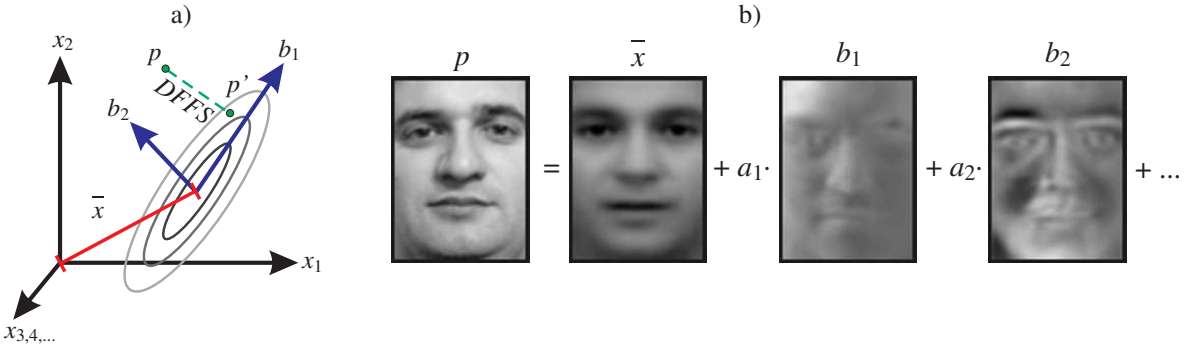


Figura 1.9: *Interpretación de la descomposición en autocaras mediante PCA. Las imágenes se pueden ver como puntos en un espacio de alta dimensionalidad, $x = (x_1, x_2, x_3, \ldots)$. a) Las caras forman un subespacio de mucha menor dimensionalidad. $\bar{x}$ es la cara media; b_1 y b_2 son las direcciones de máxima varianza (las autocaras); cualquier imagen p se puede proyectar en el subespacio de las caras en p'; DFFS es la distancia del punto al autoespacio (distance from feature space) o error de reconstrucción. b) Cualquier cara, p, se puede expresar como la suma de la cara media, $\bar{x}$, más una combinación lineal de las autocaras, $b_1, b_2, \ldots$, siendo $a_1, a_2, \ldots$ los coeficientes de la combinación lineal (la proyección de p en el autoespacio).*

Se han propuesto muchos usos, variaciones, extensiones y métodos alternativos, como la *descomposición en componentes independientes* (ICA), el *análisis de discriminantes lineales* (LDA), *PCA probabilístico*, los *analizadores de factor* (FA, en inglés *factor analyzer*) y otras técnicas de proyección no lineal, como *kernel PCA* y *kernel LDA*.

- **Filtros de convolución.** La utilización de filtros de convolución es común a otros muchos ámbitos de la visión artificial. Su importancia reside en la necesidad de conseguir robustez frente a factores como la iluminación global de la imagen, el contraste, el ruido, y la inclinación de los objetos. Muchas veces los filtros están orientados a obtener características de bajo nivel, con las que se aplica un procesamiento posterior. Entre los métodos más frecuentes tenemos los clásicos operadores de suavizado y detección de bordes, la transformada de Fourier y del coseno, los filtros de *wavelet* –en especial los de Haar y Gabor–, y los operadores de morfología matemática. También se pueden considerar dentro de este grupo las integrales proyectivas, que han sido aprovechadas previamente en tareas que incluyen desde la detección hasta el reconocimiento de caras.

- **Modelos deformables de forma y apariencia.** Muchos sistemas de detección, seguimiento y reconocimiento de caras se basan en la utilización de modelos. En general, un modelo almacena información *a priori* sobre cierta clase de objetos, codificada por el humano o bien obtenida mediante entrenamiento. Una de las ideas más extendidas es la definición de modelos que separan la forma y la apariencia. En analogía al contexto de generación gráfica, la *forma* sería la malla de puntos, y la *apariencia* sería la textura aplicada sobre la anterior. La combinación de ambas surge de la observación de que ninguna de las dos por separado es suficiente: la cara tiene una forma 3D compleja, y una textura

no uniforme. Se han utilizado modelos 2D y 3D. La tendencia de la investigación parece encaminarse al desarrollo de estos últimos.

Desde la perspectiva de los modelos, el procesamiento de caras sigue la filosofía conocida como *análisis a través de la síntesis*: los distintos problemas se pueden entender como un ajuste de la posición y los parámetros del modelo deformable, buscando un resultado que sea lo más parecido posible a la imagen actual.

- **Técnicas basadas en aprendizaje.** Como hemos mencionado ya, muchos trabajos se apoyan en la utilización de complejos mecanismos de aprendizaje y clasificación. En estos casos, los modelos o la extracción de características suelen tener una menor importancia, y la atención se centra en buscar la mejor forma de aplicar métodos como las máquinas de vectores de soporte (SVM), las redes neuronales, o el algoritmo AdaBoost, entre otros. En estos sistemas el aspecto central es validar la efectividad del clasificador. Las técnicas de *boosting* y *generación de muestras virtuales* contribuyen a obtener conjuntos representativos de caras y de no caras.

1.1.3. Aplicaciones del análisis de caras

Existe un vasto campo de aplicación de los sistemas de procesamiento facial, asociado a los diversos problemas ya descritos –normalmente, haciendo uso de más de un problema–. Estos son algunos de los principales usos propuestos:

- **Biométricas y seguridad.** La superioridad de las caras frente a otras biométricas ha sido reconocida en muchos estudios [80, 11], que les otorgan una excelente relación entre fiabilidad, coste e "intrusividad". El interés por este tipo de tecnologías afecta tanto al sector público como al privado. Entre algunos usos específicos que han sido descritos podemos encontrar [212]:
  - control de aduanas, inmigración, pasaportes;
  - documentos nacionales de identidad, seguridad social, permiso de conducir, empadronamiento, etc., con información biométrica;
  - sistemas de *login* biométrico, doméstico o empresarial;
  - seguridad en aplicaciones como acceso a Internet, a bases de datos, registros médicos, etc.;
  - terminales de punto de venta seguros;
  - entrada y control de acceso a edificios.

  En todos ellos existe un fundamento común: el reconocimiento facial. Pero el dominio de aplicación puede cambiar drásticamente los aspectos a considerar. Así, no es lo mismo el caso de la aduana que debe controlar millones de identidades, que un sistema doméstico que puede no llegar a la docena de usuarios.

- **Vídeo-vigilancia y monitorización.** Existen otras aplicaciones relacionadas con la administración judicial/policial y de seguridad, que situamos en este grupo. En relación con la categoría anterior, la particularidad es que los usuarios pueden no ser conscientes de que están siendo controlados; es más, puede que no quieran ser reconocidos. Esto supone un dificultad añadida. Entre los ejemplos de este tipo tenemos:

  - sistemas de vigilancia en aeropuertos, en ciudades o en edificios públicos;
  - búsqueda de personas desaparecidas;
  - seguimiento automático de sospechosos de robo en centros comerciales;
  - sistemas de análisis *post-evento*.

  Otro gran obstáculo es que muchas de estas aplicaciones trabajan con cámaras CCTV (circuito cerrado de TV), que suelen ofrecer una muy baja calidad de imagen. En cuanto a la monitorización, se suele referir al control de un individuo para el cuál existe un comportamiento esperado; por ejemplo, de pacientes en un hospital, o los sistemas de alerta de conductores.

- **Interfaces perceptuales y entretenimiento.** El procesamiento automático de los rostros permite añadir nuevos mecanismos de interacción hombre/máquina que, convenientemente diseñados, pueden resultar más intuitivos, sencillos y potentes que los sistemas tradicionales de teclado y ratón. Por ejemplo, tenemos:

  - interfaces perceptuales para la navegación en entornos virtuales;
  - interacción natural con robots;
  - videojuegos basados en la percepción del rostro;
  - sistemas de aprendizaje a distancia;
  - interpretación de las emociones del usuario;
  - sistemas de reconocimiento de voz que combinan sonido y lectura de labios;
  - control automático de la cámara, por ejemplo, para centrar los rostros.

  Particularmente, pensamos que estos sistemas no tienen por qué suponer la desaparición de los dispositivos físicos como teclado, ratón y joystick, sino que el verdadero beneficio se obtiene aumentando –más que sustituyendo– las posibilidades de interacción con la máquina.

- **Indexación multimedia.** En estas aplicaciones, la entrada del sistema puede ser una fuente pública, o particular, de imágenes o vídeo: televisión, Internet, películas, vídeo casero, etc. El análisis automatizado evita la laboriosa tarea de buscar y etiquetar la presencia de rostros humanos. Algunas posibles aplicaciones pueden ser:

  - etiquetado y catalogación automática de secuencias de vídeo;

- búsqueda de contenido por criterios faciales;
- bases de datos de caras.

El estándar MPEG-7 viene a dar soporte a muchas de estas aplicaciones, al definir una manera de etiquetar las secuencias de vídeo e identificar distintos objetos en las mismas.

- **Generación gráfica y codificación de vídeo.** En estas aplicaciones converge la informática gráfica con la visión artificial. El objetivo de estos sistemas es transformar los resultados del procesamiento facial en nuevas imágenes. Tenemos como ejemplos:
  - sistemas de actores virtuales y captura de movimiento;
  - generación de avatares para comunicación a distancia entre personas;
  - compresión de vídeo para videoconferencia;
  - ocultación de la identidad de testigos y personas protegidas.

  En la actualidad, gran parte de los sistemas de captura de movimiento facial requieren la colocación de marcadores en el rostro del sujeto. Esto es una prueba evidente de que sigue siendo necesario mejorar los algoritmos de seguimiento e interpretación de las expresiones faciales.

- **Sistemas de ayuda a minusválidos.** Para personas con reducida movilidad, la utilización de la cara como un método de comunicación con las máquinas puede suponer un gran avance. De hecho, existen ya sistemas y proyectos encaminados a este tipo de aplicaciones. Los ejemplos de uso serían similares a los listados en el caso de los interfaces perceptuales. Sin embargo, la fiabilidad del sistema resulta más crítica, ya que una mala interpretación del rostro podría tener consecuencias más negativas.

En la actualidad, existen ya soluciones dentro de todas las categorías listadas. Por ejemplo, las empresas fabricantes de cámaras web están empezando a incorporar sistemas de seguimiento de caras y generación de avatares, como parte del software de las propias cámaras; su rendimiento es bastante interesante, aunque resultan claramente mejorables. A medida que vaya avanzando la disciplina del análisis facial, veremos multiplicadas las posibilidades de uso de estos sistemas de interacción visual con humanos.

1.2. Motivaciones para el uso de integrales proyectivas

Si en la sección anterior hemos motivado el indudable interés de abordar los problemas de procesamiento facial, en la presente vamos a exponer las razones por las que proponemos resolver dichos problemas desde el punto de vista de las integrales proyectivas[4]. El germen

[4]El término "proyectiva" aparece en otros ámbitos de la visión artificial que no deben ser confundidos con el que nos ocupa, como la *geometría proyectiva*. También existe una *geometría integral*, que incluye todas las transformaciones integrales en espacios arbitrarios.

de esta tesis surge de la convicción de que las proyecciones –una de las técnicas clásicas en la visión artificial– ofrecen una capacidad expresiva y una potencia mucho mayor que la que se ha obtenido con el tipo de técnicas aplicadas hasta la fecha, particularmente en análisis de caras humanas.

1.2.1. Historia de las proyecciones y técnicas relacionadas

La idea de una transformación sobre imágenes basada en la acumulación de valores a lo largo de una dirección surge por primera vez en 1914. Johann Radon [146], introduce y describe los posibles usos de una operación que posteriormente sería conocida como la *transformada de Radon*. Originalmente, la técnica está orientada a modelar el comportamiento de ciertos fenómenos, como el proceso de formación de imágenes en un sistema de rayos X. Intuitivamente, la clave es que cada valor puntual en la imagen de rayos X resulta de la acumulación de atenuaciones del rayo al atravesar determinados objetos.

Transformada de Radon 2D

La definición original de la transformada de Radon es la siguiente, [146]. Sea f una función en dos dimensiones, $f(x, y)$. Representamos una recta cualquiera en el plano con: $x\cos\theta + y\sin\theta = s$, siendo θ el ángulo de la recta, y s la menor distancia al origen. La transformada de Radon, $\mathcal{R}[f]$, es otra función 2D definida por:

$$\mathcal{R}[f](\theta, s) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - s)\, dx\, dy \tag{1.1}$$

Donde δ es la delta de Dirac. En la figura 1.10 se pueden ver dos imágenes de ejemplo y sus respectivas transformadas de Radon. Las imágenes transformadas se suelen denominar también *sinogramas*, puesto que el resultado para un objeto pequeño (un simple punto) es una función sinusoidal, como se aprecia en la figura 1.10b).

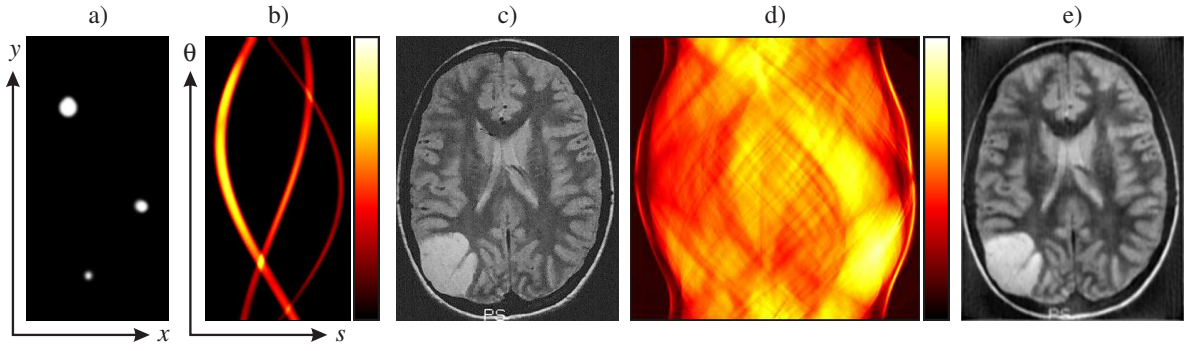


Figura 1.10: *Sinogramas, transformadas de Radon y transformada inversa. a) Imagen de ejemplo. b) Sinograma de la imagen a); se usa una escala de color (ver leyenda a la derecha) para destacar las variaciones. c) Imagen de ejemplo obtenida con TAC. d) Sinograma de la imagen c). e) Reconstrucción a partir del sinograma d), usando el algoritmo 2.1 (ver la página 47).*

Pero, realmente, el interés inicial de esta técnica no era aplicar la transformada sobre

imágenes existentes, sino calcular la transformación inversa de sinogramas dados. Este es el caso de la *tomografía axial computerizada* (TAC). En esta aplicación la entrada son los sinogramas 3D, y el objeto es reconstruir la estructura interna de los objetos. Estos problemas han despertado mucho interés en parte de la comunidad investigadora. Ya Radon en 1917 proponía un método de reconstrucción tomográfica, basado en las relaciones de su transformada con la de Fourier. En el capítulo 2 proponemos un método numérico más sencillo e intuitivo de reproyección, orientado a verificar que el proceso de proyección es invertible y, en consecuencia, no se pierde información al trabajar con integrales proyectivas. La figura 1.10e) es un ejemplo de la aplicación de este método.

Transformada de Hough

En 1972, Duda y Hart definen un operador destinado a la extracción de rectas y curvas de una imagen [45], basándose en una patente previa de Paul Hough de 1962. Curiosamente, la formulación de la *transformada de Hough* para el caso de las rectas resulta exactamente igual que la de la ecuación 1.1, propuesta por Radon 58 años antes.

Como novedad, generalizan el operador de Radon con la introducción de otros tipos de parametrizaciones de formas geométricas, asociadas a curvas, círculos, elipses. Posteriormente, la técnica sería popularizada hacia 1981 gracias al trabajo de Dana Ballard [5], que extiende el método a formas geométricas arbitrarias.

A diferencia de la transformada de Radon, el objetivo no es describir un fenómeno físico, sino detectar los segmentos y curvas más destacados en una imagen dada. El uso típico de la transformada de Hough incluye una serie de pasos comunes, ilustrados en la figura 1.11:

1. extracción de bordes con filtros de convolución u otros, por ejemplo, en la figura 1.11b) se ha usado el operador de Canny [22];

2. aplicación de la transformada de Hough sobre la imagen de bordes, figura 1.11c); y

3. búsqueda y selección de máximos locales en el sinograma resultante.

Idealmente, los máximos indican el ángulo y la posición de las rectas más *salientes*. Como puede comprobarse en la figura 1.11d), esto no siempre resulta tan evidente.

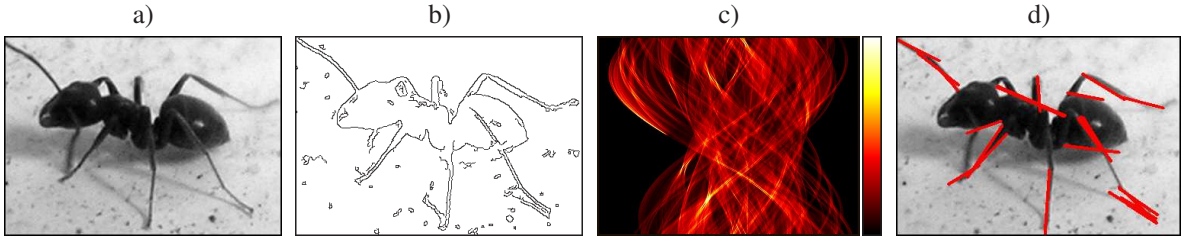


Figura 1.11: *Detección de segmentos utilizando la transformada de Hough. a) Imagen de entrada. b) Resultado del detector de Canny sobre la imagen a). c) Transformada de Hough de la imagen b). d) Selección de los segmentos más salientes.*

En casos como la detección de segmentos de la figura 1.11, la transformación se suele aplicar incrementado para cada píxel activo (de bordes) de la entrada, las celdas correspondientes del sinograma. Cada celda del mismo puede incluir información adicional, por ejemplo, para deducir los extremos del segmento correspondiente. No obstante, esto no varía el fundamento matemático del método.

Integrales proyectivas

Partiendo del contexto de la transformada de Radon –o, equivalentemente, de la de Hough para detección de rectas– las *integrales proyectivas* se pueden considerar como simples filas de un sinograma, fijando un ángulo concreto, θ. Así, la integral proyectiva vertical de una imagen i es la función $\mathcal{R}[i](0, y)$, y la proyección horizontal $\mathcal{R}[i](90^o, x)$, de acuerdo con la ecuación 1.1. No obstante, la literatura sobre el tema suele adoptar una notación simplificada, denotando por $PV(y)$ y $PH(x)$ a las proyecciones verticales y horizontales, respectivamente. De esta manera, podemos reducir la ecuación 1.1 a decir que $PV(y)$ es la media de los píxeles en la columna y de la imagen, y $PH(x)$ es la media de la fila x.

A pesar de la estrecha relación existente entre la transformada de Radon, la de Hough y las integrales proyectivas, pocos autores han descrito las similitudes e interacciones entre técnicas. Es más, en última instancia todas ellas son transformaciones lineales[5], al igual que la transformada de Fourier, del coseno, o las técnicas de proyección en autoespacios lineales.

Más bien, al haberse usado cada una de ellas en aplicaciones diferentes, han ido adoptando su propia notación y terminología. Si la transformada de Radon es usada en *reconstrucción tomográfica* e *imágenes médicas*, y la de Hough se aplica en *detección de rectas y curvas*, la utilización de integrales proyectivas está orientada normalmente al *análisis de imágenes* y la *extracción de características* distinguibles por niveles de gris.

Uno de los ejemplos típicos y donde más han sido usadas las proyecciones es en el *reconocimiento óptico de caracteres* (OCR). Las proyecciones permiten resolver de forma robusta muchas de las etapas preliminares del problema. En la figura 1.12 se muestra un posible ejemplo del proceso llevado a cabo en un OCR, desde la rectificación del texto hasta la segmentación de las palabras.

La aplicación típica de las proyecciones en OCR suele seguir la siguiente estructura:

1. **Determinar la orientación del texto.** Se aplican proyecciones en distintos ángulos (dentro del margen de inclinaciones permitidas), seleccionando el que provoque una señal con mayor varianza –figura 1.12a)–, que será normalmente el que mejor separe las líneas de texto del interlineado.

2. **Segmentación de las líneas** del texto. Una vez rectificada la imagen con el ángulo óptimo, la proyección vertical ayuda a encontrar la extensión vertical de cada línea. Para

[5]Es decir, aquellas en las que el valor de cada píxel de salida es una combinación lineal de todos los de la entrada. La única diferencia entre las diversas técnicas son los coeficientes de las combinaciones lineales.

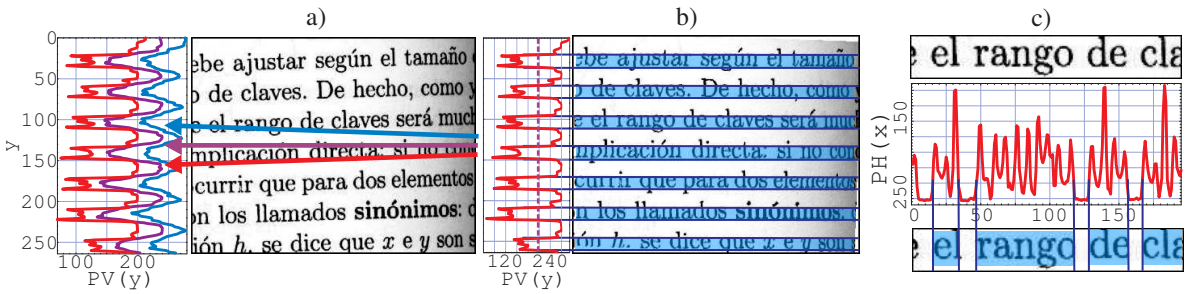


Figura 1.12: *Segmentación de texto en OCR con integrales proyectivas. a) Estimación de la inclinación del texto: se aplican proyecciones en distintos ángulos y se selecciona la de mayor varianza. b) La imagen rectificada se proyecta verticalmente y se umbraliza (segmentación de las líneas). c) Las líneas del texto se proyectan horizontalmente y se umbralizan (segmentación de las palabras).*

ello, podría bastar con una simple umbralización de la señal –figura 1.12b)–, puesto que la distinción entre texto y no texto es bastante evidente.

3. **Segmentación de palabras** dentro de una línea. La idea anterior se aplica también en sentido horizontal para encontrar los espacios. Como se puede ver en la figura 1.12c), la separación de letras concretas puede resultar más dificultosa. Sin embargo, la separación entre palabras es más sencilla porque aparecen zonas anchas de color claro.

De manera muy similar, las proyecciones han sido utilizadas también en la segmentación de matrículas de coches yen inspección de cadenas de transporte; se pueden situar dentro de los sistemas de OCR, los métodos para la lectura de fecha y hora en imágenes de CCTV [63].

Otros usos y variantes de las proyecciones

Aparte de los OCR, las proyecciones han sido aplicadas en otros muchos contextos. Es más, el dominio de las caras humanas es posiblemente uno de los más antiguos. En 1973 Takeo Kanade construye un sistema completo de localización y reconocimiento facial de personas basado en proyecciones [93]. En este trabajo, las proyecciones no son aplicadas sobre los niveles de gris, sino sobre las imágenes de bordes asociadas. De esta manera, los máximos locales de las proyecciones verticales y horizontales ayudan a determinar la extensión espacial del rostro y de sus principales componentes. La figura 1.13 muestra una comparación entre proyectar la intensidad o la imagen de bordes.

Las proyecciones han sido usadas también en los problemas de detección de caras (como veremos en el capítulo 3), localización de componentes faciales (capítulo 4) y en identificación biométrica (capítulo 6). En esta tesis añadimos otros problemas: el seguimiento (capítulo 5), el análisis de expresiones y la estimación de pose (capítulo 7). Y, lo que es más importante, desarrollamos una nueva metodología para el manejo de proyecciones basada en modelos, en lugar del tradicional análisis de máximos y mínimos. Se pueden encontrar algunos trabajos relacionados con el tipo de técnicas que proponemos en los dominios del *registro*, o *alineamiento*, *de imágenes médicas* y en la *estimación de movimiento con proyecciones*:

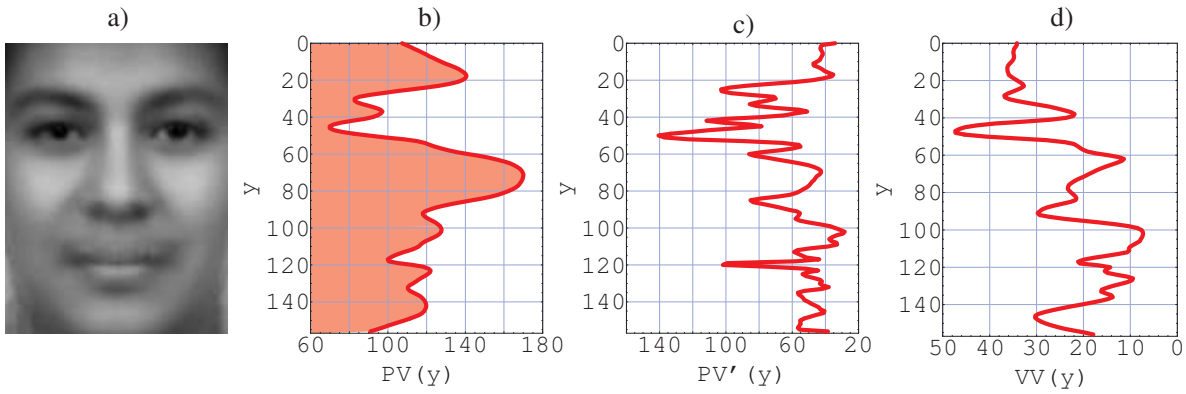


Figura 1.13: *Comparación entre distintos tipos de proyecciones verticales. a) Imagen de entrada. b) Proyección de la intensidad (obsérvese el parecido, no casual, de la curva con una cara vista de perfil). c) Proyección de la magnitud del gradiente (filtro de Sobel). d) Proyección de la varianza [50].*

- **Alineamiento de imagen médica.** El objetivo de estos sistemas es encontrar una transformación rígida 3D para alinear de forma óptima dos imágenes volumétricas, es decir, situar características comunes en las mismas posiciones. El problema es de elevada complejidad, debido a que existen 6 grados de libertad y se trabaja con información volumétrica. Algunos autores han sugerido el uso de proyecciones como una forma de simplificar el problema: las imágenes 3D se proyectan en 2D, a lo largo de diferentes direcciones; se realiza el alineamiento en 2D; y se deducen los parámetros de transformación 3D a partir de los alineamientos 2D. Se puede ver un trabajo reciente con algunas referencias adicionales en [95].

- **Estimación de movimiento.** Siguiendo un esquema similar al anterior, se han aplicado proyecciones 1D de las imágenes 2D para obtener rápidamente los vectores de flujo óptico por bloques. En el caso de movimiento de traslación y escala, es suficiente con 2 proyecciones en ángulos perpendiculares para resolver el problema. Añadiendo otras proyecciones se puede calcular también la inclinación y otros movimientos más complejos. La estimación del movimiento con proyecciones ha demostrado una elevada eficiencia, alta inmunidad al ruido blanco, y precisión comparable a las técnicas 2D [149] (para más información sobre estas técnicas, consultar la citada referencia).

El dominio de las caras añade algunas dificultades que son prácticamente inexistentes en las anteriores aplicaciones: los cambios de iluminación y pose; la variación no rígida por las expresiones faciales; las diferentes formas del rostro, color de piel y elementos faciales de las personas; la oclusión parcial, etc.

Volviendo al contexto del procesamiento de caras, algunos autores han planteado variantes del operador de proyección. En concreto, en 1998 Feng y Yuen [50], proponen las llamadas *proyecciones de varianza*, donde el valor de las señales no es la media de los píxeles, sino la varianza de la fila o columna correspondiente. Se argumenta que este método puede resultar más informativo en situaciones donde las proyecciones son poco discriminantes. En la figura

1.13d) se puede ver un ejemplo de esta operación.

Más recientemente, Zhou y Geng [213], introducen las *funciones de proyección generalizadas*, que definen como una simple media ponderada entre las integrales proyectivas y las proyección de varianza. En ambos casos, el problema estudiado es la localización de ojos. Describiremos más detenidamente estos trabajos en el apartado 4.2.1 del capítulo 4.

1.2.2. Propiedades de las proyecciones

Las integrales proyectivas ofrecen una serie de características interesantes, que apoyan su uso tanto en el caso general como en el dominio específico de las caras humanas. No pretendemos entrar aquí en detalles o justificaciones formales de estas propiedades. Trataremos más detenidamente algunas de ellas en el capítulo 3.

- **Invarianza frente a modos de variación de las imágenes.**

  En determinadas situaciones, las proyecciones ofrecen una capacidad de generalización muy superior al uso de patrones 2D –ya sean las propias imágenes o autoespacios asociados a las mismas–. Esto es debido a que las proyecciones resultan muy robustas con algunas transformaciones en las que un modelo 2D se vería más afectado. Obsérvense, por ejemplo, los casos de la figura 1.14.

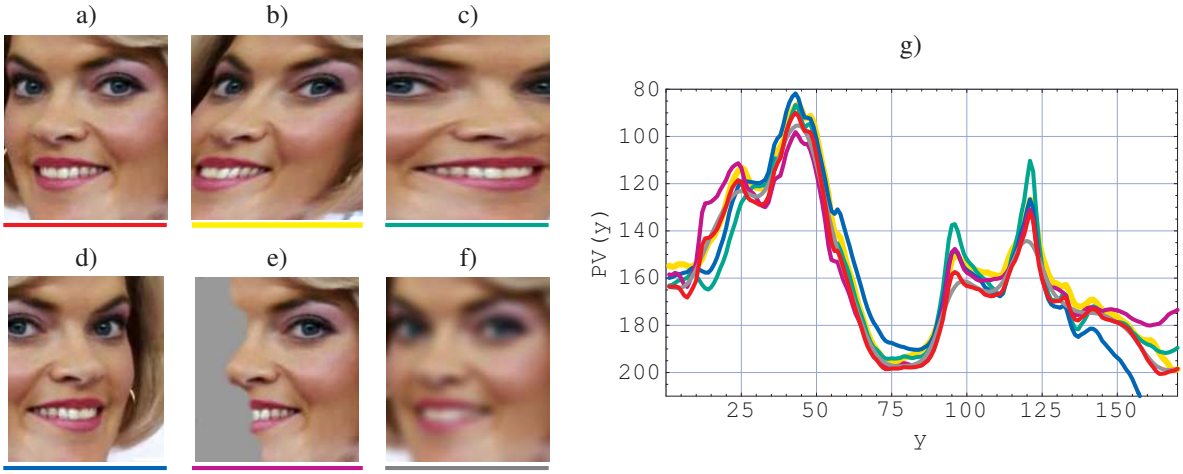


Figura 1.14: *Invarianza de las proyecciones frente a diversas transformaciones. a) Imagen original, extraída de la imagen "1083.avi.jpg" de la base UMU. b) Inclinación en X. c) Espejo y escalado en X. d) Espejo y traslación en X. e) Oclusión parcial. f) Suavizado gaussiano. g) Proyecciones verticales de las imágenes a-f).*

La imagen original, de la figura 1.14a), es sometida a suavizado, oclusión y a diferentes transformaciones geométricas en el eje X: inclinación, espejo, escalado y traslación. En los ejemplos de la figura 1.14, la comparación entre patrones 2D fallaría en la mayoría de los casos, ya que no existe un alineamiento perfecto entre las imágenes. Sin embargo, las proyecciones verticales resultan prácticamente invariantes a estos cambios, incluso aunque aparezcan fragmentos diferentes de la escena.

Por su parte, una transformación geométrica de escala/traslación de la imagen en el eje Y dará lugar a una idéntica escala/traslación de la proyección vertical. Y una transformación global de la intensidad en la imagen se traducirá en una modificación global equivalente en la proyección.

- **Inmunidad frente al ruido.**

  Son muchos los trabajos que han demostrado la gran robustez de las integrales proyectivas frente al ruido blanco [149, 213]. En la página 41 trataremos más detenidamente esta cuestión. Intuitivamente, el promediado de valores que supone el proceso de proyección hace que se compense el efecto del ruido a lo largo de todos los píxeles proyectados. Suponiendo que el ruido es aditivo y aleatorio, las proyecciones tienen un efecto de disminuir la varianza del ruido, sin afectar a la forma de las señales.

  En la figura 1.15 se muestra el comportamiento de la transformada de Radon frente al ruido aditivo. Sobre una imagen de entrada –figura 1.15a)– se han añadido distintos niveles y tipos de ruido –fila superior–. Después se han obtenido sus sinogramas, y a partir de ellos se reconstruyen las imágenes correspondientes.

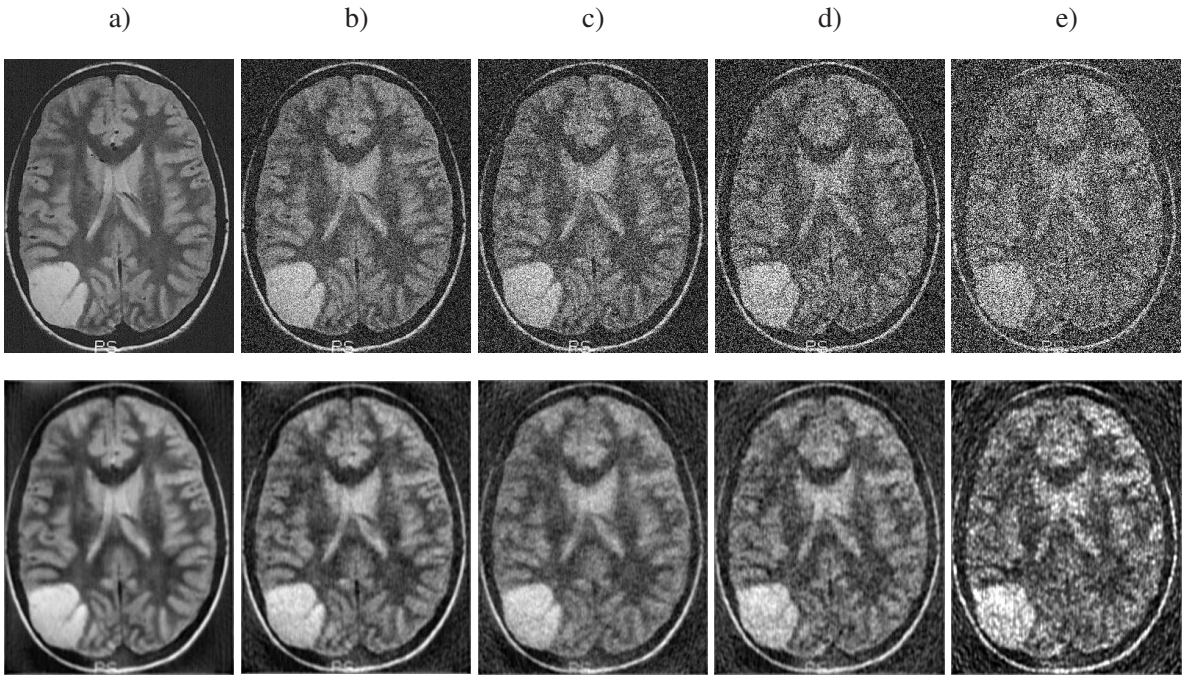


Figura 1.15: *Reproyección de imágenes con ruido aditivo y aleatorio. Fila superior: imágenes de entrada. Fila inferior: reproyecciones usando 100 proyecciones de las imágenes correspondientes. a) Imagen original. b) Con ruido gaussiano $\mathcal{N}(0, 40^2)$. c) Ruido gaussiano $\mathcal{N}(0, 60^2)$. d) Ruido uniforme en el intervalo (-128, 128). e) Ruido uniforme en (-256, 256).*

Los resultados de la figura 1.15 demuestran que las proyecciones minimizan el efecto del ruido, siendo capaces de trabajar incluso en condiciones donde la imagen está completamente degenerada. Si analizamos más detenidamente los resultados, podemos apreciar

que se introduce cierto grado de suavizado –en parte, también puede ser debido a los errores de precisión en los cálculos–; sin embargo, un suavizado gaussiano difuminaría la imagen al mismo tiempo que disminuye el ruido, algo que no ocurre en los casos de la figura 1.15.

- **Caracterización de clases de objetos.**

  Todas las propiedades anteriores resultarían irrelevantes si en los objetos de interés las proyecciones adoptan formas aleatorias e impredecibles, es decir, si no existe una estructura coherente. Por ejemplo, la proyección vertical u horizontal de un tablero de ajedrez es una señal constante. Sin embargo, esto no ocurre en nuestro dominio de interés. El rostro humano presenta una estructura común de zonas claras y oscuras asociadas a la frente, los ojos, la nariz, la boca, etc., aunque puedan existir diferencias entre individuos.

  Para comprobar esta propiedad, en la figura 1.16 se han representado casi 4000 proyecciones verticales de caras alineadas de unas 1200 personas distintas de la base FERET. Cada proyección corresponde a una simple columna de la imagen de la derecha.

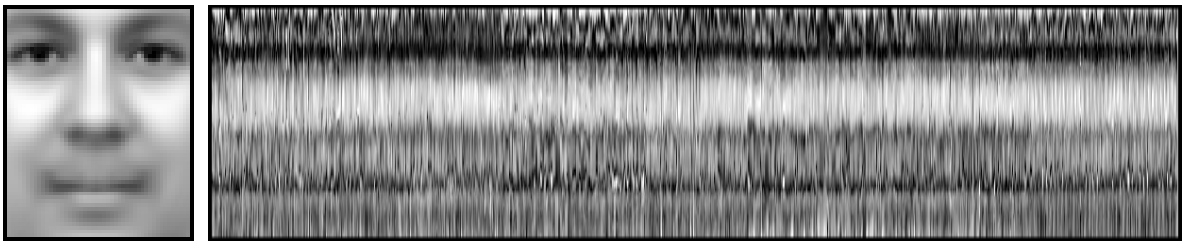


Figura 1.16: *Proyecciones verticales de 3818 caras humanas de la base FERET. Izquierda, la cara media del conjunto (con ajuste lineal del histograma). Derecha, las 3818 proyecciones verticales alineadas de 1196 individuos diferentes. Cada columna de la imagen representa una proyección.*

  Es bastante interesante observar la enorme similitud entre las diferentes proyecciones de la figura 1.16. Recordemos que la base FERET incluye imágenes de hombre y mujeres, de diferentes razas, con edades entre 10 y 72 años, distintos tipos de iluminación, con variaciones de la expresión facial, y algunas con pequeños giros laterales. A pesar de ello, la gran mayoría de las proyecciones se ajustan a una estructura común de zonas claras, oscuras e intermedias.

  Paradójicamente, la práctica totalidad de los trabajos previos se han limitado a buscar máximos y mínimos locales o zonas de variación rápida, sin hacer un uso *holístico* que aproveche la estructura común de las proyecciones asociadas al rostro. En la página 52 veremos que un sencillo modelo de proyección vertical media puede ofrecer mejor discriminación cara/no cara que un patrón 2D medio.

- **Invertibilidad y conservación de la información.**

  Existe la creencia extendida de que trabajar con proyecciones supone peder información de las imágenes. Una sola proyección es, en efecto, menos representativa que la imagen

completa. Pero un número suficiente de proyecciones en diferentes ángulos contiene tanta información como la imagen asociada. Las técnicas de reconstrucción tomográfica son la prueba de ello. La posible pérdida proviene de tomar un número reducido de integrales proyectivas, no del proceso de proyección en sí.

En el dominio de las caras humanas, unas pocas proyecciones bastan para conservar la mayor parte de la información relevante: la proyección vertical del rostro, la horizontal de los ojos, etc. Por ejemplo, la desviación estándar en niveles de gris de la cara media de la figura 1.16 es de 46,2. Si proyectamos esa imagen verticalmente, la desviación de la proyección es de 34,9 niveles de gris. De forma intuitiva, podemos decir que la proyección por sí sola conserva el 75,5 % de la información original[6].

- **Modos de variación de las proyecciones.**

  En relación con la conservación de información, también es interesante señalar que las proyecciones asociadas a los rostros humanos –aunque hemos visto que se aproximan a un patrón medio común–, disponen de modos de variación que nos permiten: diferenciar unos individuos de otros, analizar la expresión facial, hacer una estimación de la pose, etc. Para comprobarlo, hemos aplicado PCA sobre las 3818 proyecciones de la base FERET, calculando los principales modos de variación de las señales. Los autovalores y autovectores resultantes se pueden ver en la figura 1.17.

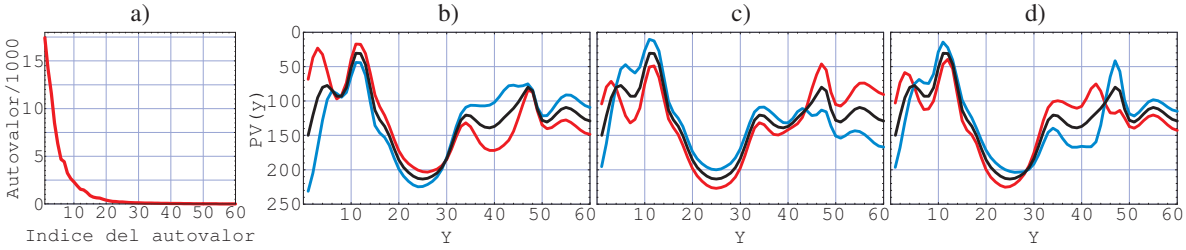


Figura 1.17: *Autovalores y autovectores de la matriz de covarianzas de las proyecciones verticales de cara. Los datos para el cálculo son 3818 proyecciones verticales de caras de la base FERET. a) Autovalores asociados a cada autovector, según el índice de los mismos. b,c,d) Primer, segundo y tercer modo de variación, respectivamente; es decir, la proyección media (en negro) más el autovector (en rojo) o menos el autovector (en azul), multiplicados por 1,5 veces la desviación típica asociada.*

El primer modo parece estar ligado a la forma global de la cara (separación cejas/ojos y nariz/boca), el segundo al estado de la boca (abierta, cerrada, entreabierta), y el tercero a la aparición de sombras. Los 15 primeros autovectores explican un 93 % de la varianza entre las señales. Podemos deducir, razonadamente, que las proyecciones varían dentro de un subespacio de reducida dimensionalidad.

- **Relación de adyacencia entre puntos.**

[6] A modo comparativo, en el contraejemplo del tablero de ajedrez, la desviación de la imagen sería 128, y la de la proyección 0: no se conserva nada de información de la imagen de entrada.

Ya hemos mencionado que las integrales proyectivas se pueden interpretar como técnicas de proyección en subespacios lineales: dado un vector de alta dimensionalidad, x, se expresa de forma compacta con otro vector $p = \{p_1, p_2, p_3, \ldots\}$. Sin embargo, a diferencia de otros métodos como PCA, LDA o ICA, las proyecciones mantienen la *propiedad de adyacencia* entre píxeles: dos puntos cercanos, p_k y p_{k+1}, corresponden a regiones adyacentes de las imágenes. Esto no sucede en los otros casos, donde p_k y p_{k+1} no tienen ninguna relación especial. Esta propiedad es esencial, ya que permite que el alineamiento tenga lugar a posteriori: antes de aplicar PCA la cara debe estar perfectamente alineada; con integrales proyectivas, se puede proyectar primero y alinear después.

- **Eficiencia computacional.**

  Otra de las grandes ventajas de las proyecciones es su elevada eficiencia computacional. Supongamos una imagen i de $n \times n$ píxeles, y una proyección asociada p de tamaño n. Un ajuste del contraste de i, por ejemplo, requeriría un $O(n^2)$, mientras que la operación equivalente en p es un $O(n)$. También una comparación por distancia euclídea entre dos imágenes, o entre dos proyecciones, tienen un orden de $O(n^2)$ y $O(n)$, respectivamente. La proyección de i en un autoespacio de tamaño m sería de $O(mn^2)$, y $O(mn)$ si lo que proyectamos es p. En esencia, pasamos de problemas 2D a 1D.

  La obtención de la proyección vertical u horizontal de tamaño n sería, en principio, un $O(n^2)$. No obstante, si se deben calcular muchas proyecciones sobre una misma imagen, se puede aplicar el concepto de *imagen integral* para conseguir reducir el coste a un $O(n)$. En el capítulo 3 estudiaremos esta técnica para el cálculo rápido de proyecciones.

1.3. Objetivos de la tesis y metodología de trabajo

La exposición realizada en las anteriores secciones nos lleva a reconocer una doble vertiente en los objetivos que se pretenden conseguir con la presente tesis doctoral: desarrollo de técnicas genéricas de modelado y análisis de imágenes usando integrales proyectivas; y estudio de problemas en el dominio específico de las caras humanas. Vamos a enunciar este objetivo, y a concretar algunos aspectos sobre el desarrollo de la investigación llevada a cabo. Acabamos la sección presentando la estructura del resto de esta memoria.

1.3.1. Objetivo principal de la tesis

El objetivo global de la tesis es estudiar y desarrollar técnicas de procesamiento de imágenes mediante proyecciones, y aplicarlas al análisis de imágenes de caras, diseñando e implementando soluciones para los problemas de:

- **Detección:** encontrar las caras que aparecen en una imagen y su extensión espacial, permitiendo un número arbitrario de rostros e imágenes en escala de grises. Se trabajará principalmente con caras en posición frontal o casi frontal.

- **Localización:** determinar la situación exacta en una imagen de las caras detectadas y la posición concreta de los dos ojos y la boca.
- **Seguimiento:** localizar todas las caras que aparecen a lo largo de una secuencia de imágenes, es decir, determinar las posiciones de los ojos y las bocas para cada instante.
- **Reconocimiento:** estudiar la efectividad de las proyecciones en los problemas de identificación en conjunto cerrado, verificación e identificación en conjunto abierto.
- **Extracción de información:** analizar la expresión facial de las caras seguidas, usando un conjunto predefinido de unidades de activación; y diseñar un método de estimación de posición y orientación 3D de las caras, para el manejo de un interface perceptual.

Todos estos problemas serán tratados dentro del contexto común de las integrales proyectivas. Se pretende así analizar cómo esta técnica puede ayudar en los diferentes subdominios, creando un marco unificado para el procesamiento de imágenes de caras humanas. Los resultados serán contrastados experimentalmente con otras técnicas disponibles y usadas habitualmente para esos mismos problemas.

1.3.2. Metodología de trabajo

En la consecución de los objetivos de la tesis, el autor de la misma parte de un trabajo previo plasmado en las publicaciones [58, 59, 60, 62, 57, 64, 61, 187], relacionadas directamente con diversos problemas del procesamiento visual de caras humanas. Tomando esta base, se ha elaborado un contexto teórico para el manejo de integrales proyectivas –previo a la resolución de los problemas específicos–, que describe las propiedades y operaciones de transformación sobre proyecciones, el modelado de objetos con proyecciones, y el proceso de alineamiento de señales unidimensionales.

A continuación, estas técnicas han sido aplicadas sobre los problemas descritos en los objetivos del apartado 1.3.1. El estudio de cada problema sigue un proceso metódico. En primer lugar hacemos un análisis de los requisitos y el tipo de acercamientos existentes. Después, el análisis desemboca en el diseño de una solución basada en el uso de modelos de proyección. Algunos de los algoritmos siguen el mismo esquema descrito en las publicaciones referidas, aunque la mayoría han sido refinados, mejorados y optimizados; varios problemas han sido abordados por primera vez en el contexto de la tesis.

Todas las propuestas realizadas han sido implementadas y verificadas en extensas series de experimentos que se describen en los sucesivos capítulos. Algunos principios y requisitos que han orientado el diseño de las diversas soluciones son los siguientes:

- Los algoritmos desarrollados deben ser **robustos** frente al mayor número posible de fuentes de variación: expresión facial, factores individuales, sistemas de adquisición, iluminación, calidad de la imagen, etc. En particular, creemos que un sistema de procesamiento de caras –como el que puede manejar un usuario de un sistema de videoconferencia– debe ser capaz de trabajar con cámaras web de bajo coste.

- Nuestro principal objeto de trabajo serán imágenes capturadas de **fuentes de vídeo**: cámaras web, cámaras comerciales/industriales, televisión analógica, TDT y DVD, ya sea extrayendo imágenes estáticas o secuencias de vídeo. Esto no impide que algunas pruebas utilicen también imágenes de fotografía digital o analógica escaneadas. Este requisito se fundamenta en que la entrada de vídeo suele ser mucho más frecuente en los sistemas de procesamiento facial.

- En cuanto a la **orientación** de los rostros, como ya hemos mencionado, estamos interesados principalmente en el análisis de caras que aparezcan de frente, o casi de frente. No obstante, se deben admitir márgenes de rotación suficientes, del orden de $\pm 15^{o}$ para la inclinación (rotación respecto del plano de imagen), $\pm 20^{o}$ para el giro vertical (mirada arriba/abajo), y unos $\pm 30^{o}$ para el giro lateral (mirada izquierda/derecha).

- Aunque se admite el uso de imágenes en color, todos los algoritmos deben ser capaces de trabajar en **escala de grises**. Esta restricción se deriva de dos hechos: (1) en algunos casos, simplemente viene impuesta por la entrada disponible; (2) incluso cuando las imágenes sean en color, resulta prácticamente imposible garantizar siempre la constancia del color.

- Todos los algoritmos implementados se deben **integrar** perfectamente, sin requerir la intervención manual del usuario. La salida del detector será la entrada para el localizador; éste producirá un resultado para el seguidor; y los resultados del mismo serán usados en análisis de expresiones y estimación de pose. También el reconocedor de personas trabajará con la salida de detector y localizador.

- La viabilidad práctica de un método no está desligada de su **eficiencia** computacional. Se evitarán mecanismos cuyo coste no sea asumible en un ordenador medio actual. También se deberá buscar la máxima simplificación del **entrenamiento** de los algoritmos. Los modelos usados en los distintos problemas serán aprendidos a partir de ejemplos, pero se pretende evitar un entrenamiento para usuarios concretos o condiciones demasiado específicas.

Implementaciones y entorno de desarrollo

Indudablemente, la programación de los métodos desarrollados desempeña un papel trascendental en el dominio que nos ocupa. De hecho, algunos estudios han señalado que los aspectos de implementación pueden tener una influencia clave en los resultados de los algoritmos [52, 143, 212][7]. Las herramientas concretas usadas en esta tesis son las siguientes:

- **Lenguaje de programación C++.** Pensamos que no es necesario extenderse en la justificación del uso de C++: es un lenguaje muy potente, ampliamente extendido, y con una

[7] En concreto, se comparan diversos reconocedores faciales mediante *autocaras*, pero utilizando diferentes implementaciones (distintas regiones extraídas, tamaño de las mismas, y métricas en el autoespacio). Los porcentajes de identificación correcta pueden variar hasta en unos 12 puntos.

elevada flexibilidad. Los mecanismos avanzados de orientación a objetos (polimorfismo, clases abstractas, y ligadura dinámica) simplifican la construcción y uso de diversos métodos alternativos para un mismo problema[8].

- **Entorno Borland C++ Builder.** Para nuestros propósitos, la propiedad básica de un buen entorno de desarrollo es la simplificación del proceso de programación: escritura del código, creación de la interface de usuario, compilación, depuración y documentación. Seleccionamos Borland C++ Builder, en concreto la versión 6, porque se ajusta muy bien a la mayoría de estos requisitos. No obstante, en la funcionalidad de procesamiento de caras se ha evitado la introducción de librerías o aspectos que no forman parte del estándar de C++.

- **Librerías de procesamiento Intel IPL y OpenCV.** Apoyarse en unas buenas librerías de procesamiento de imágenes resulta imprescindible para evitar un trabajo laborioso e innecesario. Intel OpenCV[9] (*Intel Open Source Computer Vision Library*) es una librería de código abierto, gratuita (tanto para uso comercial como no comercial), muy completa, multiplataforma, rápida, fácil de utilizar, en continuo desarrollo y cada vez más extendida entre la comunidad de visión artificial. En el dominio específico de las caras humanas, contiene algunas funcionalidades avanzadas para la detección y seguimiento del rostro. Además, se solucionan algunos aspectos auxiliares como la lectura y escritura de imágenes y vídeo, y la captura de cámara. En particular, hemos manejado la versión beta 5 (aunque recientemente se ha publicado una versión más reciente, y esperamos que sigan apareciendo en el futuro).

  Unas pocas operaciones no incluidas en OpenCV han sido tomadas de las librerías Intel IPL (*Intel Image Processing Library*). Este paquete es también gratuito aunque no de código abierto. Hacia el año 2000 su desarrollo fue reemplazado por las librerías IPP[10] (*Intel Performance Primitives*), aunque éstas no son completamente gratuitas.

Experimentación y bases de caras

Una buena parte del trabajo llevado a cabo en esta tesis se ha centrado en la validación experimental de los métodos propuestos, tanto de forma cualitativa como cuantitativa. Para ello, se han utilizado conjuntos grandes de imágenes donde aparecen rostros humanos, junto con etiquetados manuales de la posición de cada uno –lo que llamamos una *base de caras*–. Algunas bases públicas están orientadas a ciertos problemas o subdominios específicos, por lo que lo más inmediato es utilizar una base propia con el tipo de imágenes de interés. En

[8] Por poner un pequeño ejemplo, existe una clase abstracta FaceRecognitor, de la que se derivan: TemplateFR, IntProyFR y EigenFaceFR. La segunda tiene como descendientes: MeanDistIPFR, NearNeightIPFR y KNearNeightIPFR. La duplicación de código en los procesos de entrenamiento y validación de los reconocedores es inexistente, ya que el código común aparecen en la parte más alta de la jerarquía de clases.

[9] Ver la página web oficial del proyecto en: `http://www.sourceforge.net/projects/opencvlibrary`.

[10] Ver la página web oficial del proyecto en: `http://www.intel.com/software/products/ipp/`.

la medida de lo posible, hemos intentado manejar también bases públicas. Los conjuntos de imágenes a los que nos referiremos en los experimentos son los siguientes:

- **Base de caras UMU.** Denominaremos así al conjunto de imágenes propias. La base contiene un total de 737 imágenes en las que hay etiquetadas 853 caras humanas. La mayoría de las imágenes son de fuentes de vídeo: televisión analógica (381 imágenes/450 caras), películas de DVD (140/162), TDT (92/120), y webcam (56/57). Unas pocas imágenes (10 casos, con 14 caras) están tomadas con cámara fotográfica digital. Las imágenes de televisión corresponden a canales públicos españoles o extranjeros (TVE, Antena 3, Tele 5, Telemadrid, Canal Sur, BBC World, etc.), principalmente de programas de noticias, publicidad, series y similares, capturados con una tarjeta doméstica de sintonización de TV. En la figura 1.18 se pueden ver algunos ejemplos típicos.

Figura 1.18: *Imágenes de ejemplo de la base de caras UMU.*

Algunas imágenes están tomadas en secuencia (es decir, en cortos intervalos sobre un mismo vídeo), donde se observan variaciones de posición y expresión de unos mismos individuos, aunque la mayoría son de escenas completamente distintas. Para las imágenes de webcam se han usado varias cámaras: Logitech QuickCam Pro, QuickCam Pro 5000, Creative Webcam NX Pro, y también una Sony DFWL 500 –aunque, realmente, esta última no se puede considerar como una webcam–. Se han incorporado al conjunto 34 imágenes de la base pública CMU/MIT (64 caras), con condiciones de resolución y calidad similares al resto de imágenes. Los tamaños de las imágenes son variados, siendo la media de 534 $\times$ 393 píxeles; y el tamaño medio de las caras (distancia entre los

ojos) es de unos 30 píxeles. En total aparecen 434 hombres y 419 mujeres.

- **Base de caras CMU/MIT.** Esta base fue creada por Henry Rowley durante el desarrollo de su tesis doctoral [152, 153], usando imágenes propias y algunas recopiladas previamente por Sung y Poggio [173]. El conjunto está orientado a la evaluación de detectores de caras de frente, en condiciones de baja resolución y calidad de captura. Las imágenes son muy variadas en diversos aspectos: desde 60×75 píxeles, hasta 1280×1024; desde imágenes de grupo con 57 caras, hasta más de media docena donde no aparece ninguna persona. Todas las imágenes están en escala de grises (y algunas de ellas son binarias). Para nuestras pruebas, hemos descartado los casos donde aparecen caricaturas o dibujos de caras. En total, manejamos 109 imágenes con 482 caras.

- **Base de caras FERET.** Se trata del conjunto de imágenes usado en las evaluaciones públicas del programa FERET [52, 143, 144] (Face Recognition Technology) del ARL (Army Research Laboratory). Ya hemos mostrado algunas imágenes medias de esta base en las figuras 1.6 y 1.16. El conjunto contiene unas 14.000 imágenes de 1196 individuos, tomadas entre 1994 y 1996. En todos los casos aparece una sola cara centrada en la imagen, con fondo más o menos uniforme. Originalmente se distribuían las versiones en escala de grises, aunque actualmente están disponibles en color. Muchas de las imágenes están etiquetadas en las posiciones de ojos y boca. Para algunas se dispone también de información como la raza del sujeto, su edad, la fecha de captura, la expresión facial y el grado de giro lateral. Utilizaremos estas imágenes en las pruebas de localización de componentes y en reconocimiento facial.

- **Bases para el reconocimiento de caras.** Algunos otros conjuntos de imágenes han sido usados exclusivamente en el capítulo 6 para la evaluación del reconocimiento. En concreto, hemos manejado las bases de la Universidad de Essex creada por Libor Spacek [82], del Instituto Tecnológico de Georgia por Nefian y Hayes [127], y la del ORL (Olivetti Research Laboratory) [159]. Describiremos más detenidamente estos conjuntos en los apartados correspondientes del citado capítulo.

Existen otras muchas bases de caras a disposición pública (véase un repaso completo en [204] y en el capítulo 13 de [108]). Por otro lado, en las pruebas de seguimiento también se han utilizado capturas de televisión y DVD, secuencias propias obtenidas con webcam, y algunos los vídeos ofrecidos públicamente por otros investigadores [19, 70]. Detallaremos estas secuencias en los experimentos del capítulo 5.

1.3.3. Estructura de la memoria de la tesis

La estructura del presente documento es un reflejo de la propia organización de los objetivos y la metodología de trabajo aplicada para la consecución de los mismos. Así, existe un desarrollo teórico de los conceptos relacionados con las proyecciones, seguido de una serie de

aplicaciones de los mismos en el dominio de los rostros humanos. En particular, en el capítulo 2 formalizamos el concepto de integral proyectiva, estudiamos sus principales propiedades y tratamos dos aspectos clave en el manejo de las mismas: el modelado de clases mediante proyecciones, y el problema de alineamiento entre señales.

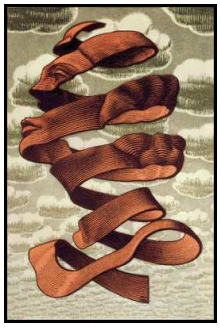
A continuación abordamos los grandes problemas del procesamiento facial, siempre desde el punto de vista de las integrales proyectivas. Todos estos capítulos de aplicación presentan una estructura común:

- en primer lugar se propone una definición para el problema, se analizan las características del mismo y las dificultades que plantea, y se describen las medidas de rendimiento más habituales en la literatura;
- seguidamente se estudia el estado del arte de ese ámbito específico de la investigación, con una atención especial a las técnicas más relacionadas, basadas en proyecciones o en conceptos similares;
- una vez hecho el análisis, se describe cómo las proyecciones pueden ayudar a resolver el problema, y se desarrolla un algoritmo adecuado, trabajando exclusivamente con integrales proyectivas; en algunos casos planteamos la posibilidad de realizar una combinación de técnicas, como una manera de mejorar las técnicas elementales subyacentes;
- el método diseñado es evaluado en una extensa serie de experimentos, comparando sus resultados con otras alternativas disponibles: algoritmos accesibles públicamente, técnicas base de implementación propia, y resultados de diversas publicaciones sobre bases estándar;
- se acaba siempre con una valoración y discusión de los resultados, y un resumen somero de las cuestiones tratadas en el capítulo.

Siguiendo ese esquema lógico, en el capítulo 3 se aborda la detección de caras en imágenes estáticas y con un número arbitrario de instancias por imagen. El capítulo 4 trata la localización de componentes faciales –en concreto, ojos y boca– tomando como partida los resultados de los detectores. El seguimiento de caras en secuencias de vídeo se estudia en el capítulo 5. Después se discute en el capítulo 6 la utilización de integrales proyectivas en los problemas biométricos de reconocimiento facial. Para acabar con las aplicaciones prácticas, el capítulo 7 da una pincelada de cómo las técnicas propuestas pueden ser aprovechadas para la resolución de otros problemas sobre rostros humanos; en particular, se diseñan soluciones sencillas para el análisis de expresiones faciales y la estimación de pose, aplicadas a un sistema de generación de avatares y un interfaz perceptual, respectivamente.

Finalmente, en el capítulo 8 se exponen las conclusiones del trabajo desarrollado, señalando la novedad y originalidad de las propuestas realizadas, sintetizando las aportaciones de los diferentes capítulos, y vislumbrando las líneas de trabajo más prometedoras que se derivan de la investigación documentada en esta tesis.

CAPÍTULO 2



"Rind", M.C. Escher, 1955

Integrales Proyectivas

"La perfección se encuentra en las cosas simples,
no en la confusión y la multiplicidad."
ISAAC NEWTON

El manejo de espacios de elevada dimensionalidad es una de las características esenciales de la visión artificial; al mismo tiempo, complica y da sentido propio a la disciplina. Fácilmente, cualquier sistema que use imágenes de tamaño medio se verá abocado a trabajar con vectores de entrada de unos cuantos miles de valores. Para abordar este ingente volumen de datos, la reducción a subespacios de menor dimensionalidad es uno de los fundamentos subyacentes en muchas de las técnicas de la visión en general, y del procesamiento de caras humanas en particular.

Una categoría de especial relevancia son los denominados *subespacios lineales*, obtenidos a través de transformaciones lineales sobre las imágenes –es decir, aquellas operaciones en las que cada valor de salida es una combinación lineal de todos los píxeles de la entrada–. Las *autocaras* [183, 125], las *fishercaras* [9], el análisis de componentes independientes [6], los filtros *wavelets* de Haar [188, 110], y Gabor [103, 186, 192, 162], además de las ubicuas transformadas de Fourier y del coseno [190, 127], son algunas de las técnicas más utilizadas en el dominio de los rostros humanos.

Las integrales proyectivas son una técnica más de reducción a subespacios lineales, donde cada valor de salida es la media aritmética de una fila o columna de píxeles de la entrada; de hecho, ya vimos en el apartado 1.2.1 que se pueden entender como una reformulación de la transformada de Radon [146, 42]. Las proyecciones han sido usadas en diversas aplicaciones del procesamiento facial, normalmente combinadas con otras características, como analizamos en el capítulo 1. Sin embargo, ya mencionamos que la mayoría de los trabajos se apoyan en métodos heurísticos *ad hoc*, derivados del conocimiento a priori sobre el problema.

En este capítulo vamos a desarrollar un contexto *formalizado* para el manejo de integrales proyectivas, que nos permitirá ir más allá del simple uso intuitivo de trabajos previos. Estamos convencidos de que las proyecciones ofrecen posibilidades que aún no han sido explotadas en toda su potencia. En primer lugar, empezamos proponiendo definiciones para los conceptos relacionados, en la sección 2.1. Sugerimos una notación para las integrales proyectivas, estudiamos las transformaciones sobre las mismas, y describimos algunas propiedades relevantes. En la sección 2.2, abordamos la cuestión de cómo modelar el conjunto de proyecciones asociadas a cierta categoría de objetos. A continuación, se estudia el problema del alineamiento de integrales proyectivas en la sección 2.3. Como veremos, es uno de los problemas fundamentales al trabajar con proyecciones. Proponemos un algoritmo eficiente y robusto para resolverlo, que será utilizado en las aplicaciones de los siguientes capítulos. Para acabar, la sección 2.4 resume las aportaciones más interesantes del presente capítulo.

2.1. Definiciones y propiedades

2.1.1. Definiciones básicas

Antes de proponer una definición matemática para las integrales proyectivas, resulta imprescindible formalizar algunos conceptos previos y básicos relacionados con las imágenes. Si bien la mayoría de los términos resultan triviales y bien conocidos en el mundo del procesamiento de imágenes, creemos conveniente empezar eliminando cualquier posible ambigüedad en lo sucesivo.

Empecemos con el concepto de imagen.

Definición 2.1 ***Imagen.***

Una imagen, i, con n canales es una función discreta en dos variables:
$i : [0, \ldots, x_{max}] \times [0, \ldots, y_{max}] \rightarrow \mathbb{R}^n$

Usando la notación matemática de funciones, supondremos que el valor de un píxel (x, y) se puede obtener mediante $i(x, y)$. Denotamos al conjunto de todas las imágenes de n canales por $\mathbb{I}^n$. Normalmente trabajaremos con imágenes en escala de grises –es decir, con el conjunto $\mathbb{I}^1$–, o bien con imágenes en el espacio de color RGB –esto es, $\mathbb{I}^3$–. En el segundo caso, accederemos a cada uno de los canales de un píxel con: $i(x, y).r$, $i(x, y).g$ e $i(x, y).b$.

Por conveniencia, introducimos también el concepto de *región* dentro de una imagen, como un conjunto contiguo de píxeles en el dominio de la imagen.

Definición 2.2 ***Región de una imagen.***

Una región, R, en una imagen, i, es un conjunto de pares de píxeles: $\{(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)\}$, con $x_i \in \{0, \ldots, x_{max}\}, y_i \in \{0, \ldots, y_{max}\}, \forall i \in \{1, \ldots, k\}$. Además, el conjunto de píxeles debe ser contiguo, es decir, todos los píxeles deben ser adyacentes entre sí, directamente o a través de otros píxeles del conjunto.

La restricción de contigüidad es importante para dar sentido al concepto de dominio, o rectángulo contenedor, de una región. Definimos el *dominio en X de una región R* como el conjunto de valores enteros: $dominioX(R) = \{min_{i=1..k}\{x_i\}, \ldots, max_{i=1..k}\{x_i\}\}$, y de manera parecida el *dominio en Y* como: $dominioY(R) = \{min_{i=1..k}\{y_i\}, \ldots, max_{i=1..k}\{y_i\}\}$.

Frente a las imágenes, como matrices de números, tenemos las integrales proyectivas, como señales unidimensionales y discretas. Así pues, manejaremos también el concepto de *señal*, que queda definido de la siguiente forma.

Definición 2.3 ***Señal unidimensional.*** *Una señal unidimensional, s, (o, simplemente, una señal) es una función discreta en una variable:*

$s : [s_{min}, \ldots, s_{max}] \rightarrow \mathbb{R}$

Obsérvese que una señal puede variar tanto en su conjunto de entrada –también llamado el *dominio de la señal*–, como en el de salida –lo que denominamos el *valor de la señal*–. Denotamos por $dominio(s)$ al conjunto de enteros $\{s_{min}, \ldots, s_{max}\}$. De esta forma, las señales quedan definidas entre un mínimo, s_{min}, y un máximo, s_{max}, al contrario que las imágenes, para las cuales hemos supuesto que el origen es siempre el píxel (0,0). En este sentido, podemos decir que las imágenes definen su propio origen de coordenadas[1], mientras que en las proyecciones el origen no es fijo, sino que viene dado desde fuera. A efectos prácticos, esta propiedad es interesante cuando definamos las transformaciones de las señales en el dominio, como veremos más adelante.

Igual que hemos hecho con las imágenes, podemos considerar el conjunto de todas las señales unidimensionales, que denotaremos por $\mathbb{S}$. Las integrales proyectivas son elementos de este conjunto, obtenidos promediando los valores de gris de una región a lo largo de cierta dirección. Básicamente, tenemos proyecciones horizontales y verticales.

Definición 2.4 ***Integrales proyectivas verticales y horizontales.*** *La integral proyectiva vertical (o, simplemente, la proyección vertical) de una región R en una imagen en escala de grises, $i \in \mathbb{I}^1$, es una señal unidimensional, PV_R:*

$PV_R : dominioY(R) \rightarrow \mathbb{R}$

definida por:

$PV_R(y) := \overline{i(x,y)},\ \forall (x,y) \in R$

De forma similar, la proyección horizontal de una región R en una imagen en escala de grises, $i \in \mathbb{I}^1$, denotada por PH_R, es la señal:

$PH_R : dominioX(R) \rightarrow \mathbb{R}$

definida por:

$PH_R(x) := \overline{i(x,y)},\ \forall (x,y) \in R$

[1] Normalmente situado en el píxel superior izquierdo, en el formato *top-left* habitual.

En las figuras 2.1 y 2.2 se muestran algunos ejemplos de proyecciones verticales y horizontales de dos imágenes diferentes. En ambos casos, se han proyectado las imágenes completas, es decir, la región R sería toda la imagen de entrada.

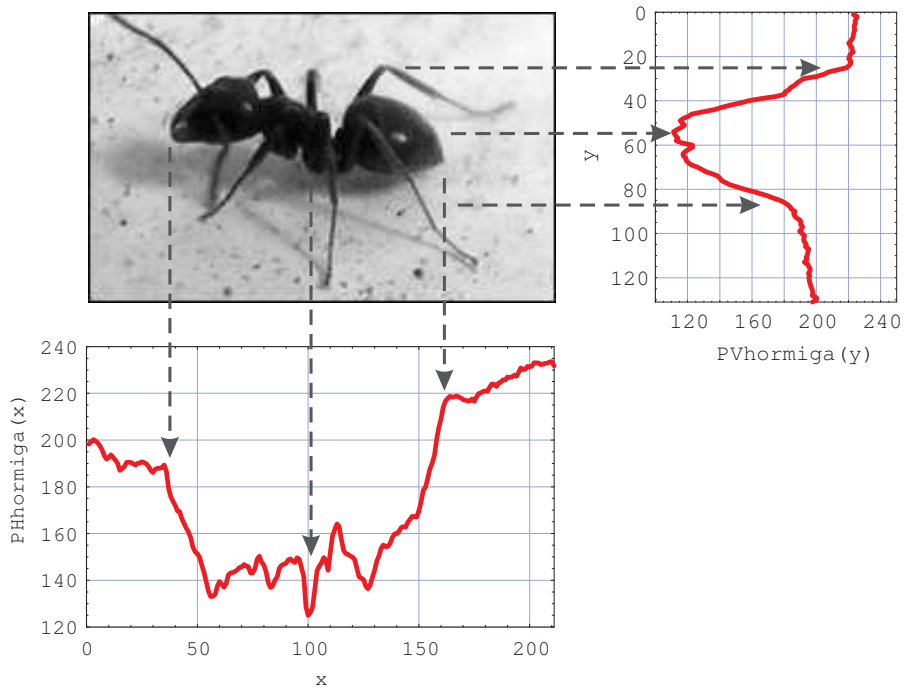


Figura 2.1: *Ejemplo de integral proyectiva horizontal y vertical de una imagen. Arriba a la izquierda, la imagen de entrada, hormiga. A la derecha, la proyección vertical, $PV_{hormiga}$. Abajo, la proyección horizontal, $PH_{hormiga}$.*

El primer hecho notable –aunque, por otra parte, bastante evidente– es que cuando proyectamos objetos sencillos, como los de ambos ejemplos, aparecen claramente destacados en las proyecciones los elementos de interés (la posición de la hormiga, en la figura 2.1, y la posición de los elementos faciales en la figura 2.2). Es más, podemos asegurar que la proyección de una gaussiana 2D es una gaussiana 1D, puesto que la suma de normales es también una normal. Veamos algunas consideraciones adicionales sobre los conceptos introducidos hasta ahora:

- Algunos investigadores optan por llamar "proyección horizontal" a lo que nosotros hemos definido como "proyección vertical", y viceversa. Incluso los hay que utilizan el término "perfil" en lugar de "proyección". Desafortunadamente, no hay un criterio uniforme en la denominación de las proyecciones, y también se pueden encontrar otros muchos trabajos que usan la misma nomenclatura que nosotros. Particularmente, creemos que es más natural referirse a una proyección por el eje en el que está definida la señal, más que por el eje a lo largo del cual se suman los píxeles[2].

[2]Por ejemplo, una proyección vertical está definida a lo largo del eje Y, pero se obtiene sumando los píxeles con el mismo valor de X. Obviamente, ambos ejes son perpendiculares, lo que da lugar a la confusión de términos existente en la literatura sobre el tema.

- Por simplicidad, hemos definido las proyecciones sobre imágenes en escala de grises, obviando las imágenes con varios canales. No resulta difícil extender el concepto de señal al caso multicanal y, consecuentemente, las proyecciones de imágenes con más de un canal. Sin embargo, la mejora que pueda suponer esta extensión resulta más discutible. En su lugar, en el caso de las imágenes multicanal, escogeremos uno de los canales para obtener las proyecciones. Por ejemplo, la proyección vertical del canal rojo de una imagen RGB, i, vendría dada por $PV_R(y) := \overline{i(x,y).r},\ \forall (x,y) \in R$. En la figura 2.2 se muestran las proyecciones de los tres canales de una imagen en color. Se puede apreciar que, aunque las tres señales resultantes toman valores dispares, todas ellas tienen una estructura muy parecida.

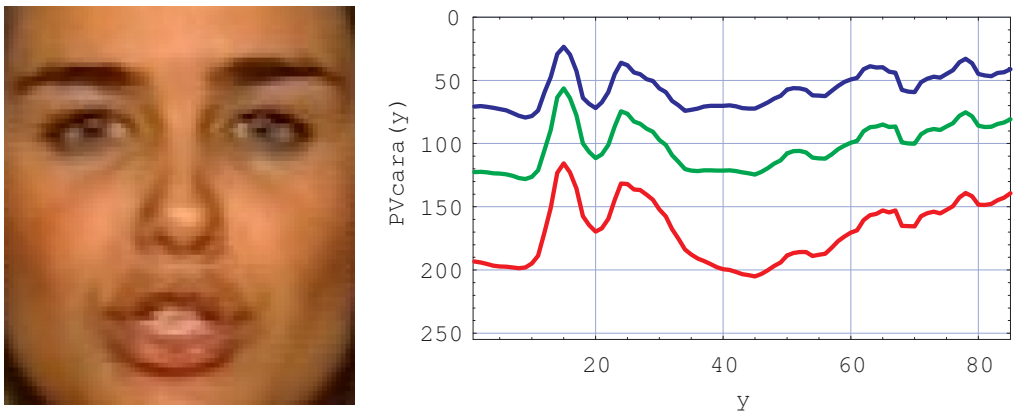


Figura 2.2: *Integrales proyectivas verticales de los canales R, G y B de una imagen. A la izquierda, la imagen de entrada en color, cara. A la derecha, las proyecciones verticales, PV_{cara}, de los canales R (en rojo), G (en verde) y B (en azul). Observar que se ha dado la vuelta a la escala vertical de la gráfica (0-arriba, 250-abajo). De esta forma, los elementos faciales aparecen como picos de las señales. En adelante usaremos esta misma forma de representar las proyecciones.*

- El cálculo de las proyecciones verticales y horizontales puede realizarse de forma muy eficiente, y en especial usando el concepto de *imágenes integrales* [188] (como veremos en el capítulo 3). Sin embargo, está claro que también tiene sentido obtener proyecciones a lo largo de otras direcciones cualesquiera. Más específicamente, podemos definir la *proyección de la región R a lo largo de un ángulo* dado, α, como la proyección vertical de R en la imagen rotada en ángulo α.

- Ya hemos mencionado que las integrales proyectivas están estrechamente relacionadas con la transformada de Radon [146]. Sin embargo, ambas transformaciones aparecen en ámbitos de aplicación muy diferenciados, por lo que tradicionalmente cada una ha seguido su propia terminología. Por un lado, las proyecciones requieren una notación simplificada, al trabajar con un número reducido de ángulos y señales discretas[3]. Por otro lado, la transformada de Radon está orientada fundamentalmente a la reconstrucción tomográfica [42], aunque ya vimos en el capítulo 1 que se han usado también en

[3]De hecho, las proyecciones se definen como señales 1D, mientras que la transformada de Radon es 2D.

algunos problemas de visión artificial [178, 149, 95].

2.1.2. Transformaciones sobre proyecciones

Un contexto completo y adecuado para trabajar con proyecciones debe incluir, necesariamente, un conjunto de operaciones de transformación sobre las mismas, que permitan manipularlas después de que hayan sido obtenidas. En general, una transformación sobre integrales proyectivas será cualquier función: $\mathbb{S} \rightarrow \mathbb{S}$. Concretamente, las operaciones que vamos a presentar se pueden entender como particularizaciones al caso 1D de las operaciones 2D definidas comúnmente sobre las imágenes.

Operaciones globales que modifican el valor

Hemos visto en el apartado 2.1.1 que las señales pueden variar tanto en el dominio a lo largo del cual se extienden, como en el valor que toman para cada punto[4]. Por lo tanto, las transformaciones pueden modificar el dominio de las señales, el valor, o ambos. Veamos en primer lugar las operaciones que modifican exclusivamente el valor de las señales.

Definición 2.5 ***Transformación en el valor.***
Una operación, t, de transformación en el valor de señales unidimensionales es una función:
$t : \mathbb{S} \rightarrow \mathbb{S}$
definida por:
$t(s)(i) := f(s(i))\ ;\ \forall i \in dominio(s)$
donde $f : \mathbb{R} \rightarrow \mathbb{R}$, es la función que define el valor resultante para cada valor de entrada.

Las operaciones de transformación en el valor son análogas a las *operaciones de procesamiento global* sobre imágenes –también conocidas como *píxel-a-píxel*–, es decir, aquellas donde todos los píxeles son tratados de forma independiente. Es más, para ciertas funciones f se cumplirá la **propiedad conmutativa**: el resultado de aplicar f a una integral proyectiva es equivalente a aplicar f a los píxeles de la imagen y después obtener la proyección. Esto sucede, por ejemplo, con las operaciones de sumar una constante, multiplicar por una constante, y en general con cualquier función lineal. La demostración es trivial.

Las transformaciones en el valor son interesantes para **normalizar**, de forma global, el conjunto de valores que toman dos o más proyecciones distintas. Obsérvense, por ejemplo, las tres proyecciones de la figura 2.2. Todas ellas presentan una estructura similar de picos máximos y mínimos, correspondientes a los elementos faciales más destacados: cejas, ojos, nariz y boca. Sin embargo, toman valores completamente distintos al haber sido obtenidas de tres canales diferentes, R, G y B. Podemos normalizar los valores de las señales, que originalmente están en el intervalo [0, ..., 255], a valores entre 0 y 1 con una operación del tipo:

[4]Para evitar ambigüedades, utilizaremos normalmente el término "punto" para referirnos a las posiciones en una señal unidimensional, y "píxel" para las posiciones en una imagen.

$$normal_{01}(s)(i) := s(i)/255 \tag{2.1}$$

Y si, de hecho, queremos que las señales resultantes tomen los valores 0 y 1 en algún punto, podemos aplicar la transformación:

$$normal_{minmax}(s)(i) := \frac{s(i) - min(s)}{max(s) - min(s)} \tag{2.2}$$

siendo:

$$min(s) := \min_{\forall i \in dominio(s)} \{s(i)\} \tag{2.3}$$
$$max(s) := \max_{\forall i \in dominio(s)} \{s(i)\}$$

En la figura 2.3a) se muestran las mismas proyecciones de la figura 2.2 normalizadas con esta transformación. El resultado es equivalente a una compensación previa del brillo de una imagen. Los valores de las tres señales, que antes tomaban intervalos muy dispares, se encuentran ahora en posiciones similares. De esta forma, las señales resultantes se pueden comparar entre sí con más facilidad.

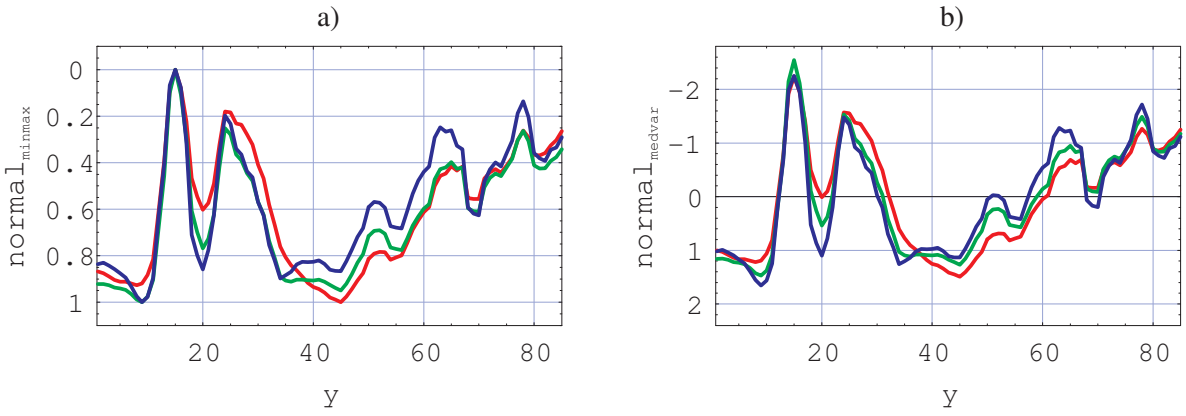


Figura 2.3: *Normalización de señales en el valor. a) Integrales proyectivas de la figura 2.2, normalizadas con la operación* $normal_{minmax}$*. b) Las mismas proyecciones normalizadas con* $normal_{medvar}$*.*

Es más, en muchas situaciones sería preferible disponer de una normalización más robusta, no basada exclusivamente en los valores máximos y mínimos. Esa normalización podría usar la media y la varianza de las señales; de esta forma, el cálculo es menos sensible a la variación esporádica de los picos de la señal. Por ejemplo, para conseguir señales con media 0 y varianza 1 la transformación a aplicar es del tipo:

$$normal_{medvar}(s)(i) := \frac{s(i) - media(s)}{\sqrt{var(s)}} \tag{2.4}$$

donde:

$$media(s) := \frac{1}{s_{max} - s_{min} + 1} \sum_{i=s_{min}}^{s_{max}} s(i) \tag{2.5}$$

$$var(s) := \frac{1}{s_{max} - s_{min} + 1} \sum_{i=s_{min}}^{s_{max}} (s(i) - media(s))^2$$

La figura 2.3b) muestra un ejemplo de esta normalización en el valor, para las mismas señales de la figura 2.3a). En este caso concreto, el resultado es muy parecido con ambas operaciones. No obstante, la normalización con media/varianza es ligeramente mejor para el ejemplo, y en general será siempre más robusta.

Operaciones locales que modifican el valor

Por definición, las transformaciones anteriores están dadas de manera que cada punto de las señales es tratado independientemente de sus vecinos. En analogía a las *convoluciones* y *filtros locales* sobre imágenes, podemos definir operaciones de procesamiento local sobre proyecciones. Estas operaciones se obtienen como simples reducciones al caso 1D de los filtros locales 2D. Podemos hablar, por lo tanto, de transformaciones locales sobre integrales proyectivas.

Definición 2.6 ***Transformación local.***
Una operación de transformación local sobre señales unidimensionales, l, es una función:
$l : \mathbb{S} \to \mathbb{S}$
dada por:
$l(s)(i) := f\left(s(i-k), \ldots, s(i-1), s(i), s(i+1), \ldots, s(i+k)\right)\ ;\ \forall i \in dominio(s)$
para algún $k \in \mathbb{N}$, donde $f : \mathbb{R}^{2k+1} \to \mathbb{R}$, es la función que define la transformación local.

El parámetro k indica el radio de vecindad utilizado. Igual que en las operaciones sobre imágenes, se debe establecer algún *tratamiento especial* para los puntos donde los vecinos de $s(i)$ caen fuera del dominio de la señal. También igual que con imágenes, podemos definir **convoluciones** discretas sobre proyecciones, que serán las transformaciones locales donde f viene dada por una combinación lineal de su entrada.

Una propiedad interesante de la convolución discreta de señales es que resulta **conmutativa** con el proceso de proyección: proyectar una imagen y aplicar luego un filtro de convolución 1D, es equivalente a pasar el mismo filtro 1D a la imagen –lógicamente, a lo largo del mismo eje de la señal–, y después proyectar. Esto no se puede asegurar de cualquier filtro de procesamiento local. Por ejemplo, no se cumple para los filtros de máximo o mínimo local.

Un ejemplo de convolución 1D es el **suavizado** de señales. Un *suavizado de media* de radio k es una transformación del tipo:

$$suaviza_{med}(s)(i) := \frac{1}{2k+1} \sum_{j=-k}^{k} s(i+j) \tag{2.6}$$

Por su parte, en un *suavizado gaussiano* los coeficientes vendrían dados por los valores de una campana de Gauss discretizada. En la figura 2.4 se muestra un ejemplo de suavizado de proyecciones utilizando la operación $suaviza_{med}$.

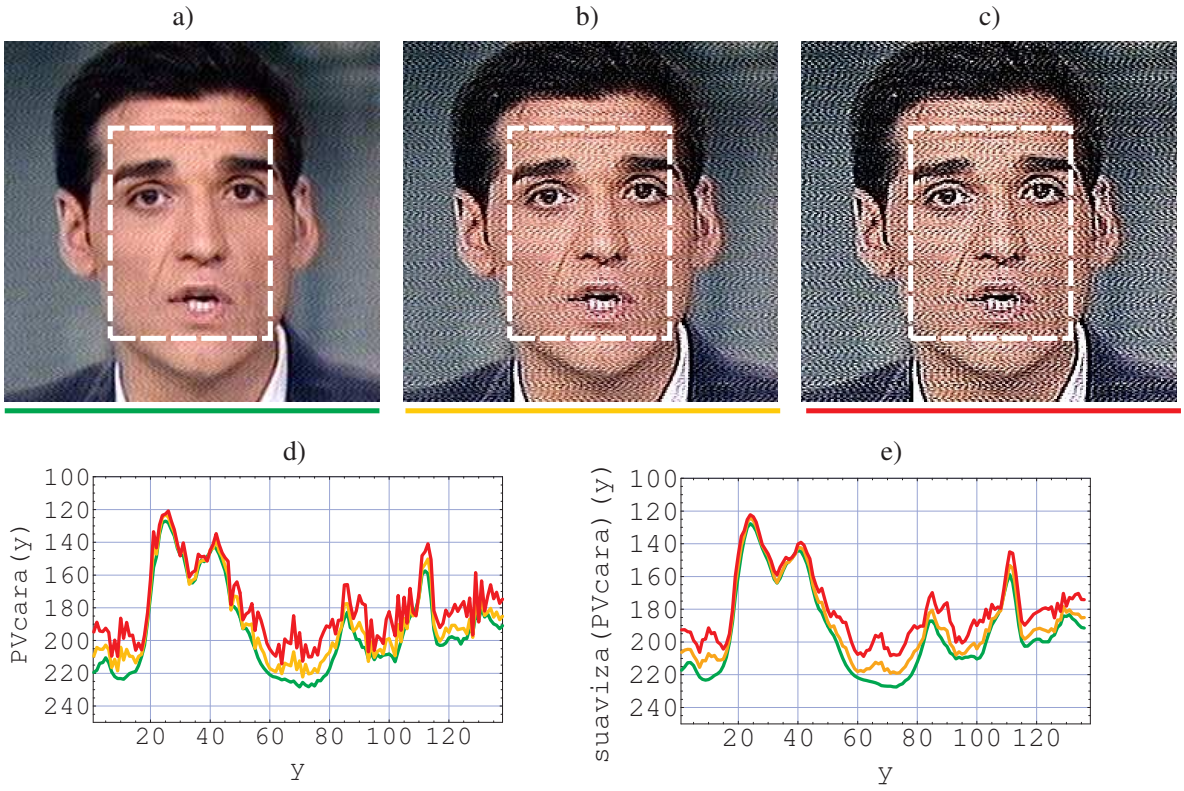


Figura 2.4: *Transformación de suavizado sobre integrales proyectivas. a) Imagen de entrada. Se señala la región sobre la que se aplica la proyección. b) Imagen perfilada al 60 %, es decir, sumando la laplaciana por 0,6. c) Imagen perfilada al 120 %. d) Proyecciones verticales de las caras a) (en verde), b) (en amarillo), y c) (en rojo). e) Las mismas proyecciones después de un suavizado de media de radio $k = 1$.*

Sobre la imagen de la figura 2.4a) –capturada de un canal ruidoso de televisión–, se han aplicado diferentes operaciones de perfilado para aumentar artificialmente el nivel de ruido. La diferencia media (en niveles de gris) entre píxeles vecinos de la figura 2.4c) está por encima de 75. Sin embargo, después de proyectar –figura 2.4d), en rojo– la diferencia media entre puntos adyacentes de la señal se reduce a unos 9 niveles de gris. Y después del suavizado –figura 2.4e), en rojo– baja hasta los 4,3.

En conclusión, el suavizado es útil para *reducir el nivel de ruido* de las señales, eliminando los detalles más finos. Sin embargo, en la práctica, las proyecciones son mucho más inmunes al ruido que trabajar directamente con las imágenes, de manera que este tipo de operaciones es normalmente innecesario. En esencia, la mayor robustez es debida a la propia acumulación de valores que supone la proyección, que actúa como una forma de compensar la perturbación sufrida en los distintos píxeles proyectados. A continuación hacemos un breve inciso para estudiar formalmente esta interesante propiedad.

Inciso: robustez de las proyecciones frente al ruido. Vamos a ver que la *comparación de patrones* es más inmune al ruido usando proyecciones que usando las imágenes. Una de las métricas de comparación más habituales es la *suma de diferencias al cuadrado* (equivalente a la *distancia euclídea* o *norma* L_2). Basándonos en ella, podemos definir una medida de distancia normalizada, d^2, dada por:

$$d^2(a,b) = \frac{1}{n}||a-b||^2 \tag{2.7}$$

donde a y b pueden ser imágenes o señales; n es el número de píxeles o de puntos, respectivamente; y $||.||$ es el módulo del vector correspondiente (la raíz cuadrada de la suma de valores al cuadrado). Aplicamos la normalización $1/n$ para obtener rangos de distancias comparables tanto con imágenes como con proyecciones. Así, si los píxeles toman valores entre 0 y 1, las distancias de ambos estarían también entre 0 y 1.

Supongamos que tenemos una imagen i, de $W \times H$ píxeles, que es perturbada con un ruido aditivo, r, dando lugar a la imagen j:

$$j(x,y) = i(x,y) + r(x,y)\ ;\ \forall x \in \{0,\dots,W-1\}; \forall y \in \{0,\dots,H-1\} \tag{2.8}$$

La distancia entre las imágenes i y j se puede deducir fácilmente:

$$d^2(i,j) = \frac{1}{WH}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(i(x,y)-j(x,y))^2 = \frac{1}{WH}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}r(x,y)^2 \tag{2.9}$$

Supongamos que el ruido, r, es una variable aleatoria con distribución gaussiana de media 0, varianza σ^2, y es independiente para cada píxel; esto es, $r(x,y) \sim \mathcal{N}(0,\sigma^2)$. La esperanza matemática de la ecuación 2.9 será:

$$E(d^2(i,j)) = E\left(\frac{1}{WH}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}r(x,y)^2\right) = \frac{1}{WH}E\left(\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}r(x,y)^2\right) \tag{2.10}$$

Teniendo en cuenta que las $r(x,y)$ son independientes y normales, todas las $E(r(x,y)^2)$ valdrán σ^2, de forma que tenemos:

$$E(d^2(i,j)) = \frac{1}{WH}WH \cdot E\left(r(x,y)^2\right) = \sigma^2 \tag{2.11}$$

Es decir, el error esperado en la métrica de distancia es exactamente igual que la varianza del ruido de los píxeles. Veamos ahora lo que ocurre utilizando integrales proyectivas, por ejemplo, tomando las verticales. Queremos obtener $d^2(PV_i, PV_j)$ siendo:

$$PV_i(y) = \frac{1}{W}\sum_{x=0}^{W-1}i(x,y)\ ;\ PV_j(y) = \frac{1}{W}\sum_{x=0}^{W-1}(i(x,y)+r(x,y)) \tag{2.12}$$

Por lo tanto, la distancia buscada será:

$$d^2(PV_i, PV_j) = \frac{1}{H}\sum_{y=0}^{H-1}\left(PV_i(y) - PV_j(y)\right)^2 = \tag{2.13}$$

$$\frac{1}{H}\sum_{y=0}^{H-1}\left(\frac{1}{W}\sum_{x=0}^{W-1} i(x,y) - \frac{1}{W}\sum_{x=0}^{W-1}(i(x,y)+r(x,y))\right)^2 = \frac{1}{H}\sum_{y=0}^{H-1}\left(\frac{1}{W}\sum_{x=0}^{W-1} r(x,y)\right)^2$$

Sea $m(y) = 1/W \cdot \sum_x r(x,y)$. La variable aleatoria $m(y)$ es una suma de W distribuciones gaussianas $\mathcal{N}(0,\sigma^2)$ independientes; por lo tanto, la suma es también una gaussiana. Claramente, su media es 0 y su varianza es la suma de varianzas multiplicada por el cuadrado del factor $1/W$. En definitiva, $m(y) \sim \mathcal{N}(0,(1/W)^2 W\sigma^2) = \mathcal{N}(0,\sigma^2/W)$. Así, la esperanza matemática de la ecuación 2.13 será:

$$E(d^2(PV_i, PV_j)) = E\left(\frac{1}{H}\sum_{y=0}^{H-1}\left(\frac{1}{W}\sum_{x=0}^{W-1} r(x,y)\right)^2\right) = \frac{1}{H}E\left(\sum_{y=0}^{H-1} m(y)^2\right) \tag{2.14}$$

Como las $m(y)$ son también normales e independientes entre sí, $E(m(y)^2) = \sigma^2/W$. De esta manera la ecuación 2.14 queda:

$$E(d^2(PV_i, PV_j)) = \frac{1}{H}H \cdot E\left(m(y)^2\right) = \sigma^2/W \tag{2.15}$$

En conclusión, la esperanza matemática del error se reduce a σ^2/W, mientras que para las imágenes era de σ^2. Esto significa que el efecto del ruido queda atenuado proporcionalmente al número de píxeles proyectados, W. Por ejemplo, en una imagen de 100×100 píxeles, el error esperado por la introducción del ruido es 100 veces menor con proyecciones que con comparación 2D, y la desviación estándar 10 veces menor. $\therefore$

Este resultado teórico coincide con los datos experimentales del ejemplo de la figura 2.4, de unos 100 píxeles de ancho. En la imagen de la figura 2.4c), la perturbación media del ruido –en este caso, por la aplicación de un perfilado– es de unos 75 niveles de gris; ésta sería la desviación estándar $\sqrt{\sigma^2}$. Con proyecciones, la desviación sería del orden de $\sqrt{\sigma^2/100} \simeq 7,5$ que es muy próximo a la variación observada en la señal de unos 9 niveles de gris.

Podría argumentarse en contra que, si bien el error es menor, la comparación con proyecciones es menos informativa que con patrones 2D. En el apartado 2.1.3 vamos a ver que esto no es así, ya que el proceso de proyección es invertible. Simplemente se debería añadir a la medida de distancia, d, el resultado de distintas proyecciones en diferentes ángulos, por ejemplo, $1/2(d(PV_i, PV_j) + d(PH_i, PH_j))$, sin que esto afecte a la esperanza del error.

Operaciones que modifican el dominio

Las dos anteriores categorías de operaciones modifican exclusivamente el valor de las señales, sin afectar al dominio de las mismas. Empero, si las regiones proyectadas tienen distinto tamaño o posición en las imágenes, será necesario utilizar transformaciones de desplazamiento y escalado de proyecciones. Estas funciones son análogas a las *transformaciones geométricas* sobre imágenes.

Definición 2.7 ***Transformación en el dominio.***

Una operación de transformación en el dominio de señales unidimensionales, g, es una función:

$g : \mathbb{S} \to \mathbb{S}$

definida por:

$g(s)(i) := s(f(i))\ ;\ \forall i \in \{f^{-1}(s_{min}), \ldots, f^{-1}(s_{max})\}$

donde $f : \mathbb{N} \to \mathbb{R}$, es la función que define la transformación en el dominio, y f^{-1} es la función inversa de f.

Por ejemplo, una función del tipo $f(i) = 2i$, serviría para *reducir a la mitad* el tamaño de la señal de entrada, mientras que una de la forma $f(i) = i + d$ produce un desplazamiento en el dominio de d puntos. En general, para conseguir un desplazamiento d y escalado e usaremos la operación:

$$escalado_{de}(s)(i) := s(d + e \cdot i) \tag{2.16}$$

Siendo el dominio resultante $\{(s_{min} - d)/e, \ldots, (s_{max} - d)/e\}$.

En la figura 2.5 se ilustra la operación de escalado, aplicada sobre proyecciones verticales de caras humanas. En este caso se ha utilizado el escalado para hacer coincidir los picos de las señales, correspondientes a cejas, ojos, nariz y boca. Tras la operación, estas características son proyectadas a los mismos puntos de las tres señales resultantes, es decir, están *alineadas*.

Puesto que las señales son discretas y f toma valores en $\mathbb{R}$, surge la necesidad de definir un método de **interpolación** de las proyecciones, esto es, una forma de asignar valores a $s(x)$ para todo x en el intervalo continuo $[s_{min}, \ldots, s_{max}]$. En nuestro caso, una *interpolación lineal* puede resultar suficiente para obtener señales de calidad; el valor interpolado sería: $s(x) := p \cdot s(\lfloor x \rfloor) + (1 - p) \cdot s(\lceil x \rceil)$, con $p = \lceil x \rceil - x$.

También las operaciones de transformación en el dominio son **conmutativas** respecto del proceso de proyección: aplicar una operación de este tipo sobre una proyección, es equivalente a aplicar una transformación geométrica, dada por f, a la imagen –lógicamente, paralela al eje de la señal–, y después obtener la proyección de esa imagen.

Otras operaciones sobre señales unidimensionales

En general, las operaciones de interés sobre proyecciones no sólo afectarán al dominio o al valor de las señales, sino a ambos a la vez. Además, las transformaciones posibles no se limitan a las antes definidas o a combinaciones de las mismas. Por ejemplo, si necesitamos hacer

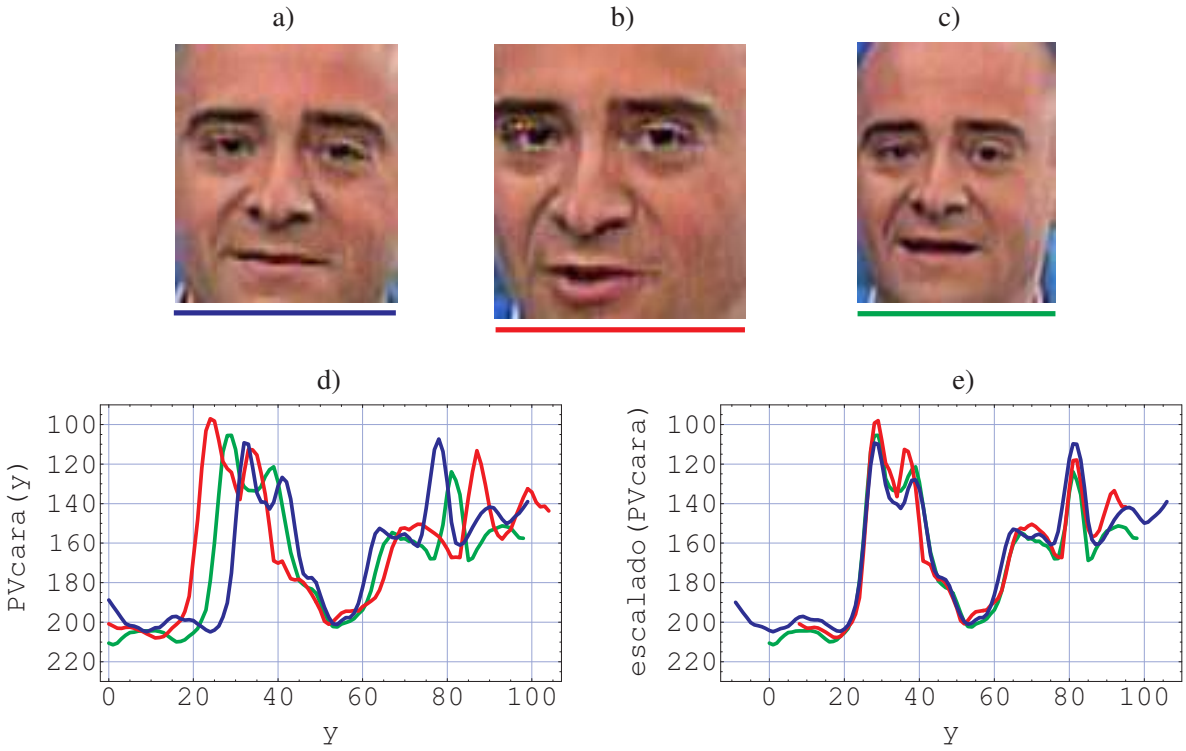


Figura 2.5: *Transformación en el dominio de integrales proyectivas. a, b, c) Imágenes de un presentador de TV, en distintos instantes de una secuencia y con diferentes tamaños. d) Proyecciones verticales del canal R de las tres imágenes anteriores. e) Las mismas proyecciones anteriores, modificadas en el dominio con una operación $escalado_{de}$, para hacer coincidir las características comunes.*

un análisis en el dominio frecuencial podrá tener sentido aplicar transformadas de Fourier sobre las proyecciones; en ese caso, el valor de cada punto resultante es una combinación lineal de toda la señal de entrada. Dentro de la misma categoría entrarían las transformadas del seno, del coseno, de *wavelets*, la descomposición con PCA, etc.; en general, las denominadas *transformaciones lineales*, aplicadas aquí sobre señales 1D.

Definición 2.8 ***Transformación lineal.***

Una operación de transformación lineal sobre señales unidimensionales, h, es una función:
$h : \mathbb{S} \rightarrow \mathbb{S}$
dada por:

$$h(s)(i) := \sum_{j=s_{min}}^{s_{max}} s(j) f(i,j) \; ; \; \forall i \in dominio(s)$$

donde $f : [s_{min}, ..., s_{max}] \times [s_{min}, ..., s_{max}] \rightarrow \mathbb{R}$, es la función que define la transformación lineal.

Por ejemplo, en la transformada de Fourier (DFT) de una señal (obviamente, suponiendo que las señales pueden tomar valores complejos), la función f sería:

$$f(i,j) := e^{-2\pi\sqrt{-1}ij/(s_{max}-s_{min}+1)} \tag{2.17}$$

Existe una relación muy importante entre las integrales proyectivas y las transformadas de Fourier, conocida como el **teorema de la sección central**, [65]. Supongamos que P_α, es la proyección de una imagen, i, en ángulo, α. Según el teorema, la transformada de Fourier de P_α es igual a un segmento de la transformada de Fourier de i, que pasa por el origen y tiene ángulo α. Por ejemplo, la DFT de una proyección vertical es equivalente al segmento vertical, pasando por el origen, de la DFT de la imagen. Vamos a ver a continuación una consecuencia interesante de esta propiedad.

2.1.3. Reproyección de integrales proyectivas

Llamamos *reproyección* a la reconstrucción de una imagen a partir de una o varias integrales proyectivas extraídas de la misma. Podríamos pensar que existe una ambigüedad implícita en el proceso de proyección/reproyección: la integral proyectiva de una imagen en cierta dirección es única, pero muchas imágenes distintas pueden producir la misma señal proyectada.

Sin embargo, la incertidumbre de la reproyección disminuye a medida que se utilizan más proyecciones. Es más, el problema es idéntico a la *reconstrucción tomográfica* a partir de transformadas de Radon (los llamados *sinogramas*), [178]. Por lo tanto, igual que en aquel ámbito, podemos aplicar el teorema de la sección central para garantizar que la reconstrucción del original es posible si se dan las condiciones adecuadas. La demostración es sencilla: si una proyección nos permite obtener un segmento de la transformada de Fourier 2D original en cierto ángulo, un número elevado de proyecciones en ángulos distintos nos permitiría obtener toda la imagen original en el dominio de Fourier.

El teorema de la sección central es la demostración teórica de que la transformación de proyección es **invertible**. Vamos a comprobarlo también de forma práctica, proponiendo un método sencillo y aproximado de reproyección para un conjunto cualquiera de proyecciones.

Algoritmo de reproyección

Supongamos que queremos reproyectar un conjunto arbitrario de integrales proyectivas de una imagen o de regiones de la misma, que denotaremos por $\{P^1, P^2, \ldots, P^n\}$. Básicamente, la reproyección se puede entender como un proceso iterativo, en el que cada una de las señales aporta cierta información que se va plasmando en la imagen resultante.

Consideremos también que se conoce el tamaño de la imagen, $ancho \times alto$, y su nivel medio de gris[5]. Además, para cada proyección, P^k, y en concreto para cada punto de la misma, $P^k(j)$, sabemos los píxeles de la imagen que influyen en ese punto de la señal, que denotaremos por $pts^k(j) = \{(x_1^k(j), y_1^k(j)), (x_2^k(j), y_2^k(j)), ...\}$. Esta información es equivalente a conocer la región y el ángulo de cada integral proyectiva. Por ejemplo, si P^k es la proyección vertical de la imagen, el conjunto de píxeles $pts^k(j)$ sería $\{(0, j), (1, j), (2, j), ..., (ancho - 1, j)\}$.

[5]Hay que observar que estas condiciones son poco restrictivas. Por un lado, el valor medio de la imagen se puede obtener fácilmente si alguna de las proyecciones fue aplicada sobre la imagen completa; y, por otro lado, con dos proyecciones de la imagen en ángulos distintos se puede calcular el tamaño original.

Con todos estos datos, podemos formular el proceso de reproyección de una imagen, que se muestra en el algoritmo 2.1.

REPROYECCION DE UNA IMAGEN
ENTRADA:
- Tamaño de la imagen: $ancho \times alto$, y valor de gris medio: $\overline{i(\cdot,\cdot)}$.
- Proyecciones utilizadas: $\{P^1, P^2, \ldots, P^n\}$.
- Píxeles asociados a cada punto de las proyecciones:
  $pts^k(j) = \{(x_1^k(j), y_1^k(j)), (x_2^k(j), y_2^k(j)), \ldots\}, \forall k \in \{1, \ldots, n\}, \forall j \in dominio(P^k)$.

SALIDA:
- Imagen reproyectada: $\hat{i}$.

ALGORITMO:
Inicialización:
$\hat{i}(x,y) := \overline{i(\cdot,\cdot)}, \forall x \in \{0, \ldots, ancho-1\}, y \in \{0, \ldots, alto-1\}$
Iteración principal:
para $k := 1$ ***hasta*** n ***hacer***
  para $j \in dominio(P^k)$ ***hacer***
    $medVal := \overline{\hat{i}(x,y)}\ ;\ \forall (x,y) \in pts^k(j)$
    para $(x,y) \in pts^k(j)$ ***hacer***
      $\hat{i}(x,y) := \hat{i}(x,y) + P^k(j) - medVal$
    finpara
  finpara
finpara

Algoritmo 2.1: *Reproyección de una imagen a partir de un conjunto de integrales proyectivas.*

La clave del algoritmo es la instrucción más interna de los bucles, encargada de garantizar que la media de los píxeles $pts^k(j)$ en $\hat{i}$ sea precisamente $P^k(j)$, conservando además la información aportada por las proyecciones anteriores. Esto se logra mediante la suma:

$$\hat{i}(x,y) := \hat{i}(x,y) + P^k(j) - medVal \tag{2.18}$$

Donde $medVal$ es la media calculada de los píxeles $pts^k(j)$ en $\hat{i}$.

Resultados de la reproyección

En las figuras 2.6 y 2.7 se muestran varias reproyecciones de dos imágenes, usando distintas cantidades de proyecciones en el algoritmo. Se puede apreciar que, a medida que sube el número de señales, la calidad de la reconstrucción aumenta gradualmente. Los problemas de redondeo y desbordamiento[6] hacen que las imágenes resultantes no lleguen a alcanzar la resolución de la original. Podemos extraer las siguientes conclusiones:

- Obviando los problemas de resolución, queda comprobado de forma práctica que el proceso de proyección es *invertible*: a partir de un número suficiente de proyecciones es posible reconstruir la imagen original mediante reproyección. Esto significa que las

[6]Se han usado imágenes con 1 byte por píxel, por lo que el desbordamiento por debajo de 0 o por encima de 255 puede ser frecuente en los cálculos intermedios.

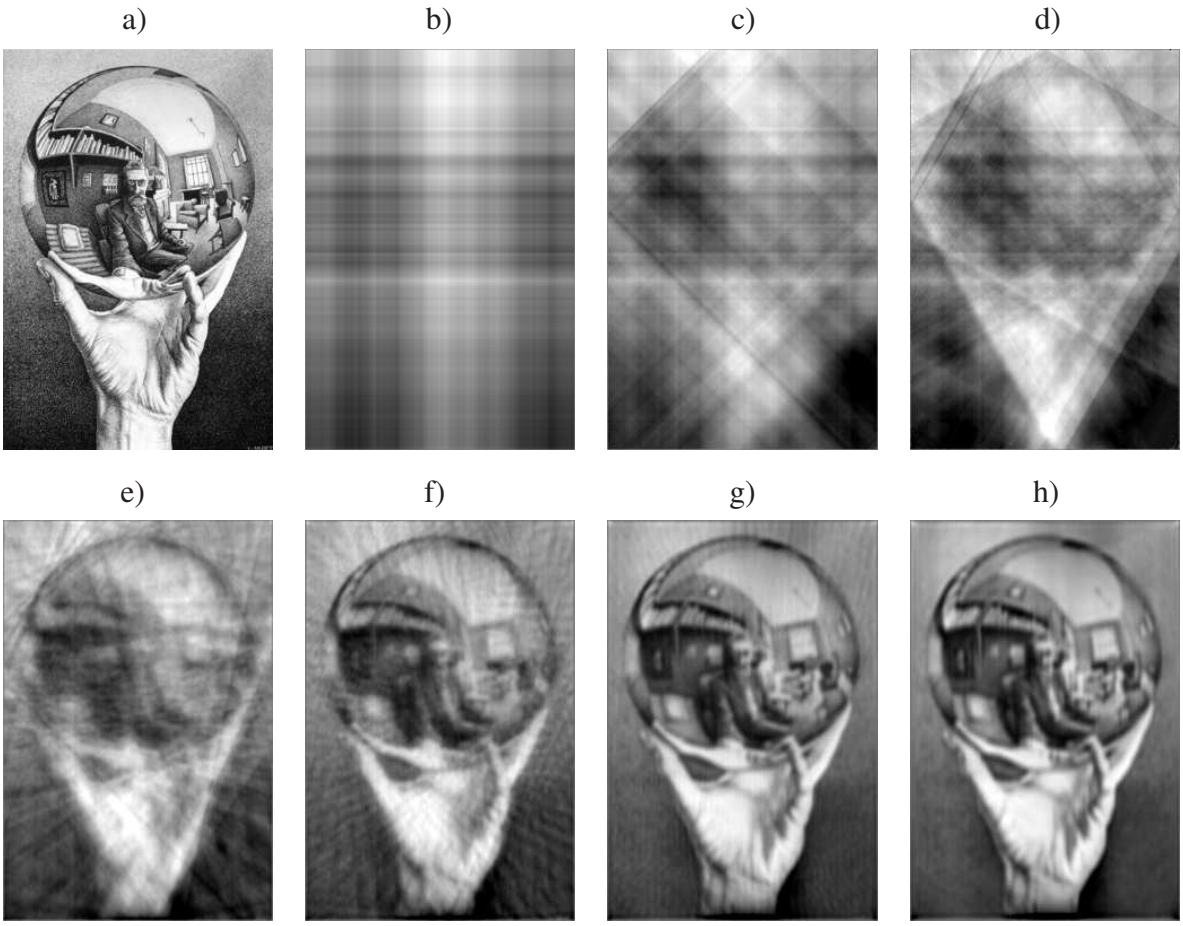


Figura 2.6: *Reproyección de una imagen a partir de integrales proyectivas. a) Imagen de entrada (239×355 píxeles). b) Reproyección con PV y PH. c-h) Reproyección con proyecciones en distintos ángulos repartidos de manera uniforme: c) 4, d) 6, e) 16, f) 40, g) 100, h) 200.*

integrales proyectivas *conservan toda la información* de las imágenes. Es decir, no hay pérdida de información inherente a la proyección, más que la derivada de usar menos integrales que las necesarias. Por ejemplo, la información utilizada para la reproyección de la figura 2.6g) es equivalente, aproximadamente, a una reducción de 1:3 sobre la información de la imagen original.

- De lo anterior se deduce que el uso de las proyecciones en una aplicación será más adecuado cuantas menos señales necesitemos para reconstruir los objetos de interés; o, en otras palabras, cuanta más información se conserve con una pocas proyecciones. En una escena compleja, como la de la figura 2.6, sólo se vislumbra el contenido original a partir de un número alto de proyecciones. Sin embargo, en el caso de la cara humana de la figura 2.7, se necesitan muy pocas para obtener una reconstrucción global de la estructura del rostro. Si tomamos la varianza de los niveles de gris como una medida de conservación de la información, para la figura 2.7a) una sola proyección preserva un 64 % de la información, con 2 proyecciones un 75 %, con 4 el 87 %, y con 8 supera el 92 %.

Es más, vamos a ver más adelante cómo es posible conseguir un modelo viable con sólo dos proyecciones, seleccionando regiones adecuadas de las caras.

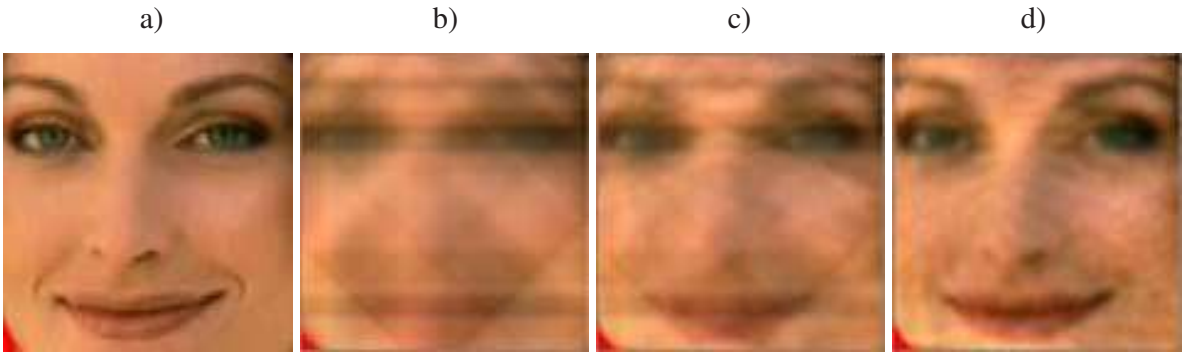


Figura 2.7: *Reproyección de una cara a partir de integrales proyectivas en R, G y B. a) Imagen de entrada (118 × 122 píxeles). b-d) Reproyección con proyecciones de los canales RGB, repartidas en distintos ángulos de manera uniforme: b) 4, c) 8, e) 20.*

El problema de reproyección abordado aquí es una versión discreta y simplificada del cálculo de tomografías a partir de sinogramas, [178], obtenidos típicamente mediante rayos X, ultrasonidos, emisión de positrones o resonancias magnéticas. El método que hemos propuesto es más sencillo que las técnicas que suelen usarse en ese ámbito, porque no hemos utilizado el dominio frecuencial sino que se trabajamos sólo en el dominio espacial. Además, a diferencia de aquellos, hemos considerado la posibilidad de que las proyecciones sean de regiones arbitrarias de las imágenes.

Debemos recordar, por otro lado, que nuestro interés en el uso de proyecciones es el análisis, y no el cálculo de reproyecciones. Seguidamente, vamos a centrarnos en dos cuestiones relacionadas estrechamente con el análisis de imágenes mediante proyecciones: el modelado y el alineamiento de integrales proyectivas.

2.2. Modelos de proyección

Un modelo, en el sentido más amplio, es una descripción genérica de cierta categoría de objetos. Existen modelos de color, modelos basados en patrones, modelos de grafos, etc. El diseño del modelo es fundamental en muchas aplicaciones de visión, porque su forma puede determinar o influir en dos de los principales aspectos de un sistema de percepción[7]: qué características se usan, y qué mecanismo de clasificación se aplica sobre las mismas.

Un *modelo de proyección* describe el conjunto –en principio, no finito– de posibles proyecciones asociadas a cierta clase de objetos, en ángulos y regiones particulares de esos objetos. Existen muchas formas posibles de modelar integrales proyectivas; en principio, tantas como

[7]Hay que observar, sin embargo, que el uso de modelos no es indispensable; de hecho, algunos investigadores evitan la construcción explícita de modelos, centrando su atención en los mecanismos de clasificación, extracción de características, o en el control del proceso de visión.

las aplicables sobre las imágenes. En esta sección vamos a proponer tres alternativas viables para construir modelos de proyección, en los apartados 2.2.2, 2.2.3 y 2.2.4, basadas en distribuciones gaussianas. Estas alternativas serán comparadas cuantitativamente usando un conjunto de caras y no caras, descrito en el apartado 2.2.1. Finalmente, en el apartado 2.2.5 estudiamos cómo combinar varios modelos de proyección para describir objetos 2D.

2.2.1. Criterios y medidas de bondad de un modelo

En general, un buen modelo de objetos debería tener las siguientes características:

- **Entrenable**. El modelo se debe poder entrenar de forma automática a partir de ejemplos, evitando al máximo la intervención de un operador humano. La tarea del humano es diseñar el modelo, pero no dar valores concretos a las variables del mismo. Esto debería aplicarse también a los posibles parámetros del modelo, para los cuales debería existir un método más o menos automático de ajuste. Paradójicamente, no es infrecuente encontrar trabajos que utilizan integrales proyectivas, donde los modelos son definidos *ad hoc* por el humano [93, 101, 199, 169, 170, 132, 59, 50, 213], típicamente a través de umbrales, estructuras de picos máximos y mínimos, o zonas de máxima pendiente, deducidas por simple inspección visual.

- **Generalizable**. El modelo debe ser válido para el mayor número posible de instancias de la clase, y no simplemente para las usadas en el entrenamiento del modelo. El sobreajuste, o sobre-entrenamiento, es uno de los grandes riesgos en el análisis de imágenes mediante modelos. No obstante, ningún mecanismo de clasificación –ya sea basado modelos o no–, está completamente libre de ese problema. Un modelo sobre-ajustado es propicio a provocar malos resultados cuando se aplica a condiciones ligeramente distintas a las de entrenamiento.

- **Distancia**. El modelo debe definir una medida de distancia, o de similitud, entre la categoría de objetos modelados y una nueva instancia. En términos estadísticos, podemos decir que el modelo define –de forma implícita o explícita– una *distribución de probabilidad* de los ejemplos de la clase; la distancia está relacionada inversamente con la función de densidad de probabilidad modelada.

En consecuencia, en el contexto que nos ocupa, las dos primeras cuestiones a resolver para cada técnica de modelado de proyecciones son: (1) cómo *entrenar* el modelo a partir de ejemplos, y (2) cómo obtener la *distancia* de una señal nueva al modelo.

La aplicación de la medida de distancia sobre un conjunto de datos concretos nos permite cuantificar la capacidad de generalización conseguida con cada modelo. En concreto, necesitamos tres conjuntos: instancias de entrenamiento (para crear el modelo), instancias de prueba de la misma clase (sobre las cuales aplicar la distancia al modelo), e instancias de prueba que no sean de la clase (para las cuales la distancia debería ser lo mayor posible). Para todos los

modelos propuestos en los siguientes apartados usaremos los mismos datos, que se muestran esquemáticamente en la figura 2.8.

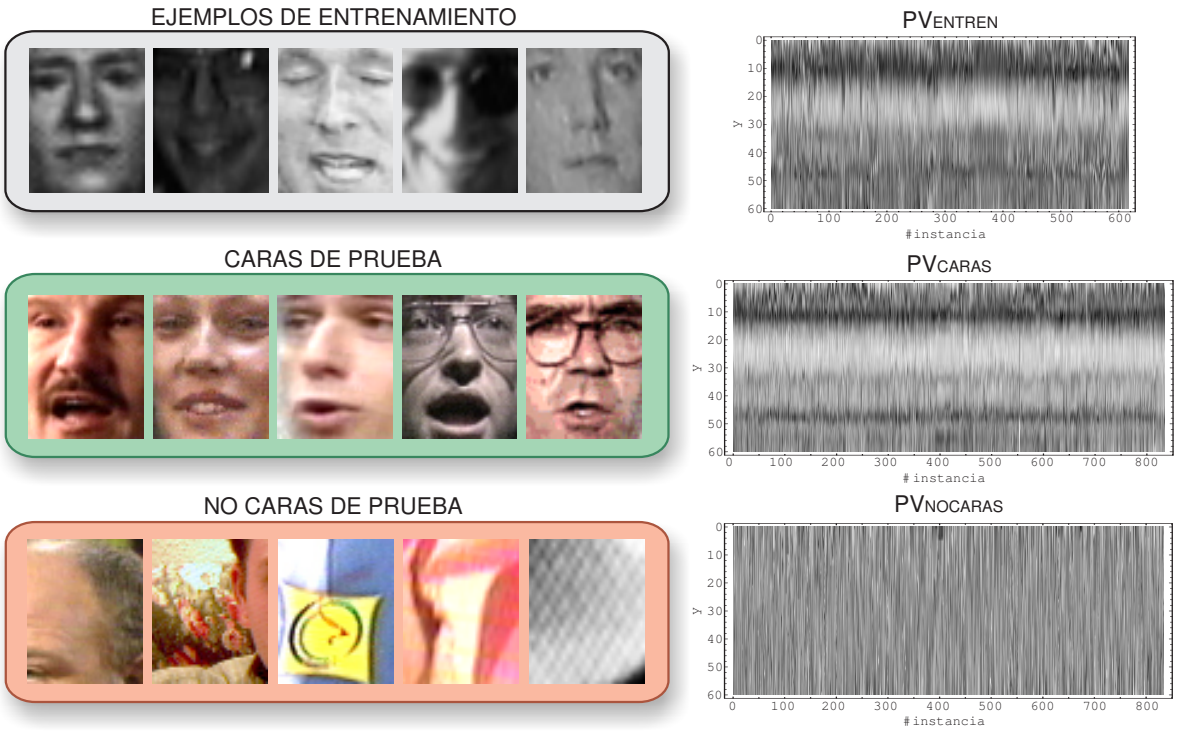


Figura 2.8: *Ejemplos de caras y no caras para el entrenamiento y prueba de los modelos. Fila superior: caras de entrenamiento de la base de caras CMU/MIT. Fila intermedia: caras de prueba de la base de caras UMU. Fila inferior: no caras extraídas aleatoriamente de la base UMU. A la derecha se muestran las proyecciones verticales calculadas para todas las instancias de cara y no cara.*

El conjunto de entrenamiento son 616 caras humanas de la base CMU/MIT [152], en escala de grises. Las instancias de prueba del modelo son 834 caras de la base UMU. Las no caras son 834 posiciones aleatorias de la base UMU. Los rostros han sido extraídos a un rectángulo estándar de 48×60 píxeles. De todas las imágenes se han obtenido las proyecciones verticales, PV_{cara} y PV_{nocara}, y después se han normalizado (con la ecuación 2.4) para tener igual media y varianza. En el caso de las imágenes en color, se proyecta el canal R. Usando estos datos, se toman las siguientes medidas para cuantificar la bondad de los modelos:

1. **dcaras**: distancia media de las caras de prueba al modelo (será mejor cuanto menor sea).

2. **dnocaras**: distancia media de las no caras al modelo (mejor cuanto más grande).

3. **propd**: proporción entre las distancias medias de caras y de no caras (interesa que sea grande, indicando una mejor discriminación clase/no clase).

4. **solap**: solapamiento entre las distancias ocupadas por las caras y las no caras. Es otra medida de separabilidad del modelo, expresada en relación al número de ejemplos. El valor ideal sería 0.

5. **eer**: ratio de error igual (en inglés, *equal error rate*). Error que obtendría un clasificador cara/no-cara por distancia al modelo, fijando el umbral a una posición con igual número de falsos positivos que de falsos negativos.

Conviene dejar claro que con estas pruebas no se trata de proponer o sugerir un método específico de detección de caras. En primer lugar, porque los modelos de proyección serán usados también en el resto de aplicaciones –no sólo en detección, sino también en localización, seguimiento, reconocimiento de personas, etc.–, como veremos en los siguientes capítulos. En segundo lugar, porque un modelo de objetos 2D constará en general de varios modelos de proyección distintos, y no de uno solo. Y, por último, porque un algoritmo de detección de caras no puede suponer conocidas las posiciones de las caras de prueba.

En consecuencia, el objetivo de las medidas introducidas es cuantificar la bondad de las distintas técnicas para el modelado de proyecciones, independientemente de la aplicación posterior del modelo.

2.2.2. Modelos de proyección media

Para hacernos una idea del problema que estamos abordando, en la figura 2.9 mostramos dos conjuntos de integrales proyectivas asociadas a dos clases de objetos. Por un lado, la figura 2.9d) corresponde a las proyecciones verticales de los rostros de la figura 2.9b), seleccionados de entre las caras de entrenamiento descritas en el apartado 2.2.1. Con el fin de generalizar, se incluyen en la figura 2.9c) ejemplos de un problema completamente distinto: patrones de la letra "m", para una posible aplicación de reconocimiento de caracteres.

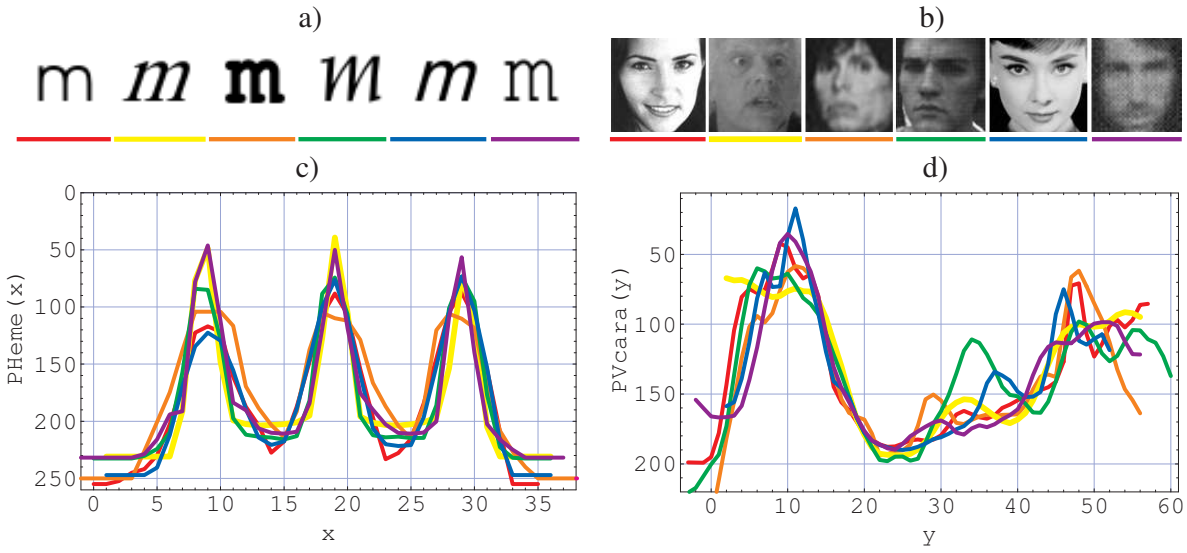


Figura 2.9: *Diferentes proyecciones de dos categorías de objetos. a) Patrones de la letra "m". b) Caras humanas de la base de caras CMU/MIT, [152]. Se muestran abajo los colores usados para representar las proyecciones correspondientes. c) Proyecciones horizontales de los patrones de a), escaladas y normalizadas. d) Proyecciones verticales de las caras de b), escaladas y normalizadas. En este segundo caso, se muestra un trozo un poco mayor que las regiones realmente proyectadas.*

En ambos ejemplos, podemos apreciar que las señales presentan estructuras parecidas de picos y valles, que corresponden, lógicamente, a una estructura interna común de elementos destacados, claros y oscuros; y ello a pesar de que en la figura 2.9b) se han incluido algunos de los casos más variados del conjunto. Las señales han sido normalizadas en el valor y escaladas en el dominio de forma conveniente –con el método de alineamiento que estudiaremos en el apartado 2.3–, para hacer las similitudes más evidentes.

Podemos argumentar, observando los ejemplos de la figura 2.9, que un simple modelo de "patrón medio" puede ser viable para describir un conjunto de proyecciones. Incluso el caso más complejo de las caras de la figura 2.9b), que presentan aspectos tan diversos, produce proyecciones muy próximas a un hipotético patrón medio. Este será, por lo tanto, nuestro primer método de modelado de proyecciones.

Definición 2.9 ***Modelo de proyección media.*** *Un modelo de proyección de media es una señal unidimensional, M, donde $M(i), \forall i \in dominio(M)$, es el valor medio de las señales de entrenamiento en el punto i, y $dominio(M)$ es la intersección de los dominios de la señales de entrenamiento.*

A partir de este modelo es posible dar varias medidas de distancia de un modelo, M, a una proyección, P. En particular, proponemos utilizar la suma de diferencias al cuadrado:

$$dist(M,P) = 1/||r|| \sum_{\forall i \in r} (M(i) - P(i))^2 \tag{2.19}$$

con:

$$r := dominio(M) \cap dominio(P) \tag{2.20}$$

El modelo de proyección media es análogo, en análisis de imágenes, a un modelo de imagen media. A pesar de su aparente simplicidad, los modelos de proyección media funcionan normalmente bastante bien. Es más, vamos a ver que en algunos casos pueden mejorar la sensiblemente capacidad de discriminación de un modelo de imagen equivalente. En la figura 2.10 se muestran los resultados del modelo de proyección media sobre los datos de entrenamiento y de prueba presentados en el apartado 2.2.1.

La separación entre caras y no caras es bastante interesante. La mayoría de los rostros (un 98 %) se encuentra a una distancia menor de 100. Sin embargo, las distancias para las no caras se extienden en un intervalo mucho mayor. Lógicamente, la propia variabilidad de las no caras implica una mayor dispersión de las distancias. Aleatoriamente, algunas proyecciones de no cara pueden tomar forma parecida al rostro medio, produciendo medidas bajas. Aun así, el solapamiento es de sólo el 6,83 %, produciendo un ratio de error igual de 3,41 %.

Comparación con modelo de imagen media

Con el fin de poder comparar resultados, se ha repetido el experimento usando modelos de patrón 2D medio de cara, es decir, la imagen media. Las caras son extraídas con una resolución de 48×60 píxeles, y se normalizan en intensidad para mejorar la medida de distancia.

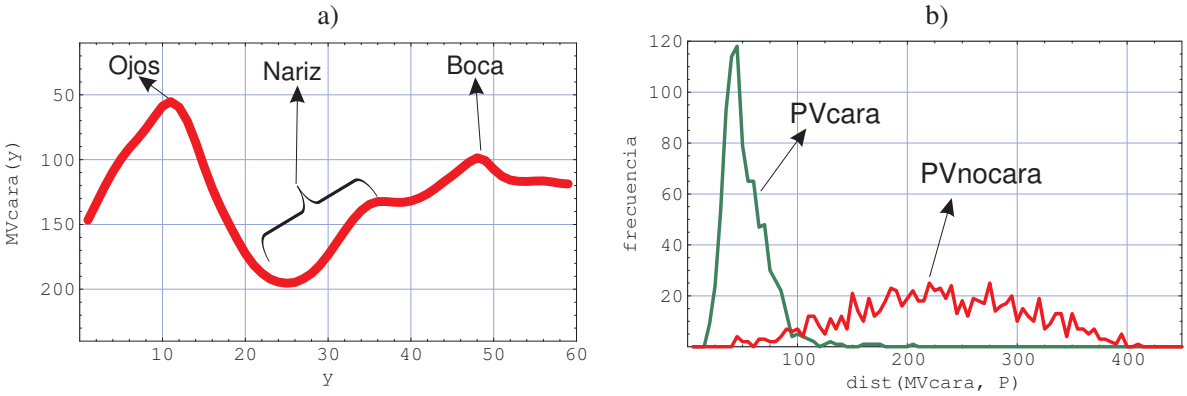


Figura 2.10: *Resultados del modelo proyección vertical media de la cara. a) Modelo creado con las 616 caras de entrenamiento. Se muestra una interpretación de las posiciones en la señal de ojos, nariz y boca. b) Distancias de las 834 caras (en verde) y 834 no caras (en rojo) respecto del modelo. Resultados:* ***dcaras****= 52,9;* ***dnocaras****= 229,4;* ***propd****= 4,33;* ***solap****= 6,83 %;* ***eer****= 3,41 %.*

Utilizamos dos medidas: la suma de diferencias al cuadrado (*difsq*) y la correlación (*corr*), en ambos casos normalizadas. Suponiendo que estamos comparando dos imágenes, i y t, ambas de tamaño $w \times h$, la primera distancia toma la forma:

$$difsq(i,t) = \frac{\sum_{\forall x,y}(i(x,y) - t(x,y))^2}{\sqrt{\sum_{\forall x,y} i(x,y)^2 \cdot \sum_{\forall x,y} t(x,y)^2}} \tag{2.21}$$

Por su parte, la medida de correlación toma valores entre -1 y 1, indicando el valor 1 una mayor similitud al modelo medio[8]. En concreto, está definida como:

$$corr(i,t) = \frac{\sum_{\forall x,y} i'(x,y) \cdot t'(x,y)}{\sqrt{\sum_{\forall x,y} i(x,y)^2 \cdot \sum_{\forall x,y} t(x,y)^2}} \tag{2.22}$$

donde i' y t' son los patrones normalizados para tener media 0. Los resultados obtenidos con estas dos medidas se presentan en la figura 2.11.

La medida de correlación funciona bastante mejor en este caso que la suma de diferencias, produciendo un solapamiento casi 4 veces menor. Por otro lado, en comparación con el modelo 1D de proyección media, los resultados obtenidos con la correlación 2D son muy similares, aunque ligeramente peores. Mientras que con proyecciones el error es del 3,4 %, con imágenes supera por poco el 4 %. Esta diferencia se podría considerar dentro del margen de error del experimento. No obstante, debemos recordar que el modelo de proyección tiene el mérito de trabajar exclusivamente en un espacio 1D. Y, a pesar de la reducción de información que supone la proyección (en este caso, un 1:48), las señales conservan buena parte de la información que permite discriminar caras y no caras.

La medida de correlación se podría aplicar también sobre integrales proyectivas, sin más que reducir la ecuación 2.22 al caso 1D. Sin embargo, curiosamente, en este caso los resultados

[8]Luego aquí un valor alto indica una mayor probabilidad de pertenencia a la clase.

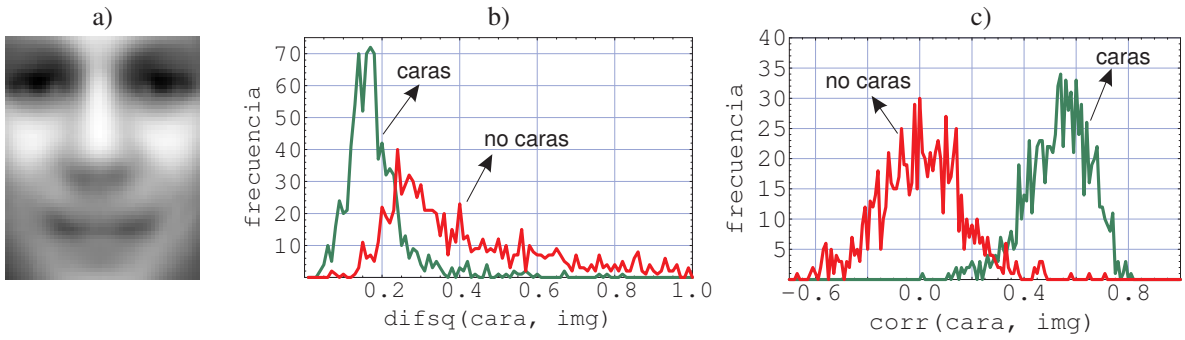


Figura 2.11: *Resultados del modelo de imagen de cara media. a) Patrón medio de las caras de entrenamiento. b) Diferencias al cuadrado, respecto del modelo, de las caras (en verde) y no caras (en rojo) de prueba. Resultados:* ***dcaras****= 0,18;* ***dnocaras****= 0,41;* ***propd****= 2,22;* ***solap****= 30,3 %;* ***eer****= 15,2 %. c) Valores de correlación para los mismos datos de prueba. Resultados:* ***dcaras****= 0,53;* ***dnocaras****= 0,004;* ***solap****= 7,89 %;* ***eer****= 4,02 %.*

obtenidos son sensiblemente peores; el solapamiento asciende hasta el 18,7 % y el ratio de error igual al 12,5 %. Podemos concluir que la correlación funciona mejor con las imágenes, mientras que con proyecciones es más adecuada la suma de diferencias al cuadrado.

2.2.3. Modelos de media/varianza

En el simple modelo de señal media, la distribución de probabilidad de la clase es una hiper-esfera en el espacio de las proyecciones. En otras palabras, la medida de distancia otorga la misma relevancia a todos los puntos de las señales. Pero es lógico pensar que ciertas zonas de las proyecciones presentarán más variabilidad que otras. Por ejemplo, en una secuencia de imágenes de una persona hablando, habrán pocas variaciones en la zona de la nariz y muchas en la boca. En consecuencia, los puntos donde se proyecta la nariz deberían tener una mayor influencia en la medida que los puntos correspondientes a la boca.

Para describir esta variabilidad implícita de las proyecciones, aumentamos el modelo añadiendo la *varianza* observada en cada punto de las señales. De esta forma, se presupone para cada punto de las proyecciones una distribución gaussiana independiente, $\mathcal{N}(\mu, \sigma^2)$. El modelo media/varianza de integrales proyectivas queda definido de la siguiente manera.

Definición 2.10 ***Modelo de media/varianza.*** *Un modelo de proyección de media/varianza es un par de señales,* (M, V)*, con* $dominio(M) = dominio(V)$*, donde:*

- $M(i), \forall i \in dominio(M)$*, es el valor medio de las señales modeladas en el punto* i*.*
- $V(i), \forall i \in dominio(V)$*, es la varianza de las señales en el punto* i*.*

La forma de hacer el entrenamiento se deriva directamente de la definición del modelo, y se detalla en el algoritmo 2.2. Suponiendo que el conjunto de proyecciones a modelar es $C_{entr} = \{P^1, P^2, \ldots, P^n\}$, el algoritmo 2.2 simplemente calcula la media y la varianza en cada punto. Puesto que el dominio de cada proyección de C_{entr} puede ser distinto, el dominio del modelo será la intersección de los asociados a las instancias.

ENTRENAMIENTO DE UN MODELO DE PROYECCIÓN MEDIA/VARIANZA
ENTRADA:
- Conjunto de proyecciones de entrenamiento: $C_{entr} = \{P^1, P^2, \ldots, P^n\}$

SALIDA:
- Modelo de proyección: (M, V).

ALGORITMO:
Inicialización:
$dominio(M) := dominio(P^1) \cap dominio(P^2) \cap \ldots \cap dominio(P^n)$
$dominio(V) := dominio(M)$
Iteración principal:
para $i \in dominio(M)$ ***hacer***
$M(i) := 1/n \sum_{k=1}^{n} P^k(i)$
$V(i) := 1/n \sum_{k=1}^{n} (P^k(i) - M(i))^2$
finpara

Algoritmo 2.2: *Entrenamiento de un modelo de proyección media/varianza.*

La definición de la medida de distancia también se deduce de manera lógica del modelo gaussiano. Sea (M, V) el modelo de proyección y P una instancia nueva, la distancia señal/modelo vendría dada por:

$$dist((M, V), P) := \frac{1}{||r||} \sum_{i \in r} \frac{(M(i) - P(i))^2}{V(i)} \tag{2.23}$$

con:

$$r := dominio(M) \cap dominio(P) \tag{2.24}$$

Igual que antes, la medida de distancia debería producir un valor bajo, próximo a cero, para las proyecciones de la misma categoría que el modelo y un valor alto para las demás proyecciones. Al dividir por la varianza, todos los valores obtenidos serán ahora menores. Los resultados de este modelo sobre el caso de las proyecciones verticales de caras humanas se muestran en la figura 2.12.

Resulta destacable la similitud con los resultados presentados en la figura 2.10, no sólo en cuanto a los parámetros resultantes sino también en la forma de los histogramas. Esto se debe a que, en este ejemplo concreto, la varianza aporta poca información sobre las zonas de mayor o menor variabilidad. En la figura 2.12a) se puede ver que la varianza es bastante uniforme a lo largo de toda la proyección vertical de cara. Consecuentemente, el aumento del modelo no supone una mejora de los resultados, sino más bien un ligero empeoramiento. A pesar de ello, los porcentajes están dentro de un margen admisible, comparable también con los obtenidos con el modelo 2D de cara media. Además, no debemos olvidar que en un caso donde las varianzas sean más significativas, el beneficio puede ser importante.

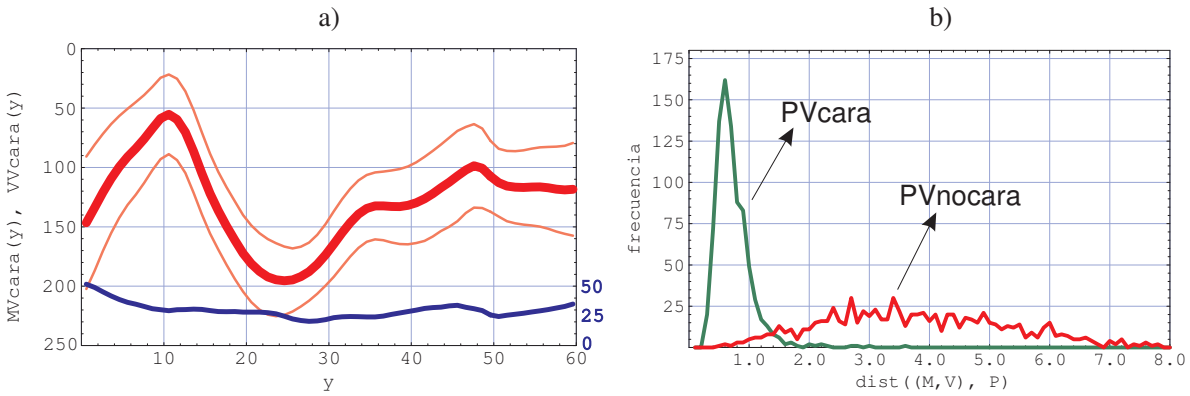


Figura 2.12: *Resultados del modelo de media/varianza de proyección vertical de la cara. a) Modelo creado con las 616 caras de entrenamiento, media (en rojo) y desviación estándar (en azul). b) Distancias de las 834 caras (en verde) y 834 no caras (en rojo), respecto del modelo. Resultados:* ***dcaras****= 0,73;* ***dnocaras****= 3,79;* ***propd****= 5,16;* ***solap****= 7,79 %;* ***eer****= 3,90 %.*

2.2.4. Modelos de media/covarianzas

El modelo media/varianza, introducido en el anterior apartado, supone una distribución de probabilidad normal e independiente para cada uno de los puntos de las señales. Sin embargo, es razonable esperar que exista cierta relación entre los puntos adyacentes de las proyecciones. En ese caso, tendría más sentido utilizar una gaussiana multivariable en n dimensiones, que n modelos gaussianos unidimensionales. Así, en lugar de la varianza, trabajamos con la matriz de covarianzas entre las distribuciones de los puntos. En definitiva, la definición del modelo de proyección media/covarianzas sería la siguiente.

Definición 2.11 ***Modelo de media/covarianzas.*** *Un modelo de proyección media/covarianzas es un par, (M, Σ), donde:*

- *M es una señal unidimensional, y $M(i), \forall i \in dominio(M)$, es el valor medio de las señales modeladas en el punto i.*
- *Σ es una matriz bidimensional de tamaño $dominio(M) \times dominio(M)$; y $\Sigma(i, j)$ es la covarianza de los puntos i y j de las señales, $\forall i, j \in dominio(M)$.*

El algoritmo para entrenar el modelo a partir de ejemplos es sencillo, y se reduce a un simple cálculo de una matriz de covarianzas del conjunto de instancias de entrenamiento, C_{entr}. Por completión, se muestra el pseudocódigo en el algoritmo 2.3.

En este caso, la diferencia entre una señal y un modelo es una distancia de Mahalanobis, que utiliza la inversa de la matriz de covarianzas, Σ^{-1}. En concreto, considerando las proyecciones como vectores columna, la distancia de la señal P al modelo (M, Σ) sería:

$$dist((M, \Sigma), P) := (P - M)^{\mathrm{T}} \cdot \Sigma^{-1} \cdot (P - M) \tag{2.25}$$

donde "$\cdot$" es el producto escalar.

ENTRENAMIENTO DE UN MODELO DE PROYECCIÓN MEDIA/COVARIANZAS
ENTRADA:
- Conjunto de proyecciones de entrenamiento: $C_{entr} = \{P^1, P^2, \ldots, P^n\}$

SALIDA:
- Modelo de proyección: (M, Σ).

ALGORITMO:
Inicialización:
$dominio(M) := dominio(P^1) \cap dominio(P^2) \cap \ldots \cap dominio(P^n)$
$\Sigma :=$ matriz de tamaño $dominio(M) \times dominio(M)$
Cálculo de la media:
para $i \in dominio(M)$ ***hacer***
$M(i) := 1/n \sum_{j=1}^{n} P^j(i)$
finpara
Cálculo de las covarianzas:
para $i \in dominio(M)$ ***hacer***
para $j \in dominio(M)$ ***hacer***
$\Sigma(i,j) := 1/n \sum_{k=1}^{n} (P^k(i) - M(i))(P^k(j) - M(j))$
finpara
finpara

Algoritmo 2.3: *Entrenamiento de un modelo de proyección media/covarianzas.*

En la figura 2.13a) se muestra gráficamente la matriz de covarianzas de las proyecciones de entrenamiento. Las distancias resultantes se pueden ver en la figura 2.13c).

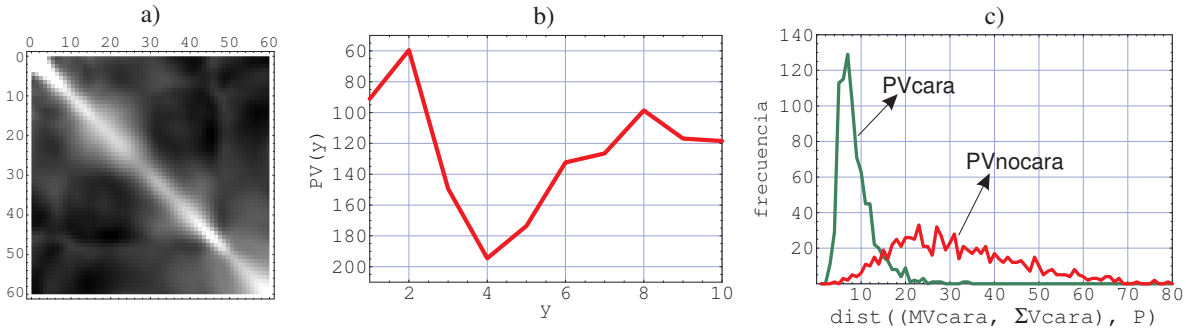


Figura 2.13: *Resultados del modelo de media/covarianzas de proyección vertical de la cara. a) Matriz de covarianzas del modelo creado con las 616 caras de entrenamiento. b) Proyección media de cara utilizada (de tamaño 10). c) Distancias de las 834 caras (en verde) y 834 no caras (en rojo), respecto del modelo. Resultados:* ***dcaras****= 7,7;* ***dnocaras****= 31,1;* ***propd****= 4,03;* ***solap****= 16,4 %;* ***eer****= 8,2 %.*

En este caso, las proyecciones son reducidas previamente de tamaño. Puesto que existe mucha dependencia lineal entre los puntos consecutivos de las señales, la matriz de covarianzas está mal condicionada y no es invertible[9]. En consecuencia, se aplica un escalado de las señales, que son reducidas a 10 puntos –que es el tamaño que produce mejores resultados–.

Los porcentajes obtenidos con el modelo de media/covarianzas son mucho peores que con los modelos más simples. La distancia media para las no caras es unas 4 veces mayor que

[9]Hemos podido comprobar que la regularización de Σ –esto es, tomar $\Sigma + \varepsilon \mathbf{I}$, donde $\mathbf{I}$ es la matriz identidad y ε es una constante real–, tampoco ayuda mucho a mejorar los resultados.

para las caras, pero el solapamiento entre las clases es mucho más grande. En concreto, para el ejemplo de la figura 2.13 supera el 16 %. Esto da lugar a un error por encima del 8 %, cuando en los demás casos no alcanza el 4 %.

Comparación de los modelos de proyección

La tabla 2.1 resume los resultados obtenidos para los distintos modelos de proyección; se incluye también el modelo de patrones 2D con las dos medidas de distancia. Se ha destacado en negrita el método con mejor resultado para los valores de **propd**, **solap** y **err**.

Modelo	dcaras	dnocaras	propd	solap	err
Patrón 2D medio (*difsq*)	0,18	0,41	2,22	30,3 %	15,2 %
Patrón 2D medio (*corr*)	0,53	0,004	-	7,89 %	4,02 %
Proyección media	52,9	229,4	4,33	**6,83 %**	**3,41 %**
Proy. media/varianza	0,73	3,79	**5,16**	7,79 %	3,90 %
Proy. media/covarianzas	7,7	31,1	4,03	16,4 %	8,20 %

Tabla 2.1: *Resultados de los distintos métodos de modelado.* ***dcaras****: distancia media de las caras.* ***dnocaras****: distancia media de las no caras.* ***propd****: proporción entre dnocaras y dcaras.* ***solap****: % de solapamiento de las clases.* ***err****: ratio de error igual.*

Si excluimos el modelo media/covarianzas y el modelo de patrones 2D con suma de diferencias al cuadrado, vemos que todos los restantes obtienen resultados muy parecidos –que podrían encontrarse fácilmente dentro de los márgenes de error de las medidas–. Pero es muy significativo que los modelos de proyección estén al nivel, e incluso mejoren ligeramente, el modelo de patrón 2D medio con correlación. Y ello a pesar de que la comparación no es completamente justa, puesto que la proyección vertical del rostro es sólo una parte del modelo de caras (como veremos en el siguiente apartado). Es decir, estamos comparando un modelo 2D con un modelo 1D.

Desde un punto de vista más amplio, podríamos referirnos a las ideas introducidas por Vapnik [185], en relación al principio de *minimización del riesgo estructural*: un modelo no es mejor cuanto más complejo sea, ya que está amenazado por un mayor riesgo de sobre-ajuste a los datos de entrenamiento. Los resultados parecen confirmarlo claramente: (1) el modelo más simple, el de proyección media –que consta de 60 valores–, produce los mejores resultados; (2) el modelo de media/varianza –que añade otros 60 parámetros–, aumenta la distancia entre clases aunque también el solapamiento; (3) el modelo de media/covarianzas –con una matriz de 100 covarianzas–, presenta una capacidad de generalización muy limitada; incluso una simple proyección media de tamaño 10 produce mejores porcentajes que este método (en concreto, un 8,7 % de solapamiento).

Debemos indicar que se ha usado clasificación por distancia con patrones medios –de imágenes y de proyecciones– por motivos de simplicidad. Posiblemente, los resultados del test cara/no-cara se podrían mejorar con otros tipos de clasificadores (como redes neuronales, AdaBoost o SVM), o extrayendo otras características (como proyección en autoespacios con

PCA o filtros de Haar). Pero, lógicamente, esos mismos clasificadores –y también esos métodos de extracción de características– podrían aplicarse con proyecciones, mejorando igualmente sus resultados. No profundizaremos aquí en la comparativa entre clasificadores y métodos de extracción de características, sino que lo haremos en las aplicaciones concretas en el ámbito del procesamiento facial, que abordamos en los capítulos sucesivos.

2.2.5. Modelado de objetos mediante proyecciones

En los anteriores apartados de esta sección hemos propuesto varias técnicas para el modelado de integrales proyectivas individuales. Un modelo de objetos 2D constará de uno o más de estos modelos de proyección, cada uno de ellos definido sobre una parte de los objetos[10] y en determinado ángulo de proyección. Por lo tanto, surgen dos aspectos básicos en la definición del modelo: (1) la posición *estándar* de los objetos en las imágenes de entrenamiento, y (2) las proyecciones aplicadas sobre esas imágenes. Tanto uno como otro dependen de la aplicación y serán determinados, normalmente, por el diseñador del sistema.

Teniendo en cuenta todas estas cuestiones, podemos dar la siguiente definición para los modelos de objetos basados en integrales proyectivas.

Definición 2.12 ***Modelado de objetos 2D mediante proyecciones.***
Un modelo de objetos 2D mediante integrales proyectivas consta de los siguientes elementos:

- ***Tamaño*** *de las imágenes de entrenamiento:* $w \times h$.
- ***Localización*** *estándar de los objetos de interés dentro de las imágenes.*
- ***Número de modelos*** *de proyección usados:* n. *Para cada modelo* $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n\}$:
  - ***Región*** *sobre la que se aplica la proyección.*
  - ***Ángulo*** *de proyección.*
  - ***Canal*** *que se proyecta, en el caso de imágenes multicanal.*
  - ***Tipo*** *de modelo de proyección (alguno de los anteriores u otro posible modelo).*

Por ejemplo, un modelo viable para los patrones de la letra "m", como los de la figura 2.9a), podría constar básicamente de un modelo de proyección horizontal más otro de proyección vertical de los patrones. Para que el modelo sea completo, debemos fijar también el tamaño de las imágenes y la posición de la letra en las mismas. En la figura 2.14 se muestra este modelo de proyecciones, entrenado con los patrones indicados.

Considerando los resultados del apartado 2.1.3, un modelo de objetos que incluyera suficientes proyecciones para reconstruir las imágenes proyectadas, sería equivalente a un modelo de patrones 2D de las imágenes. Esto es, si la reproyección del modelo se asemeja al patrón

[10] Obsérvese que, por el momento, suponemos que en las imágenes con las que trabajamos los objetos están bien segmentados y localizados. Estamos hablando de las imágenes para el *entrenamiento* del modelo. Evidentemente, esa suposición no se puede hacer al abordar la detección, la localización o el seguimiento.

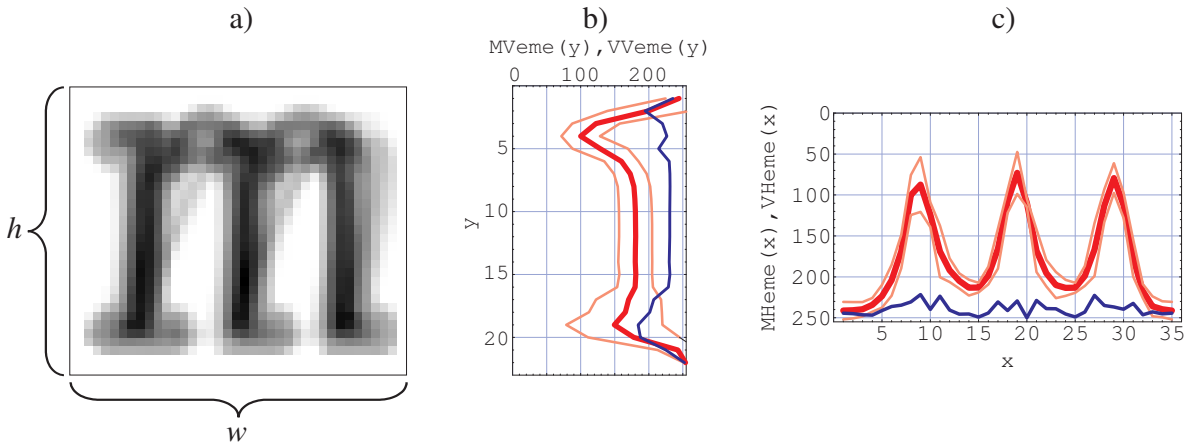


Figura 2.14: *Modelo de integrales proyectivas de la letra "m". a) Imagen media, usando los patrones de la figura 2.9a). El tamaño es de 35 × 23 píxeles. b) Modelo de proyección vertical del patrón. c) Modelo de proyección horizontal del patrón. Media en rojo y desviación estándar en azul.*

2D medio, la suma de distancias a los modelos de proyección sería equivalente a la diferencia con el patrón de imagen media. En cierto sentido, podemos decir que el modelo de proyecciones es un *modelo 1,5D*; cuantas más proyecciones, más próximo será a un modelo 2D. Sin embargo, hay al menos dos motivos para limitar el número de proyecciones usadas: primero, que como hemos visto, un solo modelo de integral proyectiva puede mejorar, por sí solo, la capacidad de generalización de un modelo de patrones 2D; y segundo, que el aumento de complejidad no necesariamente mejora el modelo, como también hemos justificado.

En consecuencia, un modelo de objetos 2D no será mejor cuantos más modelos de proyección incluya, sino cuanto mejor contribuyan esos componentes a la separación clase/no-clase. De hecho, en la práctica, es preferible un modelo que contenga un número muy reducido de proyecciones, pero suficiente para describir lo que hay de común en los objetos. Además, un modelo con pocas proyecciones reduce el riesgo de sobre-ajuste.

Podemos analizar los modelos combinados en función de los mismos criterios introducidos en el apartado 2.2.1: posibilidad de ser entrenados, capacidad de generalizar, y definición de una medida de distancia. Los dos primeros aspectos se derivan de los modelos de proyección subyacentes. El tercero supone la definición de una forma de combinar las distancias correspondientes a cada modelo.

Modelado de caras mediante integrales proyectivas

En este punto vamos a centrarnos en el dominio del procesamiento facial. Proponemos un método de modelado para las caras humanas usando proyecciones. El diseño del modelo intenta extraer la información relevante de los rostros, tomando el menor número posible de proyecciones. Este modelo –de forma completa o en parte–, es usado en los problemas descritos en los restantes capítulos de la tesis. Debemos recordar que los valores concretos del modelo se aprenden mediante entrenamiento; aquí fijamos solamente la forma del modelo.

Vamos a discutir, punto por punto, cada uno de los elementos del modelo presentes en la definición 2.12.

- **Tamaño de los patrones de cara**

  El tamaño de los patrones extraídos de cara determina dos aspectos: la resolución y la proporción ancho/alto. La resolución óptima puede variar en distintas aplicaciones. En un problema de detección interesa una resolución baja (que permita encontrar caras de tamaño reducido), mientras que en localización de componentes puede ser conveniente aumentar la resolución para mejorar la precisión de los puntos localizados. Por su parte, la proporción ancho/alto viene dada por la relación de aspecto del rostro –o, más exactamente, de la parte que nos interesa de las caras–. Normalmente, trabajaremos con relaciones del tipo 4:5, y resoluciones de 24×30 píxeles ó 48×60 píxeles.

- **Posición estándar de las caras en el patrón**

  Existen muchas formas posibles de fijar la posición de las caras en un patrón de tamaño estándar. La manera de establecerlo, a su vez, está relacionada con la forma de extraer una cara de una imagen. Podemos introducir tres suposiciones razonables:

  1. caras se extraen mediante transformaciones *similares* (es decir, transformaciones afines de traslación, rotación y escalado igual en X y en Y);
  2. las caras deben estar centradas verticalmente en el patrón; y
  3. los dos ojos deben estar a la misma altura.

  La primera suposición significa que hay 4 parámetros libres en el problema de extraer las caras, mientras que las otras dos suposiciones fijan un parámetro cada una. En consecuencia, la posición se puede determinar de forma no ambigua con sólo 2 parámetros. Proponemos los siguientes: y_{ojos} = posición en el eje Y de los ojos; y_{boca} = posición en Y de la boca. Valores típicos de estos parámetros son: $y_{ojos} = 0,2h$, $y_{boca} = 0,8h$, siendo h la altura de las imágenes. Modificándolos podemos conseguir que se vea más o menos parte de la frente, de la barbilla, o de los lados del rostro.

- **Modelos de proyección**

  Como hemos mostrado en los ejemplos previos, la proyección vertical del rostro resulta bastante discriminante en una clasificación cara/no cara. Esto es debido a la típica estructura de zonas claras y oscuras de una cara media: la frente (zona clara), cejas-ojos (oscura), nariz (clara), boca (oscura). Esa proyección, por lo tanto, será parte del modelo. En las imágenes en color, proyectaremos el canal R.

  También resulta bastante informativa la proyección horizontal de la región de los ojos, donde se distinguen normalmente los ojos (más oscuros) del entrecejo/nariz (más claro). Finalmente, podemos añadir la proyección horizontal de la boca, que se espera que sea

más oscura que su entorno. En definitiva, tenemos las siguientes proyecciones con los modelos media/varianza correspondientes:

- **Proyección vertical de la cara:** $PV_{cara} \rightarrow (MV_{cara}, VV_{cara})$. Se proyecta todo el patrón de imagen, de $w \times h$ píxeles.
- **Proyección horizontal de los ojos:** $PH_{ojos} \rightarrow (MH_{ojos}, VH_{ojos})$. Se define la región *ojos* como el rectángulo del patrón con esquinas $(0, y_{ojosmin}) \leftrightarrow (w-1, y_{ojosmax})$. Obsérvese que la región está parametrizada por $y_{ojosmin}$ y $y_{ojosmax}$. En los ejemplos se han asignado los valores $0,1h$ y $0,26h$, respectivamente.
- **Proyección horizontal de la boca:** $PH_{boca} \rightarrow (MH_{boca}, VH_{boca})$. La región *boca* está definida por el rectángulo $(0, y_{bocamin}) \leftrightarrow (w-1, y_{bocamax})$. Unos valores comunes son del tipo: $y_{bocamin} = 0,65h$, $y_{bocamax} = 0,95h$.

En la figura 2.15 se interpretan gráficamente los parámetros del modelo de caras propuesto, y se muestra una realización concreta del modelo, creada con los datos de entrenamiento del apartado 2.2.1.

Utilizando también los datos de prueba del mismo conjunto, se han repetido los experimentos aplicados antes sobre las PV_{cara} a las nuevas proyecciones del modelo: PH_{ojos} y PH_{boca}. Los modelos son del tipo proyección media/varianza, y los resultados se pueden observar en la figura 2.15d-f).

De los resultados expuestos en la figura 2.15 podemos deducir que tanto las proyecciones verticales de las caras como las proyecciones horizontales de los ojos son bastante discriminantes en el test cara/no cara. No obstante, los resultados son algo peores para PH_{ojos}, con casi el doble de ratio de error igual. Sin embargo, un caso bien distinto es la proyección horizontal de la boca, que demuestra una capacidad prácticamente nula para distinguir los rostros. Es muy sintomático el hecho de que las varianzas del modelo de PH_{boca} –en azul en la figura 2.15c)–, sean muy superiores a las de PV_{cara} y PH_{ojos}, ya de por sí altas para el conjunto CMU/MIT. Lógicamente, la boca presenta una variabilidad muchísimo mayor que la región de los ojos, y un modelo basado en la media resulta completamente inviable.

Descartando los resultados de PH_{boca}, el modelo de caras propuesto ofrece dos medidas de distancia de un fragmento de imagen a la clase cara. Podemos combinar estas medidas para conseguir una discriminación más fiable que usando cada modelo de proyección por separado. En la figura 2.16 se muestran las dos distancias obtenidas para las caras y no caras de prueba. En base a estos resultados, se propone una combinación mediante la suma ponderada de distancias. Como en los ejemplos previos, se exponen en la figura 2.16b) los resultados de una hipotética clasificación por distancia al modelo.

El método de discriminación cara/no cara establecido de esta manera es equivalente a un clasificador lineal en el plano 2D definido por ambas distancias, que hemos denotado por $dist(MV_{cara})$ y $dist(MH_{ojos})$. La ponderación usada, $dist(MV_{cara}) + 0,4dist(MH_{ojos})$, se ha obtenido por simple ensayo y error.

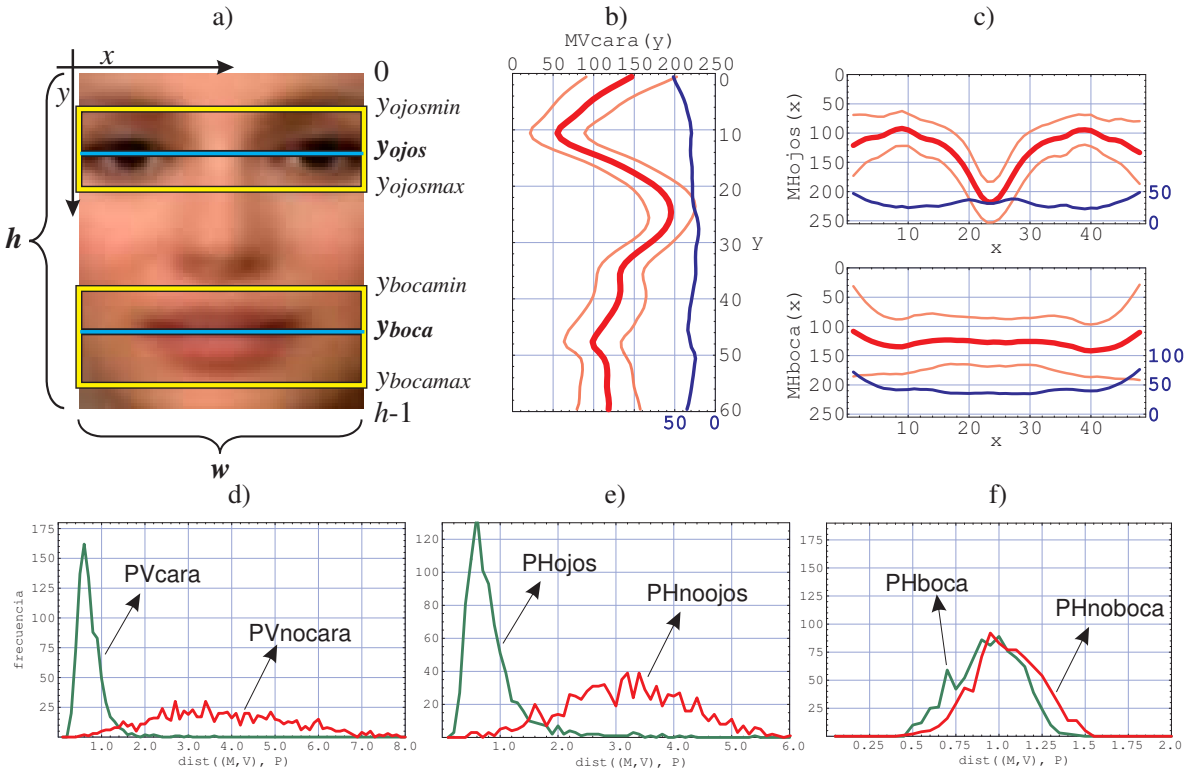


Figura 2.15: *Parámetros y modelo de caras de integrales proyectivas. a) Parámetros genéricos del modelo de caras. b,c) Modelos de proyección vertical de la cara, y proyección horizontal de los ojos y la boca, creados con los datos de entrenamiento detallados en el apartado 2.2.1 (en rojo la media y en azul la desviación estándar). d) Distancias al modelo de PV_{cara} de los ejemplos de cara y no cara. Resultados (ver figura 2.12):* **dcaras**= *0,73;* **dnocaras**= *3,79;* **propd**= *5,16;* **solap**= *7,79 %;* **eer**= *3,90 %. e) Distancias al modelo de PH_{ojos} de los ejemplos de cara y no cara. Resultados:* **dcaras**= *0,86;* **dnocaras**= *3,18;* **propd**= *3,7;* **solap**= *13,3 %;* **eer**= *7,40 %. f) Distancias al modelo de PH_{boca} de los ejemplos de cara y no cara. Resultados:* **dcaras**= *0,94;* **dnocaras**= *1,04;* **propd**= *1,11;* **solap**= *80,0 %;* **eer**= *41,1 %.*

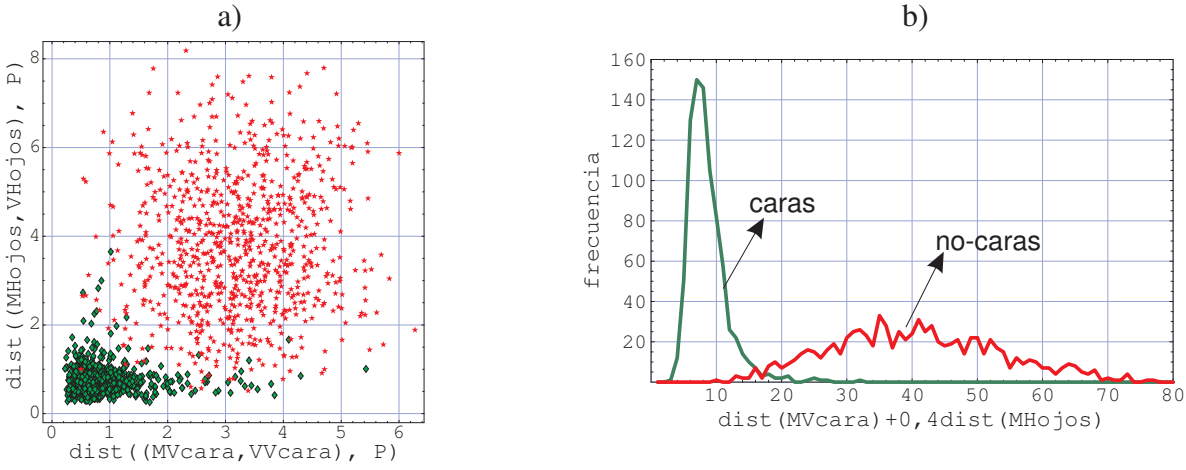


Figura 2.16: *Combinación de distancias a los modelos de proyección. a) Distancias de los patrones de cara (en verde) y no cara (en rojo), a los modelos de proyección vertical de cara y proyección horizontal de ojos. b) Distancias combinadas de los ejemplos de cara y no cara. La combinación es $dist(MV_{cara}) + 0,4dist(MH_{ojos})$. Resultados:* **dcaras**= *1,08;* **dnocaras**= *5,06;* **propd**= *4,69;* **solap**= *4,08 %;* **eer**= *2,04 %.*

En la tabla 2.2 se resumen los resultados del método combinado, comparados con los alcanzados por los distintos tipos de proyecciones definidos en el modelo de cara.

Proyección/modelo	dcaras	dnocaras	propd	solap	err
PV_{cara}	0,73	3,79	**5,16**	7,79 %	3,90 %
PH_{ojos}	0,86	3,18	3,70	13,3 %	7,40 %
PH_{boca}	0,94	1,04	1,11	80,0 %	41,1 %
$PV_{cara} + 0,4PH_{ojos}$	1,08	5,06	4,69	**4,08 %**	**2,04 %**
Patrón 2D medio (*corr*)	0,53	0,004	-	7,89 %	4,02 %

Tabla 2.2: *Comparación de distancias para las distintas proyecciones del modelo de caras. Se muestran también los resultados del método combinado y del método de correlación, para poder contrastar.* ***dcaras****: distancia media de las caras.* ***dnocaras****: distancia media de las no caras.* ***propd****: proporción entre dnocaras y dcaras.* ***solap****: % de solapamiento de las clases.* ***err****: ratio de error igual.*

Incluso con la sencilla combinación de medidas utilizada, el modelo de proyecciones mejora sensiblemente los resultados de los modelos individuales. Es más, respecto del modelo de imagen media se consigue reducir el error a la mitad, pasando del 4 % a poco más del 2 %. Esto conduce a una conclusión interesante: las integrales proyectivas mejoran la capacidad de generalización de los modelos de imagen media. Es decir, la reducción de información que supone el proceso de proyección, no sólo no empeora la descripción de la clase cara, sino que la mejora sustancialmente, al menos para este caso concreto.

Evidentemente, existen otros posibles métodos de combinar las medidas y de ajustar los parámetros. No profundizaremos más aquí, porque la clasificación cara/no cara –al menos tal y como está planteada en estos experimentos– no es un objetivo de aplicación inmediata.

Reproyección del modelo de caras mediante proyecciones

Para concluir esta sección, vamos a ver cómo la reproyección de un modelo de caras –o de cualquier otra clase de objetos, en general– nos permite apreciar la estructura de la clase descrita por ese modelo de proyecciones. Un buen modelo debe producir una reproyección similar a un patrón medio de la clase, mientras que lo contrario podría ser un indicio de que el modelado mediante integrales proyectivas no es adecuado para el problema.

El modelo manejado aquí ha sido obtenido de un subconjunto de 45 caras de la base UMU. Para este experimento se han usado caras con mayor resolución y calidad que las del anterior apartado, aunque manteniendo una alta variabilidad en el número de individuos diferentes. En concreto, las regiones extraídas de los rostros son de 122×77 píxeles. En la figura 2.17 se presenta el modelo resultante y la reproyección del mismo.

El cálculo de la reproyección, en este caso, es un poco especial. Primero, porque la imagen resultante está saturada a blanco y a negro. Y segundo, porque las señales de ojos y boca se reproyectan a toda la mitad superior e inferior[11], respectivamente.

[11] En lugar de reproyectarse sólo a las regiones de las que, teóricamente, se han obtenido. Por ejemplo, la nariz no interviene en PH_{boca}, pero en la reproyección de esa región sí se ha usado MH_{boca}.

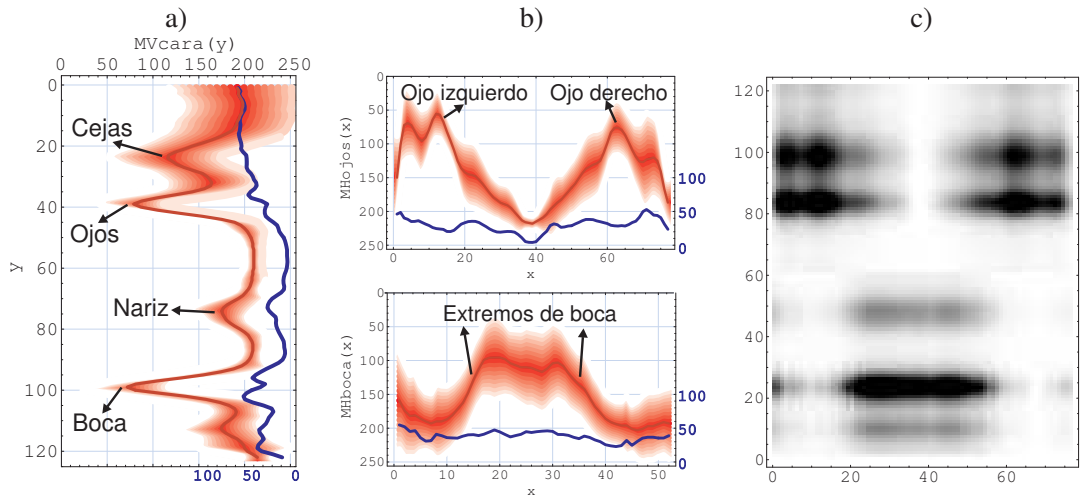


Figura 2.17: *Modelo de caras mediante proyecciones y reproyección del modelo. a,b) Modelos de proyección vertical de la cara, y proyección horizontal de los ojos y la boca, creados con 45 caras de la base de UMU (en rojo la media y en azul la desviación estándar). c) Reproyección del modelo de caras. Se ha ajustado el brillo para saturar a negro y a blanco.*

El hecho de que las características faciales aparezcan en PV_{cara} más claramente destacadas que en el modelo de la figura 2.15, se debe a la mayor resolución de entrada. Pero el resultado realmente interesante es que, incluso con un número tan reducido de proyecciones, se obtiene una reconstrucción bastante verosímil de una cara media. Se pueden distinguir fácilmente en la figura 2.17c) todos los principales componentes faciales: cejas, ojos, nariz y boca.

La capacidad de obtener buenas reconstrucciones es una de las claves para garantizar el buen funcionamiento de las proyecciones con el objeto cara: la propia estructura del rostro humano se puede representar adecuadamente mediante un número muy reducido de integrales proyectivas. En consecuencia, podemos decir que existe una buena justificación empírica para el uso de proyecciones en este ámbito de aplicación concreto.

2.3. Alineamiento de proyecciones

El alineamiento es uno de problemas clave cuando trabajamos con integrales proyectivas. No se puede construir un buen modelo si las proyecciones de los objetos no están correctamente alineadas. Y, a su vez, un modelo de proyección –y especialmente uno que utilice la media de las señales–, fallará cuando se aplique sobre proyecciones de la misma categoría de objetos pero no alineadas correctamente. En la figura 2.18 se ilustra gráficamente por qué es tan importante el alineamiento. Aunque las caras están bien centradas en las imágenes, las proyecciones verticales muestran una gran diferencia entre sí, como se ve en la figura 2.18b). Sin embargo, después de alinear se aprecia claramente la estructura facial común.

En esta sección vamos a abordar el alineamiento de integrales proyectivas, independientemente de su aplicación posterior. En primer lugar formulamos el alineamiento en términos matemáticos como un problema de optimización. A continuación proponemos un algoritmo

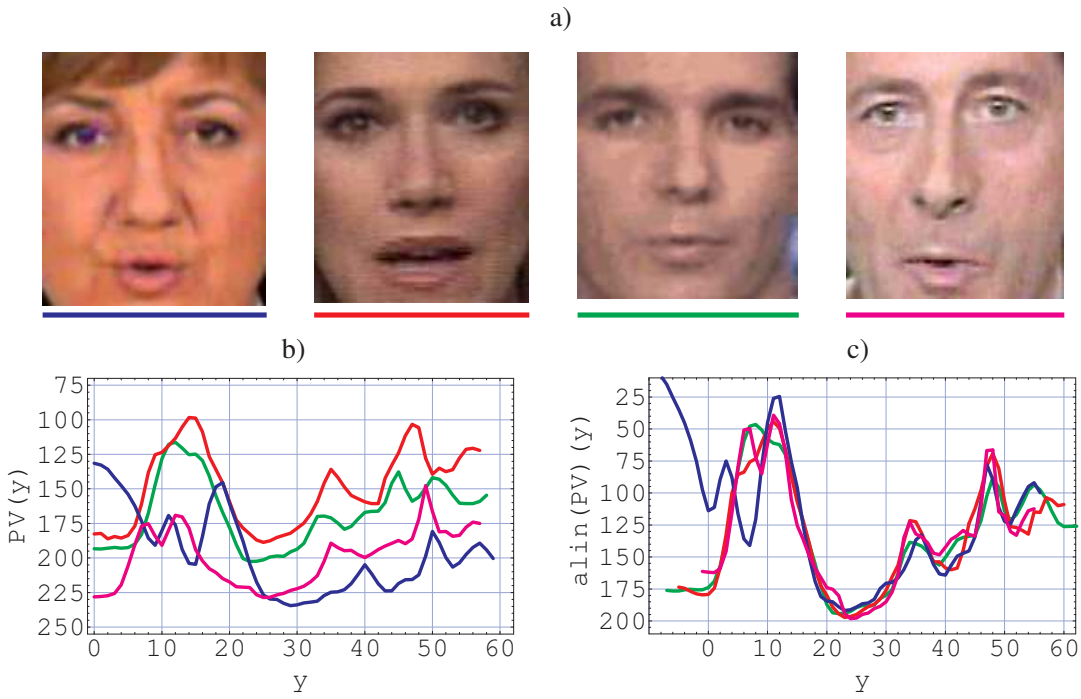


Figura 2.18: *Integrales proyectivas de caras, antes y después del alineamiento. a) Ejemplos de caras de presentadores de TV, tomadas de la base de caras UMU. b) Proyecciones verticales del canal R de las caras. c) Las mismas proyecciones, después del alineamiento. El alineamiento es realizado respecto de un modelo media/varianza.*

eficiente y robusto para el alineamiento de dos señales, o el alineamiento de una señal respecto de un modelo. El diseño cuidadoso de este algoritmo es fundamental, pues será la base de muchas de las aplicaciones descritas en los siguientes capítulos.

2.3.1. Funciones y criterios de alineamiento

En los términos introducidos en el apartado 2.1.2, el alineamiento, o normalización, es una función de transformación sobre integrales proyectivas que garantiza cierta propiedad en las señales resultantes. Podemos definirlo formalmente de la siguiente manera.

Definición 2.13 ***Función de alineamiento.***

Una función de alineamiento, o normalización, de integrales proyectivas es una función:

$n : \mathbb{S} \rightarrow \mathbb{S}$

tal que todas las señales normalizadas cumplen cierta propiedad, $q : \mathbb{S} \rightarrow \mathbb{B}$. Es decir, $q(n(s))$ es cierto para cualquier señal $s \in \mathbb{S}$. Decimos que n normaliza la propiedad q.

Por ejemplo, la operación $normal_{medvar}$ definida en la fórmula 2.4 normaliza las señales a valores con media 0 y la varianza 1. En consecuencia, el alineamiento siempre estará definido en función de una propiedad, el *objetivo de la normalización*. Según ese objetivo, la técnica de alineamiento tomará una u otra forma. En general, estamos interesados en métodos más potentes, que afecten no sólo al valor de las señales sino también al dominio de las mismas.

Centrándonos en el alineamiento de integrales proyectivas para el procesamiento y análisis de imágenes, proponemos los dos siguientes criterios para una buena función objetivo:

- **Criterio de invarianza.** Idealmente, una buena función de alineamiento debería producir proyecciones invariantes a las condiciones de captura. Por ejemplo, en el caso de las caras humanas, las señales deberían ser robustas frente a cambios de iluminación, expresión facial, sexo, edad y otras características de la persona, existencia de elementos faciales adicionales, posición 3D respecto de la cámara, etc. Está claro que algunos de estos objetivos son muy difíciles de alcanzar. Pero podemos concretar dos criterios específicos derivados de este criterio general:
  - **Invarianza en el valor.** Las proyecciones resultantes deben tomar los valores iguales, o equiparables, independientemente de las condiciones de iluminación y el brillo de las imágenes capturadas.
  - **Invarianza en el dominio.** Las características correspondientes deben situarse en los mismos puntos de las proyecciones alineadas. Al calcular las proyecciones, las distintas imágenes de objetos de una misma clase pueden tener posiciones y tamaños diferentes –figura 2.18b)–. Pero, después del alineamiento, las señales se deben modificar en el dominio para que las características comunes se proyecten a las mismas posiciones de las señales –figura 2.18c)–.
- **Criterio de distancia.** Si consideramos que el alineamiento se realiza respecto de un modelo de proyección, podemos formalizar el problema de manera más precisa. De esta forma, el objetivo del alineamiento sería minimizar la distancia de las instancias de la clase al modelo, y maximizar la distancia para las señales que no son de la clase.

Del primer criterio, el de invarianza, podemos deducir una familia de transformaciones cuya aplicación permite la normalización en el valor y en el dominio de las señales. Por su parte, del segundo criterio, el de distancia, se deducirá un método para ajustar los parámetros concretos de la transformación dentro de esa familia.

2.3.2. Formulación del problema de alineamiento

Proponemos la siguiente familia de transformaciones sobre integrales proyectivas:

$$t_{abcde} : \mathbb{S} \to \mathbb{S} \tag{2.26}$$

definida por:

$$t_{abcde}(s)(i) := a + b \cdot i + c \cdot s(d + e \cdot i),\ \forall i \in \{(s_{min} - d)/e, \ldots, (s_{max} - d)/e\} \tag{2.27}$$

Podemos entender esta operación como una *transformación afín sobre proyecciones*, tanto en el valor como en el dominio de las señales: permite un desplazamiento y escalado en el valor

(parámetros a y c), y en el dominio (d y e), y una inclinación (o *shear*, en inglés) en el valor (b). En la figura 2.19 se muestra una interpretación gráfica de la operación propuesta.

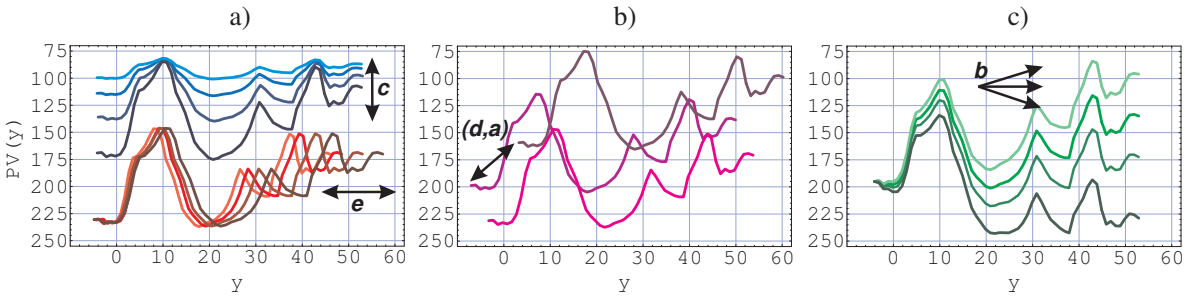


Figura 2.19: *Operación de transformación afín sobre proyecciones. Interpretación de los parámetros (a, b, c, d, e) de la fórmula 2.27. a) Escala en el valor, c, y en el dominio, e. b) Desplazamiento en el valor, a, y en el dominio, d. c) Inclinación, b.*

Todos los términos de la transformación tienen un equivalente en el espacio original de las imágenes, como se describe en la tabla 2.3. En consecuencia, aplicando la operación con los parámetros correctos es posible conseguir un alineamiento robusto frente a los factores descritos en la tabla: cambio de brillo, de posición, de contraste, aparición de sombras, etc.

Parámetro	Significado	Equivalente en imágenes
a	Desplazamiento en el valor	Aumentar o reducir el nivel de luminosidad de las imágenes de forma global
b	Inclinación en el valor	Compensación de la luminosidad en la imagen usando un modelo de fondo lineal, por ejemplo, una sombra o un gradiente claro/oscuro
c	Escala en el valor	Aumentar o reducir el contraste de las imágenes de forma global
d	Desplazamiento en el dominio	Desplazar la posición de las imágenes, a lo largo del eje de proyección
e	Escala en el dominio	Aplicar un escalado de aumento o reducción de las imágenes

Tabla 2.3: *Parámetros de las transformaciones afines sobre proyecciones, $t_{abcde} : \mathbb{S} \to \mathbb{S}$. Ver la definición de t_{abcde} en la ecuación 2.27.*

Basándonos en la familia de transformaciones t_{abcde}, podemos dar una formulación matemática precisa para el problema del alineamiento. Suponiendo un modelo de media/varianza, (M, V), y una proyección, P, el alineamiento consiste en encontrar los parámetros (a, b, c, d, e) que minimicen la expresión:

$$dist((M, V), t_{abcde}(P)) \tag{2.28}$$

Es decir, buscar:

$$\{a^*, b^*, c^*, d^*, e^*\} = \arg \min_{a,b,c,d,e} \frac{1}{||r||} \sum_{i \in r} \frac{(M(i) - t_{abcde}(P)(i))^2}{V(i)} \tag{2.29}$$

con:

$$r := dominio(M) \cap dominio(t_{abcde}(P)) \tag{2.30}$$

Esta formulación incluye también el alineamiento respecto de un modelo de proyección media[12], sin más que considerar que $V(i) = 1$ para todo i.

Desafortunadamente, no es sencillo encontrar una solución cerrada para la fórmula 2.29. El hecho de que haya una transformación en el dominio, y lo que es más, que esa transformación sea discreta, hace pensar que esa solución cerrada no existe. Sí que existe solución cerrada considerando exclusivamente los parámetros en el valor, (a, b, c); de hecho, su aplicación será muy útil para resolver el problema general. Veamos cómo obtener esa solución.

Alineamiento de la señal en el valor

Si suponemos que los parámetros (d, e) de la transformación afín sobre integrales proyectivas son fijos, la ecuación 2.29 se puede plantear como un simple problema de optimización cuadrática. Supongamos, sin pérdida de generalidad, que d vale 0 y e vale 1. La ecuación 2.29 puede escribirse como:

$$\{a^*, b^*, c^*\} = \arg \min_{a,b,c} \sum_{i \in r} \frac{(M(i) - (a + b \cdot i + c \cdot P(i)))^2}{V(i)} \tag{2.31}$$

Nótese que el factor $1/||r||$ ha sido eliminado, puesto que no influye en la obtención del mínimo. La solución puede calcularse igualando a 0 las derivadas parciales respecto de a, b y c, o equivalentemente, resolviendo por mínimos cuadrados el sistema de ecuaciones:

$$\begin{aligned} (a + b \cdot i_1 + c \cdot P(i_1))/V(i_1) &= M(i_1)/V(i_1) \\ (a + b \cdot i_2 + c \cdot P(i_2))/V(i_2) &= M(i_2)/V(i_2) \\ &\dots \\ (a + b \cdot i_m + c \cdot P(i_m))/V(i_m) &= M(i_m)/V(i_m) \end{aligned} \tag{2.32}$$

Para todo $i_j \in r$. El factor $1/V(i_j)$ aparece en ambos lados de las ecuaciones; este término tiene el sentido de ponderar la importancia de cada punto en el ajuste de mínimos cuadrados, por lo que no debe ser eliminado de las ecuaciones. En los puntos del modelo con varianza grande, $1/V(i_j)$ será pequeño y la ecuación correspondiente influirá poco en el ajuste de mínimos cuadrados. Por el contrario, en los puntos con varianza pequeña, $1/V(i_j)$ será grande, dando una mayor importancia al ajuste de ese punto en la solución global.

Expresando el sistema de ecuaciones 2.32 de forma matricial, tenemos:

[12] Y por ende, también el alineamiento de dos proyecciones entre sí.

$$A \begin{pmatrix} a \\ b \\ c \end{pmatrix} = B \tag{2.33}$$

con:

$$A = \begin{pmatrix} 1/V(i_1) & i_1/V(i_1) & P(i_1)/V(i_1) \\ 1/V(i_2) & i_1/V(i_1) & P(i_2)/V(i_2) \\ \dots & \dots & \dots \\ 1/V(i_m) & i_m/V(i_m) & P(i_m)/V(i_m) \end{pmatrix};\ B = \begin{pmatrix} M(i_1)/V(i_1) \\ M(i_2)/V(i_2) \\ \dots \\ M(i_m)/V(i_m) \end{pmatrix} \tag{2.34}$$

La solución para los incógnitas, (a, b, c), se puede obtener fácilmente por *pseudoinversa* aplicando la fórmula:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \left(A^{\mathrm{T}} A\right)^{-1} A^{\mathrm{T}} B \tag{2.35}$$

Los cálculos necesarios para resolver este sistema no son excesivamente costosos, computacionalmente hablando. El producto $A^{\mathrm{T}}A$ requiere $3m^2$ multiplicaciones y sumas; la inversa es de una simple matriz de 3×3; el producto $A^{\mathrm{T}}B$ usa $3m$ operaciones; y el producto de $(A^{\mathrm{T}}A)^{-1}$ por $A^{\mathrm{T}}B$ añade otras 9. En definitiva, necesitamos $O(m^2)$ operaciones en punto flotante; m es el tamaño de las proyecciones, y normalmente no pasará de 60 ó 70.

Distancia mínima de alineamiento en el valor

En algunos casos, como veremos más adelante, podemos estar interesados en conocer la distancia mínima de la señal normalizada al modelo, independientemente de cuánto valgan a, b y c. De esta forma, definimos una nueva medida de distancia señal/modelo –alternativa a la distancia de la ecuación 2.23–, basada en la solución de mínimos cuadrados de la ecuación 2.35. Es decir, es una distancia posterior a una normalización de la señal en el valor. La nueva distancia sería:

$$mindist((M, V), P, d, e) := dist((M, V), t_{abcde}(P)) \tag{2.36}$$

con (a, b, c) calculados según la ecuación 2.35, usando (M, V) y la señal $escala_{de}(P)$.

Para un modelo de proyección media, la definición de *mindist* sería equivalente a suponer que V es una señal constante. Y, a su vez, ese caso es análogo a un problema de alineamiento de dos proyecciones entre sí.

2.3.3. Algoritmo rápido de alineamiento de proyecciones

El alineamiento de proyecciones en el dominio es equivalente al problema de localizar –o *registrar*– objetos en imágenes. Sin embargo, el uso de integrales proyectivas supone una significativa reducción en los grados de libertad del problema. Permitiendo únicamente traslación y escalado, pasamos de cuatro grados de libertad en imágenes a dos en las proyecciones.

El problema de alineamiento en el dominio

Recordemos que nuestra función de alineamiento consta de 5 parámetros, $t_{abcde} : \mathbb{S} \rightarrow \mathbb{S}$. Supongamos que los parámetros (a, b, c), de transformación en el valor, se resuelven como se describe en la ecuación 2.35. El problema del alineamiento, formulado en la ecuación 2.29, se puede reducir a encontrar los valores (d, e) que verifican:

$$\{d^*, e^*\} = \arg \min_{\forall d,e} mindist\left((M, V), P, d, e\right) \tag{2.37}$$

De esta forma, hemos descompuesto el alineamiento en dos subproblemas: encontrar los parámetros de transformación en el valor, (a, b, c), y encontrar los parámetros en el dominio, (d, e). Pero mientras que el primero tiene una solución en forma cerrada, para el segundo parece inviable una solución analítica. Por un lado, porque P es una función escalonada, y por lo tanto no derivable. Por otro lado, porque el escalado es una transformación discreta que hace uso de interpolaciones, como ya hemos comentado.

Es más, si tenemos en cuenta el factor $1/||r||$ en la ecuación 2.29, podemos argumentar que habrá una solución óptima cuando $||r||$ valga uno[13]. Sin embargo, tal situación es sólo un caso degenerado del problema, que se deriva de un desplazamiento o una escala inadmisibles[14]. De ahí se deduce que, normalmente, existirá un intervalo de valores válidos para d y para e. Así, buscamos la solución de la ecuación 2.37 con las restricciones: $d \in [d_{min}, \ldots, d_{max}]$, $e \in [e_{min}, \ldots, e_{max}]$. Por ejemplo, $e_{min} = 0,5$ y $e_{max} = 2$ significa que el alineamiento puede escalar como máximo las señales al doble de su tamaño, y como mínimo a la mitad.

En definitiva, tenemos que encontrar el mínimo de cierta función objetivo en una región acotada de un plano bidimensional; las dos dimensiones del plano son el desplazamiento, d, y la escala, e, de las señales alineadas. En la figura 2.20 se ilustra el problema con una proyección y un modelo concretos. Las señales tienen formas parecidas, aunque con valores y dominios distintos. Aplicando desplazamientos y escalados sobre la proyección (en verde) obtenemos diferentes distancias señal/modelo, como se aprecia en la figura 2.20b); el punto óptimo sería el mínimo de esta gráfica. La señal alineada con ese par, (d, e), aparece en la figura 2.20c).

[13] En ese caso, sólo habrá un punto en la intersección entre $dominio(M)$ y $dominio(t_{abcde}(P))$. Eso significa que el ajuste en el valor hará coincidir ese punto de P con el de M, con lo que $mindist\left((M, V), P, d, e\right)$ valdrá 0.

[14] Por poner un ejemplo, si a una señal de tamaño 30 se le aplica una escala $e = 30$, la proyección resultante tiene tamaño 1. Su distancia de alineamiento sería 0. Pero esa solución será, con toda seguridad, inadecuada para el problema.

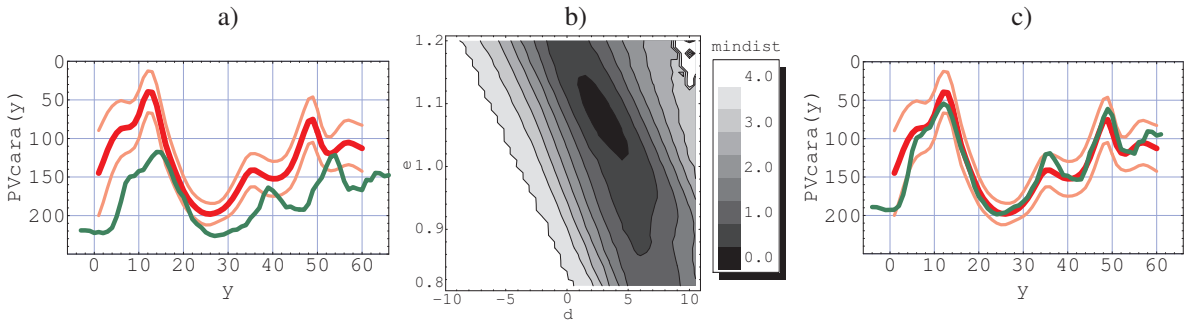


Figura 2.20: *Ejemplo de alineamiento y mapa de distancias mínimas. a) Modelo (MV_{cara}, VV_{cara}) de proyección vertical de la cara (en rojo) e instancia de cara PV_{cara} no alineada (en verde). b) Mapa de distancias mindist $((MV_{cara}, VV_{cara}), P, d, e)$, para $d \in [-10, ..., 10]$, $e \in [0,8, ..., 1,2]$. c) Alineamiento óptimo de la proyección, $t_{abcde}(PV_{cara})$, con $a = -101,1; b = 0,0018; c = 1,59; d = 2,44; e = 1,065$.*

Planteamiento del algoritmo de alineamiento rápido

Como acabamos de discutir, el alineamiento óptimo se puede conseguir buscando el mínimo de cierta función, *mindist*, en un plano. Pero, en la práctica, calcular todos los valores de esa función dentro del plano, y con una resolución suficiente, puede ser excesivamente costoso. Aunque el cálculo de cada valor de *mindist* es rápido (como explicamos al final del apartado 2.3.2), la repetición del mismo para cada posible escala y traslación haría el proceso muy ineficiente. Es más, el alineamiento será una parte básica y muy utilizada en los algoritmos de procesamiento de imágenes, algunos de los cuales, además, deberán trabajar en tiempo real. La eficiencia es, por lo tanto, un factor a tener muy en cuenta.

La función objetivo no es necesariamente convexa. Pero, afortunadamente, los mapas de distancias –como ocurre con el de la figura 2.20b)– variarán de forma suave y no existirán muchos mínimos locales destacados, aunque pueden ocurrir. En estas condiciones, podemos aplicar un algoritmo de basado en un **muestreo y acotamiento** progresivo de la posición del mínimo: el algoritmo calcula la función en un conjunto predefinido de puntos, selecciona el mínimo, y reduce la búsqueda a un entorno del mínimo encontrado. El muestreo inicial de los puntos puede utilizar diferentes criterios. Hemos podido comprobar que un simple muestreo uniforme, en ambas dimensiones, funciona correctamente en la mayoría de los casos.

En definitiva, en el algoritmo 2.4 se propone una implementación en pseudocódigo del proceso de alineamiento rápido de proyecciones. La clave del algoritmo es la aplicación de la función *mindist* sobre una matriz de valores definida por el par de arrays $valD$ y $valE$. El proceso se repite un número dado de veces, n.

En la figura 2.21 se explica gráficamente del funcionamiento del algoritmo 2.4, para la señal y el modelo de la figura 2.20. El resultado conseguido en este caso es bastante interesante, como se deduce por la forma de la función y como se puede comprobar en la figura 2.20c), donde se ha aplicado el resultado del alineamiento.

ALINEAMIENTO RÁPIDO DE INTEGRALES PROYECTIVAS
ENTRADA:
- Proyección a alinear: P
- Modelo de proyección: (M, V)
- Intervalo de búsqueda en desplazamiento: d_{min}, d_{max}
- Intervalo de búsqueda en escala: e_{min}, e_{max}
- Número de iteraciones: n

SALIDA:
- Desplazamiento de la señal alineada: d
- Escala de la señal alineada: e

ALGORITMO:
Inicialización:
$valD$: array [1..3] = $\{d_{min}, (d_{min} + d_{max})/2, d_{max}\}$
$valE$: array [1..3] = $\{e_{min}, \sqrt{e_{min} \cdot e_{max}}, e_{max}\}$
$mapa$: matriz [1..3, 1..3] = $\{mindist((M, V), P, valD[i], valE[j])\};\ \forall i, j \in \{1, 2, 3\}$
Iteración principal:
para $iter := 1$ ***hasta*** n ***hacer***
$(i^*, j^*) := \arg\min_{\forall i,j \in \{1,2,3\}} mapa[i, j]$ /* Buscar distancia mínima */
según i^***:** /* Acotar en el desplazamiento */
caso 1: $valD[3] := valD[2];\ valD[2] := (valD[1] + valD[3])/2$
caso 2: $valD[1] := (valD[1] + valD[2])/2;\ valD[3] := (valD[2] + valD[3])/2$
caso 3: $valD[1] := valD[2];\ valD[2] := (valD[1] + valD[3])/2$
finsegún
según j^***:** /* Acotar en la escala */
caso 1: $valE[3] := valE[2];\ valE[2] := \sqrt{valE[1] \cdot valE[3]}$
caso 2: $valE[1] := \sqrt{valE[1] \cdot valE[2]};\ valE[3] := \sqrt{valE[2] \cdot valE[3]}$
caso 3: $valE[1] := valE[2];\ valE[2] := \sqrt{valE[1] \cdot valE[3]}$
finsegún
para $i := 1$ ***hasta*** 3 ***hacer*** /* Recalcular las distancias */
para $j := 1$ ***hasta*** 3 ***hacer***
$mapa[i, j] := mindist((M, V), P, valD[i], valE[j])$
finpara
finpara
finpara
Resultado final del algoritmo:
$(i^*, j^*) := \arg\min_{\forall i,j \in \{1,2,3\}} mapa[i, j]$
$d := valD[i^*]$
$e := valE[j^*]$

Algoritmo 2.4: *Algoritmo rápido de alineamiento de integrales proyectivas, respecto de un modelo de proyección. La precisión del resultado depende del número de iteraciones, n.*

Discusión del método y ejemplos de ejecución

En la figura 2.22 se presentan algunos ejemplos de ejecución del algoritmo rápido de alineamiento de proyecciones. Se muestran las imágenes de entrada, las proyecciones originales y las obtenidas después de alinear con los resultados del algoritmo.

Podemos destacar las siguientes características del método de alineamiento propuesto:

- **Genericidad y capacidad de entrenamiento.** La técnica de alineamiento desarrollada es

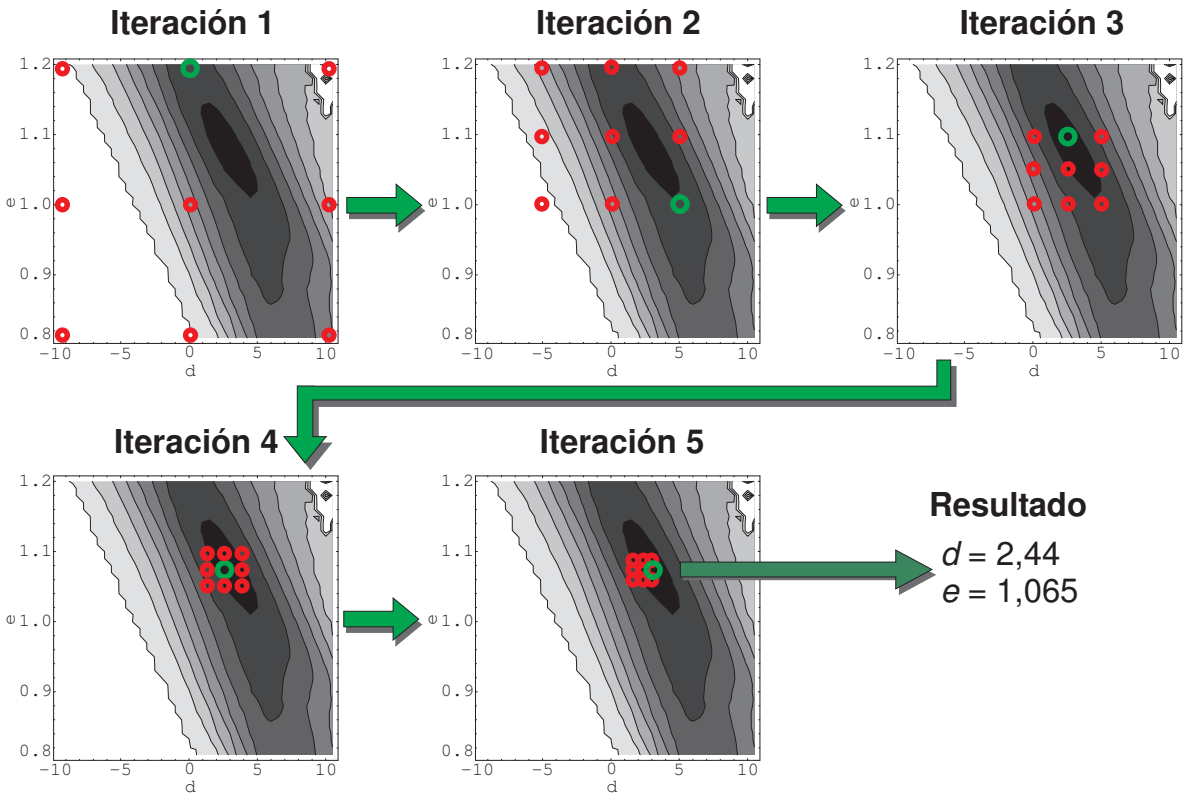


Figura 2.21: *Ejemplo del algoritmo rápido de alineamiento de integrales proyectivas (algoritmo 2.4), aplicado sobre el modelo y la proyección de la figura 2.20a). Partiendo de una región inicial de búsqueda, se realiza un muestreo uniforme de la función mindist, y una reducción sucesiva del espacio de búsqueda a la mitad. En rojo se muestran los puntos muestreados y en verde el de valor mínimo.*

genérica, e independiente de su aplicación posterior. El alineamiento no se basa en conocimiento a priori sobre el problema, sino que se realiza a través de un *modelo entrenado con ejemplos*. Es más, se podrían utilizar otras clases de modelos, sin más que cambiar la definición de la función *mindist*. Es importante destacar este hecho, en contraposición a otros muchos trabajos previos sobre proyecciones, en su mayoría basados en búsquedas heurísticas de picos máximos y mínimos, umbralización de las proyecciones, o a lo sumo aplicando lógica difusa con reglas definidas por el humano.

- **Invarianza frente a cambios.** En el dominio de las caras, los resultados demuestran una gran robustez frente a condiciones de iluminación, individuos, expresión facial, pequeños cambios de posición, escala y rotación de las regiones proyectadas. Incluso la aparición de sombras, que modifican sustancialmente la apariencia de las imágenes, es tratada adecuadamente en muchos casos, como puede comprobarse en los ejemplos de la figura 2.22. Tras el alineamiento, las proyecciones resultantes se aproximan bastante al modelo medio calculado en el entrenamiento.

- **Aplicación de los resultados.** Los resultados del algoritmo tienen dos aplicaciones fundamentales. La primera es la definición de un método de *localización de los objetos proyec-*

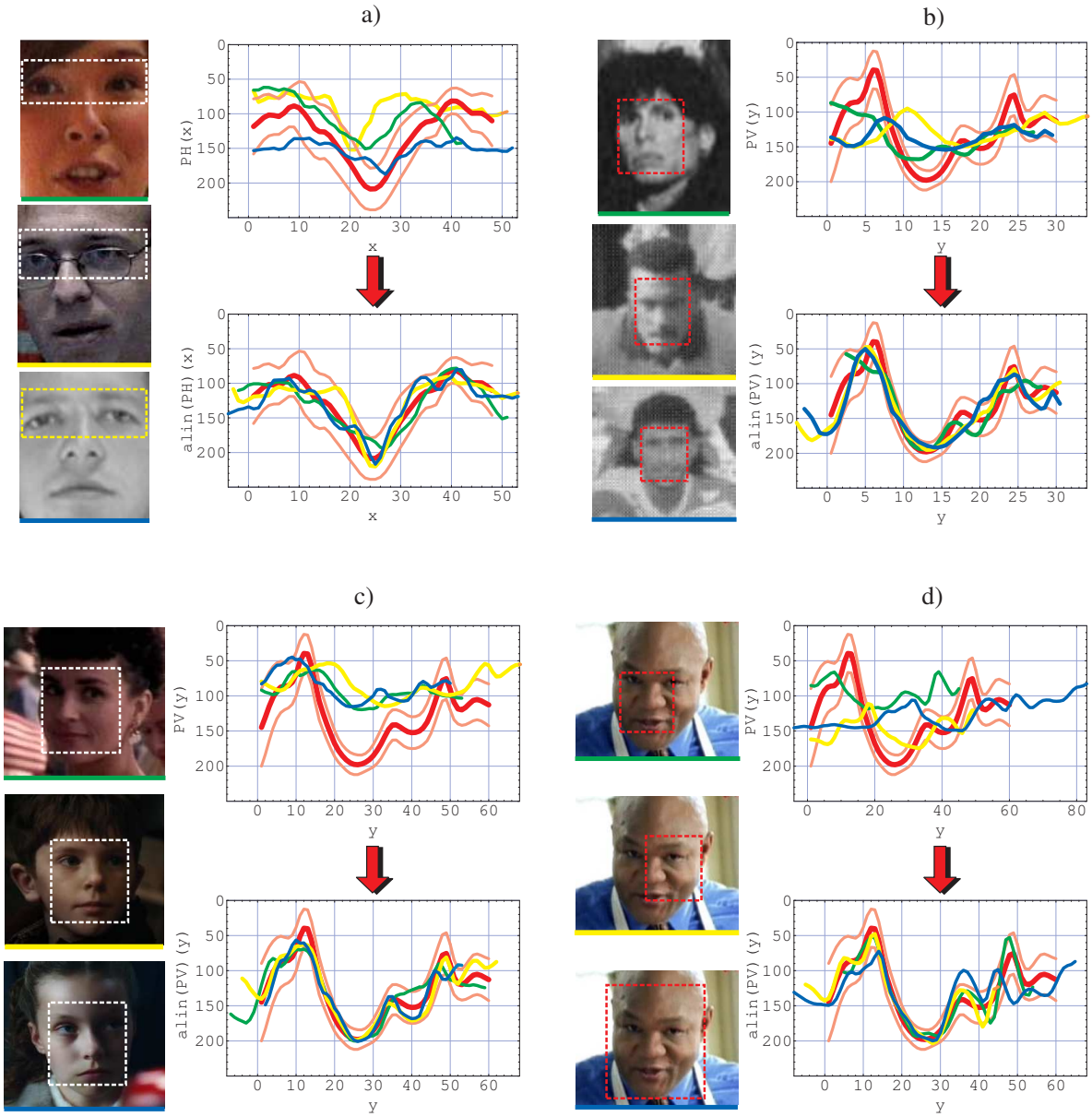


Figura 2.22: *Resultados del algoritmo rápido de alineamiento de integrales proyectivas (algoritmo 2.4). Para cada ejemplo se muestran las imágenes de entrada (a la izquierda), las proyecciones obtenidas (arriba a la derecha), y las proyecciones resultantes del alineamiento (abajo a la derecha). En rojo, los modelos de proyección media/varianza. Se indica en línea discontinua la región proyectada. a) Proyecciones horizontales de la zona de los ojos. b,c,d) Proyecciones verticales de caras, variando distintos factores. b) Caras con muy poca resolución (*35×36*,* 28×28 *y* 23×24 *píxeles, respectivamente). c) Caras con sombras muy destacadas. d) Variación de la región proyectada.*

tados. Ya adelantamos en el apartado 2.2.1 que, en una aplicación realista, no se puede suponer que los objetos están alineados en las imágenes. Partiendo de una localización imprecisa, el alineamiento produce dos parámetros de transformación en el dominio, (d, e); estos valores se traducen en una traslación y escala en el espacio de la imagen, a lo largo del eje de proyección. Por lo tanto, el uso de los mismos conduce a una lo-

calización precisa de los objetos proyectados. La segunda aplicación es el *refinamiento de la medida de distancia* señal/modelo. Al no suponerse un alineamiento previo de los objetos, esta medida sí podrá aprovecharse en aplicaciones de detección, localización y seguimiento de objetos.

- **Eficiencia computacional.** La idea básica del algoritmo rápido de alineamiento consiste en la reducción logarítmica del espacio de búsqueda en cada iteración. Esta estrategia da lugar a una gran eficiencia en cuanto al tiempo de ejecución. Suponiendo que el algoritmo empieza con un espacio de tamaño w (en la dimensión d o en la e), después de n iteraciones el ancho de la región de búsqueda sería $w/2^n$. Por ejemplo, si las señales son de tamaño 30 y $d = [-10, \ldots, 10]$, después de 5 iteraciones se consigue una precisión de 0,625 puntos. En general, para conseguir una resolución por debajo del píxel, el número de iteraciones debería ser $n = \lceil \log_2 w \rceil$. Normalmente, w será una proporción del tamaño de las señales, que denotamos por m; es decir, $w = pm$. Un valor típico es $p = 0,2$, lo que permite un desplazamiento máximo del 20 % de las señales.

  Cada una de las n iteraciones aplica 9 veces la función *mindist*[15]. Si tenemos en cuenta los resultados de la página 71, donde vimos que cada cálculo de *mindist* requiere $3m^2 + 3m + \varepsilon$ operaciones (siendo ε una constante, de valor despreciable frente a los otros términos), el número de operaciones en punto flotante sería: $t(m) = 9 \log_2 w(3m^2 + 2m + \varepsilon) = 9 \log_2(pm)(3m^2 + 2m + \varepsilon) \in O(m^2 \log m)$. Frente a esto, un típico algoritmo de *matching* de patrones, con regiones de tamaño $m \times m$ requeriría un $O(m^4)$; y esto resolviendo sólo el desplazamiento, sin tener en cuenta la variación de escala.

- **Puntos débiles.** A pesar de las buenas características descritas, es evidente que el método tiene también sus limitaciones, derivadas del algoritmo de alineamiento, del tipo de modelos definidos, o del propio uso de las proyecciones. En la figura 2.23 se muestran algunos casos donde el algoritmo de alineamiento no produce buenos resultados.

  Podemos señalar los siguientes puntos débiles, tanto del algoritmo de alineamiento como del uso de integrales proyectivas.

  - En primer lugar, las proyecciones se pueden calificar como una técnica basada en apariencia; podríamos hablar concretamente de la apariencia "a lo largo de una dirección". Cuando la apariencia de los objetos varía de forma sustancial, los modelos de media son poco adecuados y el alineamiento con los mismos no funciona correctamente. Aunque hemos visto en la figura 2.22 que las proyecciones son robustas en condiciones muy adversas, llevadas a un extremo el modelo fracasará con toda seguridad. En las caras humanas, por ejemplo, una rotación 3D en ángulos elevados –como sucede en la figura 2.23a,d)–, o la existencia de sombras muy

[15] En realidad podría hacerse con unas cuantas menos, ya que algunos cálculos están repetidos, como se puede observar en la figura 2.21. La eliminación de cálculos repetidos se ha tenido en cuenta en la implementación realizada, aunque por simplicidad no se contempla en el algoritmo 2.4.

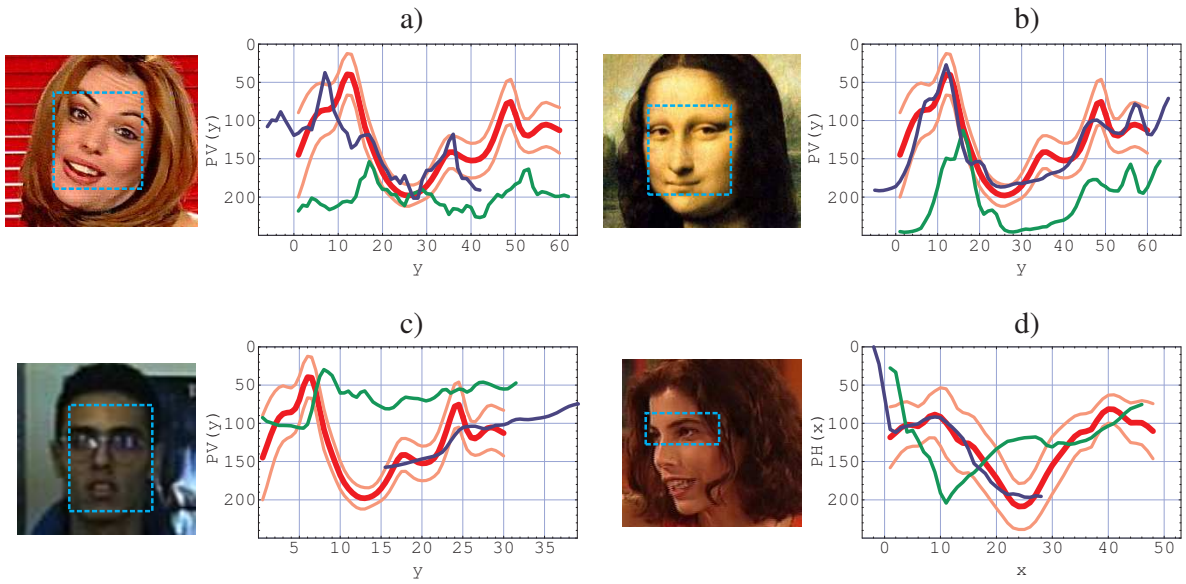


Figura 2.23: *Ejemplos de mal funcionamiento del algoritmo rápido de alineamiento de proyecciones (algoritmo 2.4). Para cada ejemplo se muestran las imágenes de entrada (a la izquierda), las proyecciones (a la derecha) originales (en verde), resultantes (en azul) y el modelo (en rojo). Se indica en línea discontinua la región proyectada. a) Ejemplo de mal alineamiento debido a inclinación excesiva de la cabeza. b) Confusión de posiciones debido a un mínimo local (la boca del modelo se asocia a la nariz en la imagen). c) Fallo del algoritmo debido a la baja calidad, existencia de elementos faciales (reflejo de las gafas) y posición de la región proyectada. d) Mal alineamiento de la proyección horizontal de los ojos, debido a un alto giro lateral del rostro.*

destacadas o el tono de piel –figura 2.23c)– pueden conducir a esta situación. En principio, pensamos que estos problemas se podrían solucionar usando otros tipos de modelos no unimodales, sino que admitieran varios modos de variación. Por ejemplo, podríamos definir modelos de k-proyecciones medias, análisis de componentes principales de las señales, o modelos de proyección deformables. Trataremos más sobre esta cuestión en los capítulos sucesivos.

- Una limitación inherente al algoritmo 2.4 es la localidad del método: se debe partir siempre de un rectángulo inicial en el espacio (d, e), dado por $(d_{min}, \ldots, d_{max})$ y $(e_{min}, \ldots, e_{max})$. Si la escala o la traslación ideal caen fuera de esos intervalos, el resultado puede no ser muy bueno. Aunque podemos intentar ampliar los intervalos para disminuir este inconveniente, siempre existirá un límite inmanente al método. Por ejemplo, para una señal de tamaño m carece de sentido aplicar desplazamientos mayores que m. Esto significa que la región proyectada debe ser próxima al objeto, aunque no necesariamente exacta. No obstante, esta limitación no es exclusiva del algoritmo, sino que se debe al uso de proyecciones: una proyección aplicada sobre una región tan extensa que incluya varios objetos puede ser de poca utilidad.

- En algunos casos, el algoritmo puede caer en mínimos locales de la función *mindist*, lo cual conducirá a un mal alineamiento de esas señales, incluso aunque sean muy

similares al modelo medio. En la práctica el problema no es muy frecuente, aunque puede darse. En caras humanas hay una situación típica de este fenómeno, derivada de la confusión del mínimo de la nariz con el correspondiente a la boca. Se puede ver un ejemplo en la figura 2.23b). Aunque la distancia para el alineamiento correcto es menor que el obtenido, el incorrecto (que hace corresponder la nariz con la boca) genera un mínimo local en el plano *mindist*, del que no llega a salir el algoritmo.

A pesar de estos inconvenientes, las ventajas de usar modelos de proyecciones y el algoritmo de alineamiento superan con creces sus limitaciones, y nos permitirán crear los métodos de detección de caras, localización, seguimiento, etc., que veremos en los restantes capítulos.

2.4. Resumen

A lo largo de este capítulo se ha desarrollado un marco teórico para el cálculo, manejo, modelado y comparación de integrales proyectivas, desde un punto de vista genérico. Este contexto constituye la base de los métodos que proponemos para resolver las diversas aplicaciones en el dominio específico de las caras humanas.

Podemos destacar las siguientes conclusiones de todo lo expuesto en el capítulo:

- La proyección es un método más de reducción de una alta dimensionalidad de entrada, al igual que otros métodos basados en **transformaciones lineales** como PCA, ICA, DFT, Haar o wavelets. De hecho, la proyección es análoga a la transformada de Radon, donde cada valor de salida es una combinación lineal de una fila o columna de píxeles.

- Al igual que otras transformaciones lineales, la **proyección es invertible**: a partir de un número suficiente de integrales proyectivas se puede reconstruir la imagen original. Esto significa que no hay pérdida de información inherente a la propia proyección, sino la derivada de usar un número reducido de proyecciones.

- Los conjuntos de integrales proyectivas asociadas a cierta clase de objetos, son descritos mediante **modelos de proyección**. Hemos propuesto varios métodos alternativos de modelado: proyección media; media/varianza; y media/covarianzas. Los modelos más simples –los dos primeros– suelen exhibir una mayor capacidad de generalización.

- Los modelos de objetos mediante integrales proyectivas son, esencialmente, **modelos 1,5D** basados en apariencia. Cada modelo de objetos incluye uno o varios modelos de proyección asociados. Aunque el método es comparable a un modelado mediante patrones 2D, hemos comprobado que pueden mejorar sensiblemente las capacidades de generalización y clasificación de los segundos.

- El **alineamiento de integrales proyectivas** es uno de los problemas fundamentales cuando trabajamos con proyecciones. Se puede entender como la reducción al caso 1D del

problema de localización de patrones en imágenes 2D. Hemos propuesto un algoritmo rápido y robusto para alinear proyecciones entre sí, o respecto de un modelo. Las dos ideas básicas del algoritmo son: (1) resolución analítica de los parámetros de alineamiento en el valor con una formulación de mínimos cuadrados; (2) resolución de los parámetros en el dominio mediante una técnica iterativa de muestreo y acotamiento. Este algoritmo es el núcleo de los métodos de detección, localización, seguimiento, reconocimiento, etc., que estudiamos en los siguientes capítulos.

CAPÍTULO 3

Detalle de la superficie marciana,
Sonda Viking, NASA, 1976

Detección de Caras Humanas

"Como en el asunto de las caras de Marte,
prefiero la cruda verdad a una fantasía reconfortante."
CARL SAGAN, *The Demon-Haunted World*, 1996

La detección facial es un problema preliminar en la gran mayoría de las aplicaciones de reconocimiento, interpretación y seguimiento del rostro humano. Indefectiblemente, el primer paso de cualquiera de estos sistemas será encontrar las caras que aparecen en una imagen; sólo en circunstancias donde se pudiera dar por conocida la posición de los individuos –condición, por supuesto, nada frecuente y poco realista– se podría evitar el papel primordial del problema de detección.

Las técnicas de detección de caras han seguido su propio camino en el mundo de la visión artificial; convergente a veces, divergente en otros casos, y en general paralelo al problema genérico de detectar objetos tridimensionales. Pero, en la mayoría de los casos, hay una clara especialización hacia el objeto *cara*, que supone introducir consideraciones específicas, aplicables o no a otras categorías de objetos. La detección con integrales proyectivas, que proponemos y desarrollamos en este capítulo, se encuentra en ese grupo de técnicas creadas para un fin específico. No obstante, veremos que la idea subyacente se puede extender con relativa facilidad al caso general, ya que no se apoya en conocimiento a priori sobre la clase cara.

En el resto de este capítulo vamos a describir cómo es posible detectar caras humanas usando, de forma exclusiva, integrales proyectivas. De esta manera, el objetivo no es tanto desarrollar un sistema que supere a los mejores métodos del estado del arte, sino demostrar que es posible construir un buen detector trabajando sólo con las proyecciones. No obstante, veremos también que combinando las proyecciones con otras técnicas se pueden mejorar sustancialmente los resultados de ambos. Tanto el primer mecanismo como el segundo se abordan en la sección 3.3. Antes, discutimos los desafíos que presenta la detección de caras humanas, en la sección 3.1, y analizamos el estado del arte en este tremendamente activo campo, en la

sección 3.2. En la sección 3.4 se detallan los experimentos que demuestran la viabilidad de la técnica propuesta, haciendo especial hincapié en la comparación con algunos de los principales métodos alternativos. Por último, la sección 3.5 extrae las principales conclusiones de estas pruebas, y la sección 3.6 resume las aportaciones más relevantes del presente capítulo.

3.1. El problema de detección de caras humanas

Repasando algunos de los numerosos trabajos sobre detección facial, podemos encontrar una variedad de definiciones del problema, aunque todas ellas esencialmente equivalentes. Una definición típica podría ser la siguiente [204].

Definición 3.1 ***Detección de cara humanas.***
Dada una imagen arbitraria, el objetivo de la detección facial es determinar si aparecen caras en la imagen y, en caso afirmativo, devolver la localización y la extensión de cada una de ellas.

De esta forma, se entiende que el resultado de un detector de caras debe ser una lista de regiones de cara, que puede estar vacía. Pero, aunque la definición parece clara y precisa, si analizamos en detalle la literatura vemos que existen diferentes criterios en cuanto a cómo debe ser la entrada, la salida, e incluso lo que parece más evidente: qué es una cara. Podemos señalar las siguientes fuentes de ambigüedad:

- **Entrada**. La mayoría de las técnicas de detección de caras parten de una *imagen estática*. Sólo algunas de ellas hacen uso de información de movimiento, por lo que requieren *secuencias de imágenes* para encontrar los rostros. Por su parte, dentro de las primeras podemos distinguir entre las que trabajan con *imágenes en color*, y las que admiten *imágenes en escala de grises*. Las de este segundo tipo presentan una mayor versatilidad, teniendo en cuenta que la conversión de color a escala de grises es inmediata. Adicionalmente, podemos encontrar también técnicas que trabajan con *imágenes de profundidad* [96, 41], o de *infrarrojos* [100], lo que, obviamente, tiene importantes implicaciones sobre los sistemas de adquisición utilizados.

- **Salida**. Hemos dicho que la salida de los detectores es una *lista de regiones* de cara. En principio, existen diferentes formas de describir una región. El formato más habitual es mediante un *rectángulo contenedor* del rostro; el tamaño y posición de ese rectángulo, respecto de la cara, puede variar ligeramente de un trabajo a otro. En algunos casos la región se describe mediante una *elipse*, lo que supone introducir un parámetro de inclinación. En otros casos, se ofrece una descripción más precisa del *contorno de la cara*, por ejemplo, mediante un polígono, un *spline* o una máscara binaria.

- **Caras humanas**. Como vamos a ver más adelante, los detectores faciales deben manejar la enorme variabilidad del objeto cara. Sin embargo, no existe un acuerdo unánime respecto a lo que los algoritmos deben reconocer como cara y lo que no. Véanse, por

ejemplo, las imágenes de la figura 3.1, tomadas de la base de caras CMU/MIT [152]. Todas ellas son clasificadas por los autores como rostros humanos. Resulta discutible admitir que una *caricatura de una cara* deba ser considerada también como una cara *humana* –sobre todo cuando el dibujo es tan esquemático como los de la izquierda en la figura 3.1–. Pero a medida que el trazo se hace más preciso, es difícil no admitirlo como cara; luego ¿dónde está el límite? Por otro lado, algunos autores limitan ciertas fuentes de variación, obviando de antemano circunstancias adversas como: *caras de perfil* o con un fuerte giro (mirada arriba-abajo, izquierda-derecha), *rotación* respecto del plano de la imagen, y *oclusión parcial*.

Figura 3.1: *Ejemplos discutibles de caricaturas y dibujos clasificados como caras humanas, en la base de caras CMU/MIT. De izquierda a derecha, se muestran extractos de las imágenes: u2-cover.gif, henry.gif, board.gif y divinci-man1.gif.*

Un problema relacionado con la detección es la *localización* de caras humanas. En la localización se parte de la premisa de que existe una cara en la imagen, por lo que el método debe centrar su atención en encontrar la posición y tamaño de la única cara presente. Esta suposición simplifica considerablemente el problema, aunque puede tener sentido en determinadas aplicaciones, como en videoconferencia o en algunos sistemas de autentificación.

3.1.1. Dificultades y desafíos en la detección de caras

Idealmente, un buen algoritmo de detección debería ser capaz de encontrar las caras que aparecen en una imagen bajo cualquier circunstancia. El problema es complejo, ya que los posibles cambios de iluminación, posición, expresión, condiciones de captura, etc., pueden afectar de forma decisiva a la apariencia de los rostros en las imágenes. En última instancia, todos los detectores tienen un cierto margen de tolerancia frente a estos factores; fuera de esos límites, la técnica fracasará con toda probabilidad.

Vamos a analizar las principales fuentes de variación de apariencia del objeto cara, mostrando algunos ejemplos de las bases de datos usadas en la experimentación posterior.

- **Condiciones de captura**

  Las aplicaciones de detección de caras pueden utilizar diferentes sistemas de adquisición (cámaras de vídeo, de fotos, escáner, etc.) y fuentes de entrada (televisión analógica,

TDT, DVD, etc.). Algunas de ellas son más propicias a la aparición de problemas de ruido, desenfoque, saturación, o escasa resolución en las caras de entrada. Un sistema de detección robusto debería ser capaz de trabajar en estas circunstancias adversas. En la figura 3.2 se muestran algunos ejemplos de estos problemas.

Figura 3.2: *Ejemplos de baja calidad de imagen debida a las condiciones de captura: escasa resolución, desenfoque y saturación del brillo. De izquierda a derecha, extractos de: Argentina.gif (CMU/MIT), c380.jpg (UMU), b189.jpg (UMU), cnn2020.gif (CMU/MIT).*

- **Posición y orientación 3D de la cara**

  Las caras humanas son objetos tridimensionales cuya apariencia varía significativamente con la posición relativa de la cámara. Con ángulos elevados puede ocurrir, incluso, oclusión parcial o total de algunos componentes faciales, como los ojos y la boca. El giro lateral (perfil derecho/perfil izquierdo) es el que más atención ha recibido en la literatura; por el contrario, los trabajos que abordan los efectos del giro vertical (mirar arriba/abajo) son más escasos. Se pueden ver algunos ejemplos presentes en nuestros experimentos en la figura 3.3.

Figura 3.3: *Ejemplos de variación de apariencia debida a la posición 3D de la cara. Todos los ejemplos son de la base de caras UMU. De izquierda a derecha, extractos de: 2076.jpg, 4016.jpg, 2M2.jpg, 1002.jpg.*

- **Inclinación**

  Normalmente, el giro de la cara respecto del plano de la imagen (lo que denominamos la *orientación* o *inclinación de la cara*) se suele considerar como un factor aparte de las rotaciones fuera del plano de imagen. Esto es debido a que la inclinación puede ser compensada con una simple rotación de la imagen en ángulo adecuado. Sin embargo, puesto que tal ángulo no es conocido, la invarianza a la orientación es una cuestión

a considerar en cualquier detector. La figura 3.4 muestra algunos casos de caras con inclinaciones elevadas.

Figura 3.4: *Ejemplos de variación de apariencia debida a la inclinación de las caras. De izquierda a derecha, extractos de: mir.gif (CMU/MIT), cast1.gif (CMU/MIT), 541.jpg (UMU), 503.jpg (UMU).*

- **Factores ambientales**

  La iluminación es uno de los factores ambientales que mayor efecto tienen en la apariencia facial. La luz no sólo afecta al nivel de brillo de las imágenes, sino que puede dar lugar a la aparición de sombras. Y, en el caso de las imágenes en color, puede modificar también el matiz de la piel. Otros problemas menos frecuentes son los reflejos y la iluminación escasa o excesiva. En la figura 3.5 se presentan algunos ejemplos típicos. Cabe destacar que en algunos detectores el fondo de las imágenes también tiene influencia en los resultados del método, bajando la efectividad con fondos complejos.

Figura 3.5: *Ejemplos de variación de apariencia debida a la iluminación de las caras. De izquierda a derecha, extractos de: 1091.jpg (UMU), 1074.jpg (UMU), d227.jpg (UMU), whussa.2.jpg (ESSEX).*

- **Diferencias entre individuos**

  Frente a los anteriores factores, que podemos calificar como "externos", tenemos otra serie de causas "internas" de variación, es decir, las debidas a las propias caras en sí. La más evidente es la originada por las características individuales de cada persona: la forma de la cara, el color de la piel, la edad, el sexo, el grupo étnico, etc. La figura 3.6 contiene una pequeña muestra de esta inagotable fuente de variación.

- **Expresión facial**

  Tradicionalmente, la visión artificial se ha centrado en el análisis de objetos 3D rígidos. Sin embargo, las caras se caracterizan por ser deformables en un número casi inima-

Figura 3.6: *Ejemplos de variación de apariencia por características del individuo, de la base UMU. De izquierda a derecha, extractos de: 304.jpg, 95H4.jpg, 69H2.jpg, l007.jpg, DSCN1132.jpg.*

ginable de formas. La flexibilidad del rostro se materializa en las expresiones faciales, intencionadas o naturales. Se puede ver, en la figura 3.7, que una expresión exagerada puede suponer también una variación sustancial del aspecto de las caras.

Figura 3.7: *Ejemplos de variación de apariencia debida a la expresión facial. Extractos de la secuencia de prueba para seguimiento de caras de Buenaposada y otros [19].*

- **Oclusión y elementos faciales**

Por último, podemos señalar la aparición de elementos faciales, tales como gafas, barba, bigote, tatuajes, etc. En algunos casos, puede ocurrir oclusión provocada por otras caras o por cualquier otro objeto de la escena. No es arriesgado afirmar que un individuo que intencionadamente se oculte de la cámara, no será detectado por ninguno de los métodos existentes. Se muestran algunos casos de estos elementos en la figura 3.8.

Figura 3.8: *Ejemplos de variación de apariencia debida a oclusión y elementos faciales, de la base de caras UMU. De izquierda a derecha, extractos de: DSCN4234.jpg, c375.jpg, 18.jpg, 2003.jpg (imagen completa).*

3.1.2. Objetivos y evaluación de los detectores

Los algoritmos de detección pueden incurrir en dos tipos de errores [204]: *falsos positivos* y *falsos negativos*. Los falsos positivos (también denominados *falsas alarmas*) son regiones detectadas como caras, pero que en realidad no lo son. Por su parte, los falsos negativos son las caras existentes pero no encontradas por el detector. Ambos suelen expresarse en función del *número de caras* presentes en las imágenes –normalmente etiquetadas de forma manual por un humano–, del *número de imágenes* existentes o del *número de tests* cara/no cara aplicados.

Una vez obtenidos los valores anteriores para cada ejecución particular del detector, las dos medidas de rendimiento estándar son:

1. **Ratio de detección**. Se define típicamente como el número de caras detectadas en función del número de caras existentes, o equivalentemente 1 – *falsos negativos*/*número de caras*. Se dice que una cara ha sido detectada si la región devuelta por el algoritmo contiene un alto porcentaje del rostro existente, por encima de cierto umbral.

2. **Ratio de falsos positivos**. No está clara cuál es la mejor forma de definir este porcentaje. Parece lógico medirlo en función del número de imágenes, más que de las caras presentes. Pero también se expresa a veces en relación al número total de detecciones, o del número de tests cara/no cara. En este último caso, la medida está centrada en la clasificación cara/no cara, que, como vamos a ver, es sólo una parte del proceso de detección; y, además, sólo de algunos métodos de detección[1]. Para evitar ambigüedades, muchas veces se indica directamente el número de falsas detecciones.

No es correcto decir que un método con mayor ratio de detección es mejor, si no se tiene también en cuenta el número de falsos positivos. Como en el problema general de clasificación de patrones, cualquier mecanismo de detección dispone de unos ajustes que permiten controlar el grado de tolerancia en la decisión cara/no cara. Un ajuste "permisivo" aumenta el número de detecciones, pero también el de falsas alarmas, mientras que un ajuste "restrictivo" reduce ambos. Los distintos modos de funcionamiento, para diferentes ajustes del detector, dan lugar al concepto de *curva ROC* y las siguientes medidas asociadas:

1. **Curva ROC** (o *Receiver Operating Characteristics*). La curva ROC es la representación gráfica del ratio de detección frente al de falsos positivos, para distintos modos de operación del detector. Por sí misma, esta curva es el resultado más interesante de la experimentación. Se puede entender como una especificación de cara al usuario de los distintos modos de ejecución del detector. Según la gravedad de cada tipo de error –es decir, la mayor o menor relevancia de los falsos positivos o negativos–, se seleccionará un modo de funcionamiento en cada aplicación concreta. En la figura 3.9 se puede ver un ejemplo de una curva ROC típica.

[1] De hecho, ese tipo de detectores se basa en aplicar un número muy elevado de tests, fácilmente por encima de 10^5, de forma que el ratio será casi siempre muy bajo. Por lo tanto, es evidente que esta medida no se puede comparar directamente con el ratio en función del número de imágenes.

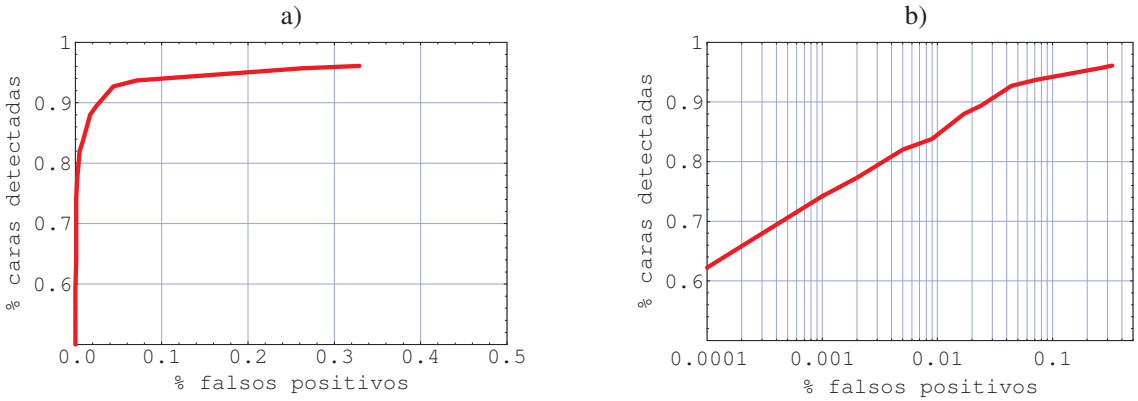


Figura 3.9: *Ejemplo de curva ROC de un método de detección combinado sobre la base UMU. a) Curva ROC del método, para diferentes modos de operación. b) La misma curva usando una escala logarítmica en el eje horizontal. Los ratios de falsos positivos están en función al número de imágenes.*

2. **Ratio de error igual**. Como vimos en el apartado 2.2.1 del capítulo anterior, es el ratio de error que produciría el detector en un ajuste con igual número de falsos positivos que de falsos negativos. Es decir, es el punto de intersección de la ROC con la recta que va de (0, 1) a (1, 0). Esta medida, por sí sola, puede utilizarse para comparar dos detectores, puesto que ofrece más uniformidad. Pero hay que tener en cuenta que el modo de funcionamiento ideal puede no ser necesariamente el asociado a ese ajuste.

3. **Área bajo la curva ROC**. Es una medida numérica extraída de la curva ROC, pero más robusta que el porcentaje de error igual. Será mejor cuanto mayor sea, puesto que indica que se alcanzan antes ratios altos de detección para un número bajo de falsos positivos. Es una medida compleja de obtener, y por ello se suele usar con muy poca frecuencia, y particularmente en la literatura de detección de caras.

Por otro lado, los detectores de caras no son meras funciones matemáticas, sino que son algoritmos cuya ejecución consume recursos de tiempo y memoria. En concreto, el **tiempo de ejecución del detector** puede ser especialmente relevante en muchas aplicaciones, y resultará un factor crítico en aquellas que requieran trabajar en tiempo real. En ocasiones también se ofrecen datos comparativos sobre el tiempo de entrenamiento del proceso, aunque pensamos que la importancia de este recurso es considerablemente inferior.

Finalmente, cualquier comparación justa entre dos mecanismos diferentes de detección de caras debería tener en cuenta también las restricciones adicionales del propio detector: el tamaño mínimo de las caras encontradas, trabajar sólo con imágenes en color o en escala de grises, el grado de tolerancia a rotaciones, la facilidad de implementación, etc.

3.2. El estado del arte en detección de caras

No es exagerado decir que, en el ámbito de la visión artificial, la detección de caras humanas ha sido uno de los problemas que más atención ha recibido en los últimos años, y especialmente a lo largo de la última década. Como muestra, podemos señalar dos fechas relevantes, marcadas por la publicación de sendas revisiones bibliográficas con impacto en la comunidad investigadora: en 1995, Chellapa y otros [26], concluyen que la investigación en detección de caras había recibido sorprendentemente una escasa atención; sin embargo, siete años más tarde, en 2002, Yang y otros [204], son capaces de identificar más de 150 acercamientos distintos al problema (posiblemente, tantos como grupos trabajando sobre el mismo).

En la actualidad, el campo de la detección facial parece haber alcanzado cierta madurez, con la existencia de varias implementaciones bastante fiables y de dominio público [110, 152]. No obstante, algunas cuestiones siguen aún pendientes [200]: la mejora de la robustez frente a expresiones, oclusiones, sombras y rotaciones; el aumento de la precisión y la eficiencia computacional de las técnicas existentes; y el establecimiento de criterios y protocolos estandarizados para la evaluación y comparación de los métodos.

Adoptando la clasificación de [204], podemos distinguir cuatro grandes categorías de métodos de detección de caras humanas: basados en conocimiento, en propiedades invariantes, en patrones predefinidos, y en apariencia. Hjelmas y Low [81], proponen una clasificación alternativa, mostrada en la figura 3.10, en la que se separa entre los acercamientos que tratan la imagen como un todo –también conocidos como *holísticos*–, y los hacen uso de características específicas de la cara.

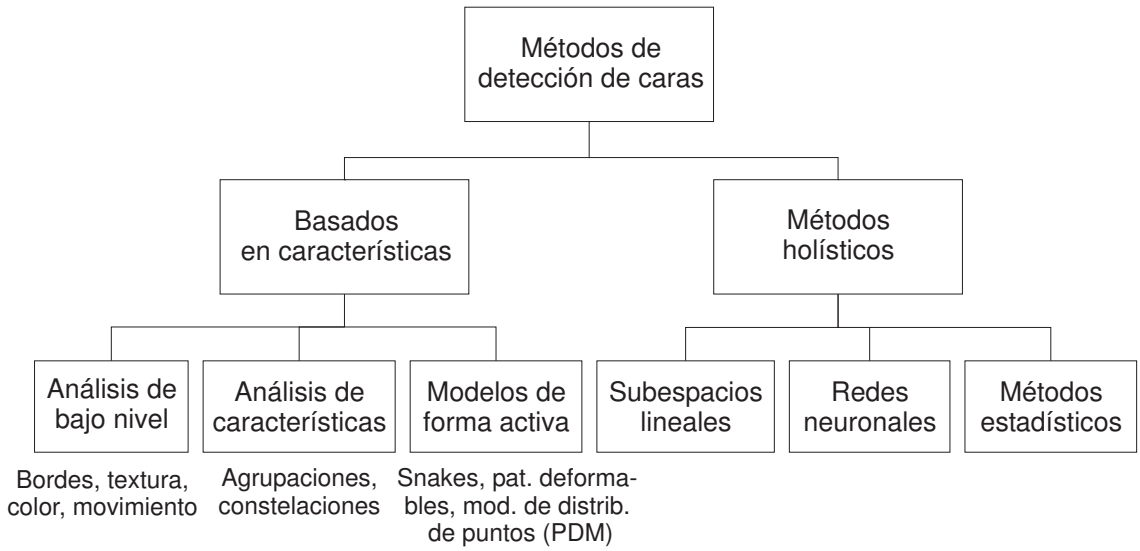


Figura 3.10: *Clasificación de métodos de detección de caras, según la propuesta de Hjelmas y Low [81].*

Realmente, ambas taxonomías son más una distinción de los principios metodológicos subyacentes que una clasificación disjunta de las técnicas existentes. De hecho, como se señala

en [204], las fronteras entre clases son, muchas veces, difusas y un algoritmo puede aplicar principios de diferentes categorías.

A continuación vamos a destacar los principales fundamentos y los trabajos más relevantes dentro cada grupo, basándonos en la primera clasificación.

3.2.1. Métodos descendentes basados en conocimiento

Las propuestas dentro de este grupo se fundamentan en el conocimiento "experto" del investigador, que codifica mediante una serie de reglas predefinidas las características que se deben buscar en las imágenes. Estas reglas pueden ser del tipo: los ojos y la boca son más oscuros que su entorno; deben aparecer dos ojos, una boca y una nariz; las caras son simétricas horizontalmente; o la piel es de color más o menos uniforme.

El problema inherente a estos métodos es cómo crear un conjunto óptimo de reglas. Si son muy detalladas, se restringe la capacidad del método de generalizar a las distintas fuentes de variación que discutimos en el apartado 3.1.1. Pero si las reglas son muy generales, se puede disparar el número de falsas detecciones. Por ese motivo, estas técnicas son más adecuadas para el caso de localización, es decir, cuando se conoce que existe una única cara en las imágenes.

Otra característica común de estos métodos es la aplicación de una estrategia de búsqueda descendente (*top-down*). Un ejemplo claro es el trabajo de Yang y Huang [198], que proponen un sistema de reglas a tres niveles, con una resolución creciente en las imágenes.

1. En el primer nivel, con la resolución más baja, se busca una distribución de intensidades, dada por un patrón de 5×5 píxeles, creado de forma manual.

2. En el nivel intermedio se define un criterio basado en detección de bordes sobre los candidatos resultantes del paso anterior.

3. Los candidatos que pasan los dos primeros criterios, se someten a otro test de mayor resolución, con reglas orientadas a la distinción de cada uno de los componentes faciales por separado.

En sus experimentos, este método logra un ratio de detección del 83 % sobre un conjunto de 60 imágenes, pero con un 47 % de falsas alarmas por imagen.

También se han propuesto sistemas de reglas predefinidas orientadas al análisis de integrales proyectivas. Por ejemplo, Kotropoulos y Pitas [101], definen un método heurístico de detección de caras usando la proyección horizontal y vertical de las imágenes completas. En concreto, los bordes exteriores de la cara se localizan en las mínimos locales de la proyección horizontal, seleccionando los cambios más abruptos en la señal. De forma parecida, en la proyección vertical los mínimos locales se asocian con la posición de los ojos, nariz y boca. En la figura 3.11 se muestran dos ejemplos donde se podría aplicar este sencillo método. En el caso de la figura 3.11a), parece viable conseguir un buen resultado; en el de 3.11b) el método es impracticable.

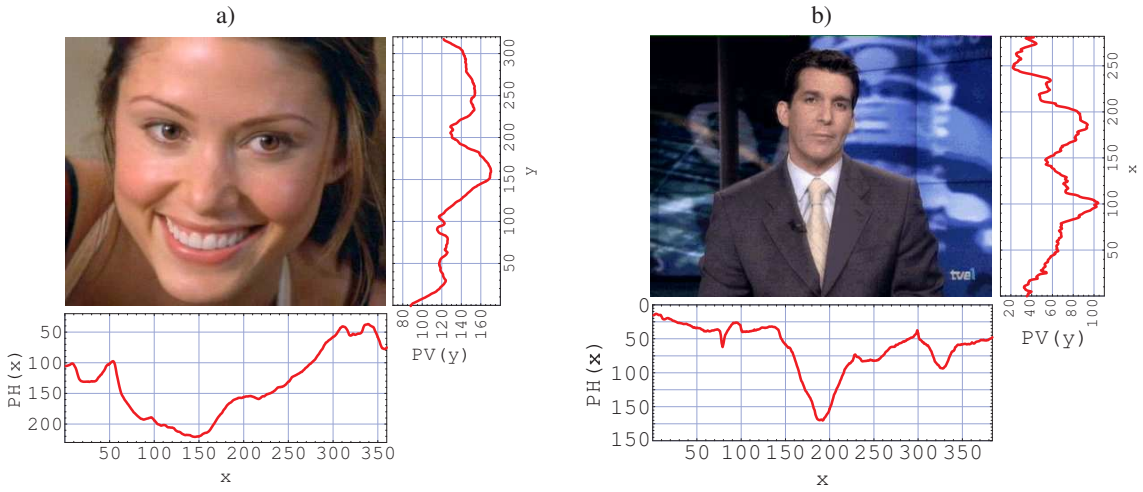


Figura 3.11: *Ejemplos de proyección vertical y horizontal de imágenes completas. El método de Kotropoulos y Pitas [101], localiza la cara buscando mínimos locales de $PH(x)$ y de $PV(y)$. a) Imagen "033.jpg" de la base UMU. b) Imagen "002.jpg" de la base UMU.*

Varias décadas atrás, en 1973, en uno de los primeros trabajos sobre análisis de caras [93], Kanade había propuesto una técnica similar para detectar el borde de la cara, pero proyectando la imagen de bordes. Kotropoulos y Pitas añaden un paso adicional de validación de los candidatos basado en reglas. En sus pruebas informan de un porcentaje de detección del 86,5 % (incluyendo también la localización correcta de los componentes faciales), aunque en las imágenes sólo aparece una cara y el fondo es uniforme. Claramente, se trata de un algoritmo orientado a localización más que a detección. No obstante, la idea de buscar picos en las proyecciones de la cara ha sido utilizada posteriormente en numerosos trabajos.

En resumen, los métodos dentro de esta categoría se ven sujetos a las enorme dificultad de expresar en un conjunto reducido de reglas la infinidad de cambios de apariencia que experimenta el rostro humano. Por ello, se puede decir que se trata de una línea abandonada hace tiempo, en la que se sitúan algunos trabajos precursores de la disciplina, pero poco viables en situaciones no triviales.

3.2.2. Métodos ascendentes basados en invariantes

La justificación biológica de estos métodos se fundamenta en la capacidad de los humanos de encontrar caras en las circunstancias más variadas. En consecuencia, se da por hecho que deben existir características de bajo nivel, fáciles de calcular, pero invariantes frente a un gran número de situaciones complejas.

Las características de interés se pueden referir al color de la piel y del cabello, la forma de la cara, la textura, los bordes, la aparición de los componentes faciales y su distribución geométrica, etc. Frente a los algoritmos basados en conocimiento, el procesamiento suele seguir un camino ascendente: primero se extraen características de bajo nivel individualmente, y luego se agrupan utilizando descriptores estadísticos.

Bordes y elementos faciales

Tradicionalmente, los bordes han sido una de las propiedades invariantes más utilizadas –como ocurre en otros muchos ámbitos de la visión artificial–, ya que aparecen típicamente en torno a los elementos faciales y al contorno del rostro. Para la extracción de bordes faciales se han propuesto diferentes posibilidades, como el operador de Canny [167, 106], convoluciones con segunda derivada de la gaussiana [209], filtros pasa-banda [72], y operaciones de morfología matemática [78]. El último puede usarse también en combinación con los anteriores, con el fin de destacar los picos de máxima respuesta [72]. En la figura 3.12 se puede ver un ejemplo típico de algunos operadores de bordes sobre una imagen de la base de caras UMU.

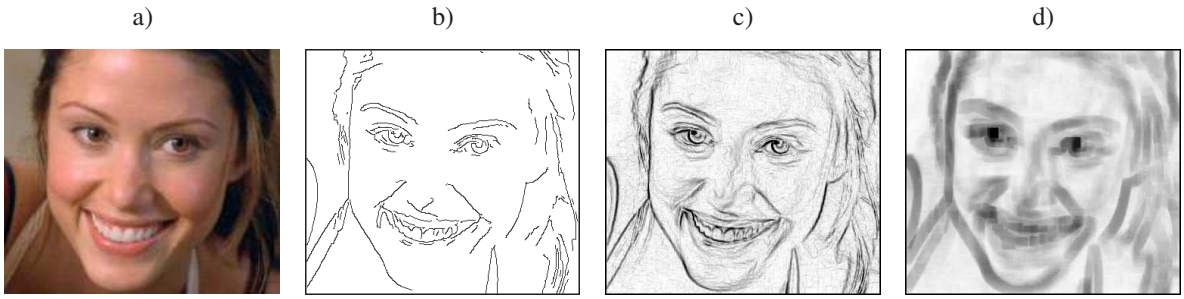


Figura 3.12: *Ejemplos de análisis de caras usando operadores de bordes, sobre la imagen "033.jpg" de la base UMU. Los bordes se sitúan normalmente en el contorno de la cara y en los componentes faciales. a) Imagen original. b) Operador de Canny. c) Gradiente obtenido con filtros de Sobel. d) Operador morfológico de gradiente.*

Después de la etapa de extracción de bordes, normalmente, se realiza una agrupación visual de las características, es decir, se asocian las que están más próximas entre sí; por ejemplo, con análisis de componentes conexos [72], o buscando segmentos de bordes con igual orientación [2, 28]. Por último, se forman las llamadas "constelaciones", definidas como los grupos de características que presentan una estructura de distancias y posiciones coherentes con un modelo de cara preestablecido. Por ejemplo, si consideramos el grupo compuesto por ojos y boca, formarían un triángulo *casi* equilátero, estando la nariz situada próxima al centro del mismo. Para describir la configuración geométrica del rostro se han usado descriptores estadísticos basados en distribuciones gaussianas [94, 117], redes bayesianas [209], o simples modelos basados en distancias.

Siguiendo una estrategia ascendente similar, algunos autores [107], utilizan detectores locales de elementos faciales (para los ojos, fosas nasales y boca), en lugar de bordes. Según la respuesta de los mismos, se buscan configuraciones plausibles de dos ojos, dos fosas nasales y una boca, mediante las técnicas de grafos conocidas como *random graph matching*.

Descriptores de textura

La textura típica de la cara es otra de las características que han sido aprovechadas, aunque en una proporción mucho menor. Por ejemplo, en el trabajo pionero de Haralick y otros [79], se utilizan *descriptores estadísticos de segundo orden* (SGLD) en bloques de 16×16 píxeles. Con

ellos, se definen tres tipos de texturas: piel, pelo y otras. La clasificación de los bloques es realizada mediante redes neuronales, y para la detección de caras sugieren –aunque no llegan a implementarlo– un esquema de votación según las ocurrencias de piel y pelo. Otro ejemplo de detección usando SGLD se puede encontrar en el más reciente [40], donde se añade información de color para aumentar la relevancia de las partes de color anaranjado.

Color de la piel humana

Como ya vimos en el apartado 1.1.2 del capítulo introductorio, el color de la piel humana es otra propiedad interesante de las caras, por su alta invarianza y la simplicidad de su cálculo. Podemos encontrar trabajos que utilizan prácticamente todos los principales espacios de color existentes: RGB [180, 116, 88], RGB normalizado [130, 25, 96, 39, 106, 24], HSV [160, 169, 170, 16, 59, 60, 66], YCrCb [130, 84], YUV [116], YES [157], CIE XYZ [195], CIE LUV [201]. En general, se buscan espacios que separen la información de intensidad de la crominancia. En algunos casos, se proponen variaciones de modelos existentes, como en [84], donde se corrigen los tonos claros de YCrCb. Incluso, se han desarrollado modelos de color específicos para el problema, como el espacio Tinta-Saturación-Luma, de Terrillon y otros [176], tratando de compactar los valores asociados al matiz de la piel.

Aparte del espacio usado, la otra cuestión básica relacionada con el color es cómo modelar un determinado tono de forma efectiva. Podemos clasificar los modelos existentes en dos categorías: paramétricos y no paramétricos.

- **Modelos no paramétricos del color.** El método de modelado más sencillo consiste en tomar unos umbrales fijos para cada canal. Por ejemplo, en [59, 60, 170] se aplica este esquema sobre los canales H y S del espacio HSV, mientras que en [25, 24] se toman los canales *Cr* y *Cb* del espacio RGB normalizado. En [39, 106] se usa el mismo espacio, pero modelando el color mediante el histograma conjunto de ambos canales, $h(Cr, Cb)$, para una muestra de entrenamiento. Un píxel es clasificado como de piel si su valor en el histograma está por encima de cierto umbral. Pero los histogramas también se han usado para definir tests piel/no piel de regiones de imagen (no de simples píxeles por separado); por ejemplo, en [160] se aplica una medida de intersección entre histogramas, en el espacio HSV. Para evitar la introducción de umbrales, algunos autores, como [195], aplican lógica difusa.

- **Modelos paramétricos del color.** Frente a las técnicas no paramétricas, la otra gran alternativa es la creación de modelos paramétricos del color, normalmente basados en distribuciones gaussianas. El mecanismo más elemental es la definición de una función de densidad de probabilidad normal, cuya media y varianza es estimada a través de ejemplos [21, 96, 201, 10]. Obviamente, estos modelos dan lugar a fronteras de decisión cuadráticas en el espacio de color correspondiente. No obstante, algunos autores han observado que el color de individuos de diferentes etnias presenta distribuciones no

unimodales. En consecuencia, existen trabajos que modelan el color de piel mediante mezclas de gaussianas [88, 202]. En este caso, los parámetros son obtenidos aplicando el conocido algoritmo EM. Paradójicamente, algunos estudios [90], han demostrado que los modelos basados en histogramas son más adecuados, tanto en precisión como en coste, que los modelos de mezcla.

Los métodos de detección basados en color presentan varios puntos débiles. En primer lugar, la variación de los matices entre diferentes sistemas de adquisición y fuentes de iluminación, además de la perturbación por ruido y por compresión, es tan grande que hace inviable fijar de antemano un espacio *universal* de todos los colores de piel humana. Tal espacio debería ser tan amplio que la clasificación piel/no piel no sería efectiva. Lógicamente, este problema no afecta tanto al uso de color en problemas de seguimiento de caras.

En segundo lugar, incluso un modelo perfecto de color de piel no es suficiente para detectar caras, puesto que también encontraría manos, brazos, cuellos, pies o cualquier región de piel. Por ello, el color se suele usar en combinación con otras características.

Combinación de color, forma y proyecciones

Dentro de los métodos de detección de caras basados en invariantes, los más exitosos se encuentran entre los que hacen uso de diferentes características [204]. El esquema típico de estos métodos es como el siguiente:

1. Clasificar los píxeles en piel/no piel usando un modelo de color.

2. Buscar caras candidatas mediante agrupación de componentes conexos o con *clustering* de los píxeles de color de piel.

3. Aplicar una etapa de verificación de candidatos, basada en comprobar la existencia de ojos, nariz y boca.

Uno de los primeros trabajos apoyados en este esquema es el descrito por Sobottka y Pitas [169, 170]. Para el modelo de color usan el espacio HSV, sobre el cual predefinen unos umbrales para el tono de piel. Después se aplica análisis de componentes conexos, que es seguido de una verificación de tamaño y forma elíptica, en los componentes encontrados, utilizando simples momentos estadísticos. Por último, sobre los candidatos –segmentados con los márgenes de cada componente conexo– se aplica morfología matemática, para mejorar las regiones oscuras, y después se obtiene la proyección vertical. La proyección se suaviza y se buscan sus mínimos significativos –usando el gradiente de la señal–. Para cada mínimo, se calcula una proyección horizontal en una región de 3 píxeles de alta. Las posiciones de los ojos, la nariz y la boca se sitúan simplemente en los mínimos correspondientes de las proyecciones verticales y horizontales, según restricciones de simetría y posición en la cara. En [170], este último proceso se describe como una búsqueda "min-max" usando lógica difusa.

En [59, 60] propusimos una estrategia similar, pero realizando la detección de regiones de piel en los llamados *mapas HIT*, compuestos por: el canal *Hue* de HSV, la intensidad y la magnitud del gradiente (calculada con filtros de Sobel). De esta manera, se espera que el contorno de la cara esté mejor delimitado, ya que muchas veces el pelo o el fondo de la imagen presentan un color similar a la piel, como se puede comprobar en la figura 3.13. La verificación de los candidatos usa también un criterio de forma elíptica e integrales proyectivas. No obstante, en este trabajo las proyecciones se analizan de manera heurística, buscando una estructura predefinida de picos máximos y mínimos, correspondientes a ojos, nariz y boca. Si no se encuentra esa estructura, se rechaza el candidato. Este proceso se ilustra en la parte inferior de la figura 3.13. En una base de datos de 195 imágenes –algunas de las cuales forman ahora parte de la base de caras UMU–, se obtenían unos porcentajes de detección próximos al 70 %, con un 2 % de falsas alarmas por imagen.

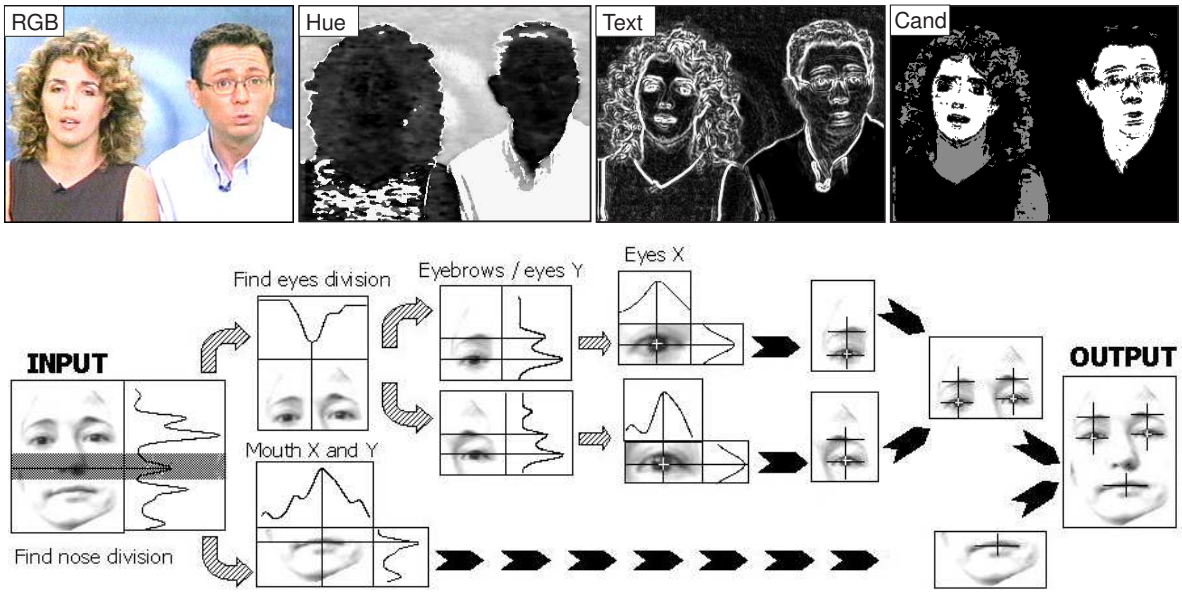


Figura 3.13: *Detección y localización de caras mediante mapas HIT [59]. En la parte superior, el proceso de obtención de candidatos mediante búsqueda de componentes conexos en el espacio HIT (Hue, Intensidad, Textura). En la parte inferior, el paso de verificación de candidatos mediante proyecciones; se aplica una serie de heurísticas para la localización de picos máximos y mínimos.*

Posteriormente, en [62] planteamos una mejora del proceso de verificación con integrales proyectivas, basada en el alineamiento de señales. Se utilizaba una versión previa, y menos robusta, del algoritmo de alineamiento 2.4 (ver la página 74). En un test de clasificación cara/no cara, se presentan ratios de acierto del 95 %. Combinándolo con el proceso de segmentación de regiones conexas por color de piel, el porcentaje de detección llegaba al 86 % con un 1 % de falsos positivos, lo cual supone una mejora bastante significativa respecto de la verificación heurística.

En estos dos últimos trabajos, el borde exterior del rostro se describe mediante un contorno poligonal, obtenido con el algoritmo IPE (*Iterative Point Elimination*) [145]: partiendo del contorno del componente conexo (que consta de un número muy elevado de vértices), se

realiza un proceso iterativo de eliminación de los vértices menos significativos (según el área del triángulo asociado a cada vértice), hasta alcanzar un número reducido de vértices. En la figura 3.14 se muestran algunos ejemplos de segmentaciones resultantes de caras y no caras.

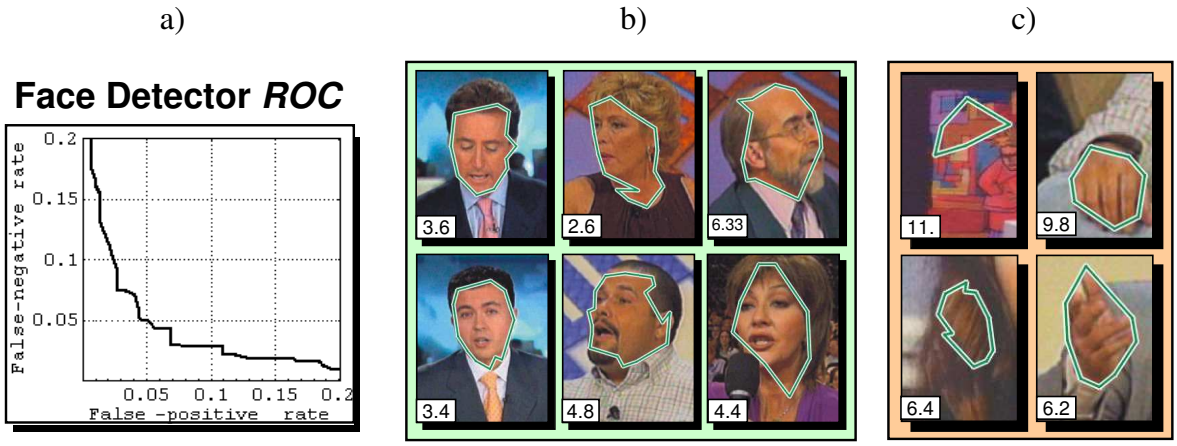


Figura 3.14: *Verificación de caras candidatas con alineamiento de proyecciones [62]. a) Curva ROC de la clasificación cara/no cara resultante. b) Ejemplos de caras segmentadas por color de piel. c) Ejemplos de no caras con color de piel. El número mostrado debajo de cada imagen indica la distancia del candidato al modelo de proyecciones de la cara.*

Otros muchos trabajos han aprovechado las proyecciones asociadas a las caras, aunque fundamentalmente para el problema de localización de los ojos y la boca. Por ello, las discutiremos más profundamente en el siguiente capítulo.

Otros métodos combinados

Nos hemos centrado en los algoritmos que usan integrales proyectivas por su especial relación con las técnicas estudiadas y desarrolladas en la presente tesis. Pero existen otros muchos métodos ascendentes que combinan otros tipos de características. Casi siempre una de ellas es el color, modelado y detectado de las formas más variadas, como ya hemos visto.

Entre las otras características que se aplican junto al color, tenemos la estructura del rostro [195], los momentos geométricos [176], la simetría [157], y la profundidad [96, 41]. En [195], la estructura facial es descrita mediante un patrón reducido (de unos pocos bloques), donde cada bloque debe contener cierta proporción de píxeles de color de piel y de pelo. Para la verificación de candidatos, se buscan bordes horizontales en las regiones de ojos y boca.

Por su parte, en [176] se propone la utilización de momentos geométricos para describir la estructura interna de los componentes de la cara. En concreto, se extraen 11 momentos, que son clasificados mediante redes neuronales. El ratio de detección alcanzado en un conjunto de 100 imágenes es del 85 %. La simetría del rostro ha sido utilizada también como un método de verificación de regiones candidatas. Por ejemplo, en [157] se usa la simetría de los ojos para guiar su localización dentro de la cara. La distancia entre los ojos ayuda a localizar después la boca y la punta de la nariz.

En otro extremo tenemos los trabajos que usan color y profundidad [96, 41]. Se parte de la suposición de que los píxeles del fondo están más alejados que los de la cara. De esta forma, se puede realizar una segmentación fondo/primer plano, basada en el histograma de profundidades de la imagen. Para la detección de caras se añade también la búsqueda de regiones por color de piel.

Debemos aclarar que la detección de regiones de color de piel no siempre se realiza mediante análisis de regiones conexas. Se han propuesto también, como alternativas, la segmentación multiescala [201], y las representaciones de *blobs* [130], donde para cada píxel se crea un vector de 4 dimensiones, formado por los componentes X, Y, y los valores de los canales R y G normalizados; sobre ellos se aplican después técnicas de *clustering*.

3.2.3. Métodos basados en patrones predefinidos

Los algoritmos basados en búsqueda de patrones (en inglés, *template matching*) se apoyan en modelos de caras definidos manualmente por un humano, ya sean fijos o parametrizados. La detección facial se basa en aplicar una medida de correlación del patrón sobre la imagen, determinando la validez del candidato según el valor resultante del cálculo.

Este simple esquema supone que la imagen sólo contiene una cara, centrada y bien alineada; es más, algunos métodos de este grupo pueden ser interesantes en la localización de componentes faciales, como veremos en el capítulo 4. Para permitir múltiples detecciones e invarianza frente a posición, escala y los demás factores que discutimos en el apartado 3.1.1, se aplican procesos de búsqueda multirresolución, multiescala, composición de subpatrones y definición de patrones deformables.

Normalmente, los acercamientos dentro de esta categoría presentan problemas parecidos a los de los métodos descendentes. Ambos descansan en el conocimiento "experto" del diseñador, por lo que se ven sujetos a una limitada capacidad de generalización frente a condiciones imprevistas. Algunos trabajos recientes han retomado la filosofía de los patrones predefinidos, aunque en general no es una de las líneas más activas de investigación.

Vamos a describir seguidamente algunos de los detectores más interesantes basados en patrones. Distinguimos dos subcategorías: los que manejan patrones predefinidos, y los que permiten patrones deformables.

Patrones predefinidos

Dentro de los métodos con patrones predefinidos, el uso de operadores de bordes es una de las elecciones más frecuentes. Es decir, no se crean modelos de las imágenes de intensidad, sino de las imágenes de bordes. En este grupo se pueden situar algunos de los trabajos pioneros en detección de caras, como los expuestos en [37, 158]. En concreto, en [37] se utilizan filtros de Sobel, como el mostrado en la figura 3.12c); los bordes obtenidos son agrupados y después se buscan instancias de un modelo de contorno facial. El proceso se repite a mayor resolución para las cejas, los ojos y los labios. En [71] se utiliza el operador de bordes de Marr-

Hildreth; el modelo consta de características extraídas de los bordes, que definen posiciones, curvaturas y restricciones geométricas. Se realiza un proceso de agrupación de contornos, y se buscan tres curvas que puedan corresponder a los bordes de la cara (lado derecho, izquierdo y línea del pelo). El porcentaje de detección[2] se estima en un 70 %. En [186] se usa un acercamiento parecido pero utilizando filtros de *wavelet*.

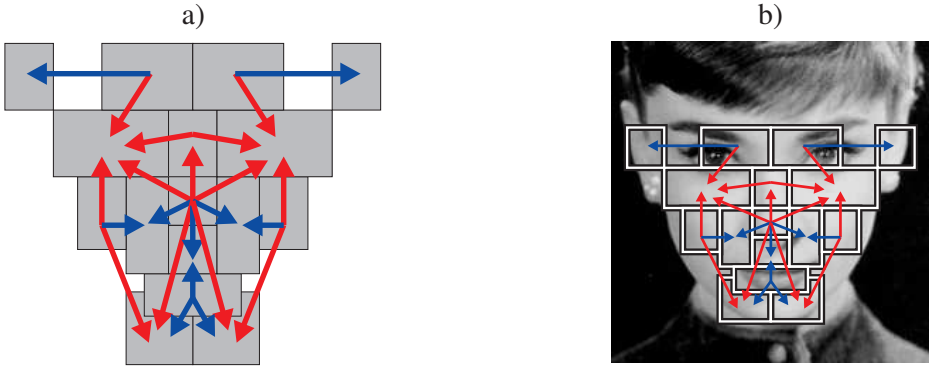


Figura 3.15: *Detección de caras mediante modelos de patrones predefinidos, propuesto en [161]. a) Modelo de bloques y relaciones entre los mismos. La relación significa "es más oscuro que", de carácter esencial (en rojo) y secundarias (en azul). b) Aplicación del método sobre la imagen "audrey2.gif".*

Otra forma de aplicar patrones basados en bordes es dividiendo la cara en bloques. En [182], el modelo del rostro está compuesto por un número reducido de bloques, donde cada uno contiene información de luminosidad y nivel de bordes del mismo. Se encontrará una cara cuando sus bordes correspondan con el modelo de bloques definido. Por otro lado, podemos encontrar trabajos que usan el operador de Laplace [124], donde el modelo corresponde a los bordes generados por los elementos faciales. En este último método, la búsqueda del patrón se repite para diferentes resoluciones y ángulos, entre -20º y 20º.

En algunos detectores, los patrones se basan en las imágenes de intensidad, como en el propuesto en [166]; para tener en cuenta los diferentes niveles de iluminación, no se usa directamente el valor de gris, sino el brillo relativo entre diferentes regiones. Un ejemplo es la propuesta de [161], mostrada en la figura 3.15, donde se define un modelo de cara compuesto por 16 regiones (que *grosso modo* corresponden a ojos, nariz, frente, mejillas, etc.) y se establecen relaciones entre pares de regiones del tipo "es más oscura que". Hay que aclarar que este método está más orientado a localización que a detección. La cara es aceptada cuando la imagen cumple la mayor parte de las relaciones especificadas en el patrón. Este método fue extendido posteriormente a una representación basada en *wavelets* [134].

Finalmente, los modelos predefinidos han sido también creados sobre las integrales proyectivas extraídas de la cara [59, 60, 213]. No obstante, las técnicas que los usan están orientadas fundamentalmente a los problemas de localización de componentes faciales, por lo que serán revisadas más detenidamente en el capítulo 4.

[2]Es curioso el método que proponen los autores para decidir el número de caras presentes en una imagen. Suponiendo que la imagen es una fotografía de un periódico, utilizan el texto a pie de foto para encontrar el número de personas.

Patrones deformables

Los patrones deformables permiten cierta flexibilidad en el ajuste de un modelo de cara. Estos métodos definen una *función de energía*, que mide el grado de ajuste en cada momento. Básicamente, la energía consta de dos componentes: uno *interno* y otro *externo*. El externo tiende a ajustar al máximo el modelo a la instancia actual, mientras que el interno hace que el ajuste actual mantenga una forma coherente y compacta. La localización se consigue buscando la configuración que minimiza la energía total.

Por ejemplo, en [210] Yuille y otros presenta un modelo compuesto por bordes, picos y valles; la función de energía enlaza estos elementos con diferentes pesos. Una desventaja de esta técnica –y de otras dentro de este grupo– es que el modelo debe ser colocado inicialmente en la proximidad de la cara. En [102] se propone un método basado en *snakes* para localizar el borde exterior de la cabeza. Se aplican operaciones morfológicas y de suavizado. Esto genera un conjunto de candidatos, que después son comprobados con patrones deformables parecidos a los de [210].

Otro de los mecanismos más populares son los *modelos de distribución de puntos* (PDM), introducidos por Lanitis y otros en [105]. Estos modelos consisten en vectores de puntos, que son entrenados manualmente situándolos en los contornos de ojos, nariz, boca y barbilla. De acuerdo con los ejemplos de entrenamiento, se deducen los posibles modos de variación del patrón deformable y los valores de intensidad asociados a cada punto. El ajuste del modelo a una nueva cara se hace mediante la técnica denominada ASM (*active shape model*) [34]. Esta técnica permite la normalización de la cara, mediante una deformación asociada al PDM, a un rostro de forma media. Cabe destacar que –como ya hemos referido para otros muchos casos– esta técnica está destinada a problemas de localización y seguimiento [47]; en consecuencia, los describiremos detalladamente en el siguiente capítulo.

3.2.4. Métodos basados en apariencia

Si recapitulamos las técnicas expuestas hasta ahora, podemos observar que la mayoría se centran en la detección de una única cara en las imágenes, con una buena resolución y, en la mayoría de los casos, mirando de frente. Siempre que se usen bordes, por ejemplo, será necesario trabajar con caras de cierto tamaño, en las que no aparezcan sombras, oclusiones ni expresiones faciales muy destacadas. Y en el caso de los métodos que usan color, se debe garantizar la constancia de los matices, lo cual no siempre es posible si no se puede controlar directamente el sistema de adquisición.

Frente a ellas, las técnicas basadas en apariencia vienen a paliar estas limitaciones, y actualmente se encuentran entre las más exitosas [204]. Básicamente, la idea subyacente consiste en reducir la detección a un problema de *clasificación cara/no cara*, que se repite de forma exhaustiva para todas las posibles subregiones y con todas las escalas admitidas [108]. Además, a diferencia de los métodos basados en patrones predefinidos –creados a partir del conocimiento "experto" del investigador–, los modelos se obtienen mediante entrenamiento sobre

un conjunto de imágenes de entrada. De esta forma, la cuestión fundamental es entrenar un clasificador binario a partir de un conjunto amplio de ejemplos de caras y de no caras. Por ello, a veces estas técnicas se denominan también *basadas en aprendizaje* [108].

Los resultados prácticos de estos métodos se derivan de la filosofía subyacente:

- Son capaces de manejar situaciones complejas, con imágenes de escasa resolución, baja calidad y un número arbitrario de rostros por imagen.
- Pueden presentar un riesgo de sobre-ajuste a los datos de aprendizaje, de manera que el entrenamiento del clasificador se convierte en el aspecto clave.
- El funcionamiento exhaustivo del proceso implica un elevado coste computacional que, en general, será uno de los principales inconvenientes de este acercamiento.

Vamos a describir, en primer lugar, la estructura común de estos métodos. Posteriormente detallaremos algunos de los trabajos más relevantes.

Esquema global de los métodos basados en apariencia

A grandes rasgos, todos los detectores basados en apariencia presentan un funcionamiento similar. Como hemos visto, el núcleo de estos algoritmos es una clasificación cara/no cara aplicada sobre subregiones de tamaño muy reducido. Tales clasificadores deben ser lo suficientemente complejos como para manejar la enorme variabilidad del objeto cara. Para conseguir invarianza frente a posición y escala, el proceso se repite en todas las posibles localizaciones y tamaños de las imágenes. La figura 3.16 muestra el esquema global de un detector genérico basado en apariencia.

Búsqueda multiescala El primer elemento del proceso es una búsqueda exhaustiva multiescala. A partir de la imagen de entrada, se obtiene una *pirámide* de imágenes con diferentes resoluciones. En concreto, se define un factor de escala f, de manera que cada imagen de la pirámide es f veces menor que la anterior. Los valores usados típicamente son del tipo $f = 1,1$ ó $f = 1,2$. Dentro del conjunto de imágenes generado, se analizan todas las subregiones posibles de cierto tamaño de *ventana* en cualquier posición. Los tamaños habituales de ventana[3] suelen estar entre 16×16 y 32×32 píxeles, aunque algunos trabajos llegan a afirmar que el tamaño óptimo es de 20×20 píxeles [109].

Preprocesamiento y foco de atención Antes de la clasificación en sí, muchos métodos incluyen una etapa inicial de preprocesamiento de las ventanas obtenidas. Este paso puede tener dos propósitos:

- Descartar regiones en las que claramente no existe una cara. Para ello se suelen aplicar heurísticas sencillas basadas en bordes (eliminar regiones que sean muy uniformes

[3] Nótese que estos tamaños de ventana determinan, en principio, el tamaño mínimo de las caras detectables.

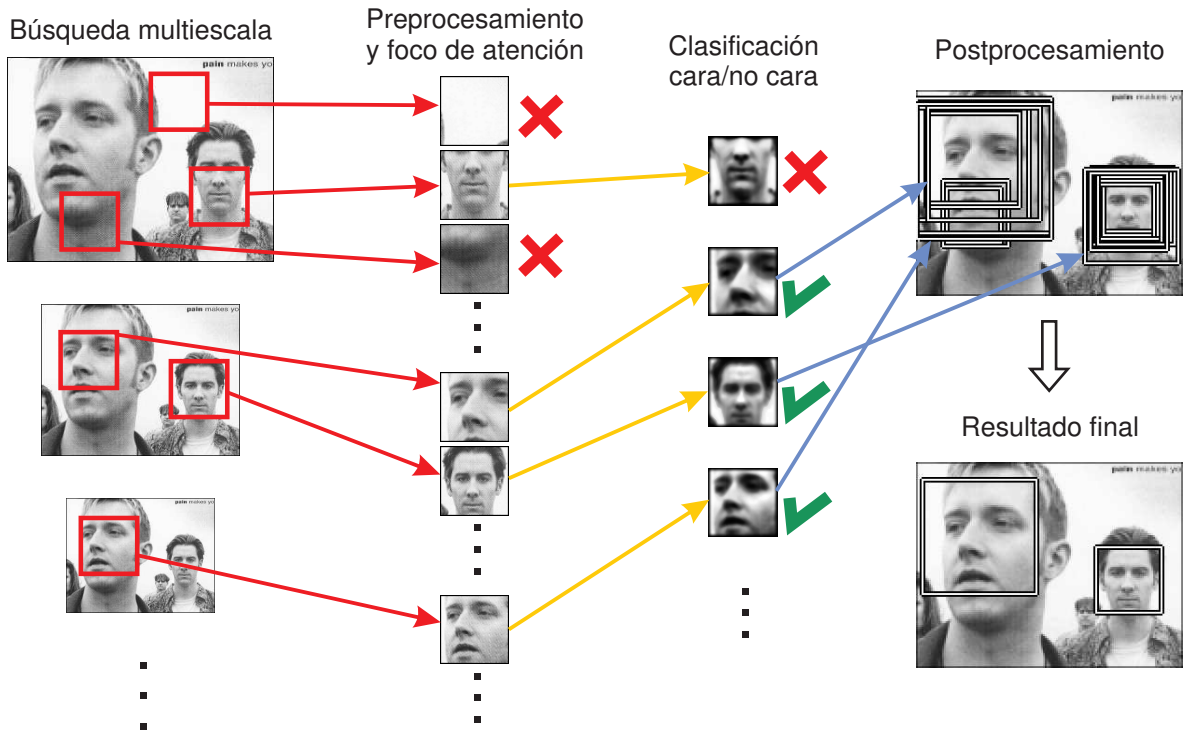


Figura 3.16: *Estructura genérica de un detector de caras basado en apariencia. Primero tiene lugar un proceso de búsqueda multiescala, en el que se analizan todas las subregiones con diferentes resoluciones y en todas las posiciones. Opcionalmente se puede aplicar un preprocesamiento a esas regiones, antes de someterlas un clasificador cara/no cara. Sobre los candidatos validados por el clasificador, se aplica un postproceso de agrupación y selección de los más fiables.*

[110]), brillo (regiones muy claras o muy oscuras), o color (considerar sólo los trozos con color de piel). Las ventanas que no cumplan estos criterios no se siguen estudiando. De esta manera, se consigue reducir el tiempo de ejecución y se disminuye el riesgo de aparición de falsos positivos.

- Mejorar las imágenes de las ventanas de cara a la etapa posterior de clasificación. Fundamentalmente, se pretenden dos cosas: aumentar el contraste de las subimágenes (mediante normalización del brillo), y reducir el posible efecto de sombras no uniformes (por ejemplo, con un modelo lineal del fondo [173, 152]).

Ambos procesos pueden compartir los mismos cálculos, que deben poder ejecutarse de forma muy eficiente; hay que tener en cuenta que en cualquier imagen de tamaño medio aparecerán muchos miles de estas subregiones[4]. El método más habitual de preprocesamiento de las ventanas es el propuesto por Sung y Poggio [173, 174], que consta de tres pasos: (1) calcular un modelo lineal del fondo, según los niveles de gris de la ventana; (2) restar el modelo lineal a los píxeles; y (3) aplicar una ecualización del histograma al resultado.

[4]Por ejemplo, en una imagen *pequeña* de 320×240 píxeles, con ventanas de 20×20 y factor de escala $f = 1,2$, el número total de subregiones, o ventanas, supera las 196.000. Y para 640×480 llega casi al millón.

Algunos autores proponen un muestreo menos exhaustivo –por ejemplo, mover las ventanas de n en n píxeles [153]– en aplicaciones donde el tiempo de ejecución sea crítico. De todas formas, la necesidad de aplicar muchas veces la clasificación cara/no cara es una de constantes de los métodos basados en apariencia.

Clasificación cara/no cara Las ventanas resultantes son la entrada de los mecanismos de clasificación subyacentes. Estos clasificadores deben ser capaces de modelar la compleja forma del objeto cara y al mismo tiempo tener una ejecución extremadamente rápida.

Podemos formular el problema dentro de un marco probabilístico. Las ventanas se consideran como variables aleatorias, x, y las clases *cara* y *no-cara* como dos distribuciones de probabilidad. El objetivo es encontrar una forma para las funciones de densidad de probabilidad condicionadas, $p(x|cara)$ y $p(x|no\text{-}cara)$. La dificultad subyace en la elevada dimensionalidad de x, y en los múltiples modos de variación posibles de la clase *cara*.

Para resolver esta cuestión se han aplicado muchos métodos diferentes de clasificación: redes neuronales [153, 152], discriminantes lineales de Fisher [9, 81], máquinas de vectores de soporte [131, 150], reducción a subespacios lineales [183, 125, 173, 174], combinación de clasificadores elementales [151], como los métodos *AdaBoost* [188, 110, 112], etc.

Agrupación de candidatos y postprocesamiento Como resultado de la clasificación de las ventanas, tenemos un conjunto de posibles regiones de cara. Muchas de ellas pueden estar asociadas a una misma cara, como se puede ver en el ejemplo de la figura 3.16, y algunas de ellas pueden ser falsas detecciones. El último paso de los detectores basados en apariencia consiste en agrupar las regiones con un alto solapamiento y eliminar los candidatos menos fiables. Para ello se aplica un simple recorrido, en el que se forman grupos de regiones muy próximas entre sí. Posteriormente, el número de regiones por grupo se suele usar como un criterio heurístico para eliminar muchos de los falsos positivos [110, 152]. Se espera que los grupos asociados a caras reales den lugar a muchas regiones candidatas, mientras que en las falsas detecciones aparezcan unas pocas.

En los siguientes puntos vamos a hacer un repaso de los mecanismos de clasificación que han demostrado mejores resultados para el problema de distinguir caras de no caras. En todos ellos damos por supuesto el esquema genérico de los métodos basados en apariencia, por lo que describimos la clasificación de ventanas individuales. No pretendemos aquí llevar a cabo una recopilación exhaustiva de publicaciones, sino centrarnos en los trabajos con mayor impacto en el contexto de la detección facial.

Clasificación basada en distribuciones de probabilidad

Supongamos que el tamaño de las ventanas es de $n \times n$ píxeles. Los ejemplos, x, se pueden ver como vectores en el espacio $\mathbb{R}^{n^2}$. Una posible forma de clasificar los patrones consiste en modelar la distribución de las caras y las no caras en este espacio multidimensional.

Autocaras y autocomponentes Uno de los primeros trabajos en usar esta idea es el de Turk y Pentland [183], donde se introduce por primera vez el concepto de *autocara*. Sea X una matriz de $m \times n^2$ con todos los ejemplos de caras (donde m es el número total de ellos); las autocaras son simplemente los vectores propios de X, o equivalentemente, los autovectores de $X^{\mathrm{T}}X$. Esta descomposición permite una proyección de los ejemplos, x, en el autoespacio definido, dando lugar a vectores de reducida dimensión: $y = Fx$, donde F es una matriz de $k \times n^2$ con los k autovectores de mayor autovalor asociado. Turk y Pentland definen una métrica de distancia al autoespacio de las caras. Si la distancia obtenida para cierta ventana está por debajo de un umbral, se dice que se ha detectado una cara.

Posteriormente, Moghaddam y Pentland [125], dividen la medida de distancia en dos componentes: la distancia dentro del autoespacio (DIFS, *distance in feature space*); y su complemento ortogonal, la distancia al autoespacio (DFFS, *distance from feature space*). La segunda es equivalente al error de reconstrucción, es decir $||x - F^{\mathrm{T}}y||$. Para la primera se crea un modelo de mezcla de gaussianas multivariables; sin embargo, no llegan a aplicar su método al caso de detección, sólo a localización, codificación y reconocimiento de personas.

Distribución de caras y no caras A diferencia de las propuestas anteriores, Sung y Poggio [173, 174], demuestran el interés de modelar tanto la distribución de las caras como de las no caras. En concreto, estos autores trabajan con ventanas de 19×19 píxeles y proyección en autoespacios. El sistema propuesto tiene las siguientes características:

- Los ejemplos de caras se agrupan, mediante un algoritmo de k medias, en 6 grupos o *clusters*. Igualmente, las no caras se agrupan en otros 6 *clusters*.

- De forma similar a [125], se definen dos distancias entre cada ejemplo, x, y cada uno de los 12 clusters. La primera es una distancia de Mahalanobis entre el centroide del cluster y la proyección de x en el autoespacio asociado al cluster, compuesto por los 75 primeros autovectores. La segunda es similar al error de reconstrucción (DFFS), pero específico para el subespacio asociado a cada cluster.

- El resultado de lo anterior es un vector de 24 distancias asociado a cada ejemplo. El último paso consiste en hacer una clasificación cara/no cara a partir de esas distancias. Para ello, utilizan un perceptrón multicapa entrenado con el algoritmo clásico de retropropagación, tomando un conjunto de unos 4.000 ejemplos de caras y 43.000 de no caras.

Sobre la base MIT (con 23 imágenes y 136 caras), alcanzan unos ratios de detección del 81,9 % con 19 falsas detecciones. Aunque estos resultados se mejorarán posteriormente, son unos excelentes valores considerando la complejidad del conjunto.

Aparte del mecanismo de clasificación, los trabajos de Sung y Poggio resultan de una enorme relevancia por la introducción de algunos mecanismos que posteriormente han sido adoptados por muchos autores. Uno de ellos, ya lo hemos comentado, es el método de

preprocesamiento de ventanas con compensación de un modelo lineal y ecualización del histograma. Otras dos ideas interesantes están orientadas al punto clave de obtener conjuntos representativos de las clases cara y no cara:

- **Generación de ejemplos de caras virtuales.** En general, todos los métodos basados en apariencia necesitan un número muy grande de ejemplos para ser entrenados convenientemente. Pero la disponibilidad de ejemplos de caras resulta muy reducida. Para resolver este problema, Sung y Poggio definen un modo de multiplicar las muestras de caras disponibles. La idea consiste en reflejar las imágenes dadas, y aplicar escalas y rotaciones en cantidades muy reducidas. Por ejemplo, se pueden realizar desplazamientos de $\pm 1/2$ píxel, aumentos y reducciones del 10 %, y rotaciones de $\pm 10^{o}$. Los valores usados para estas trasformaciones se establecen de forma aleatoria. En definitiva, es un mecanismo que permite crear una galería de ejemplos virtuales arbitrariamente grande.

- **Mejora del entrenamiento con *bootstrapping*.** En el caso de las no caras, el problema no es tanto conseguir muchos ejemplos, sino que éstos sean representativos. En principio, interesa usar los que más se parezcan a caras. Para ello, definen una estrategia, llamada *bootstrapping*, en la que el entrenamiento se convierte en un proceso iterativo: (1) se entrena el clasificador; (2) se aplica sobre una serie de imágenes que no contienen caras; (3) si se encuentran falsas detecciones, se añaden al conjunto de no caras y se vuelve al paso inicial. La mejora que se consigue con este método es bastante significativa. Se pueden ver algunos ejemplos típicos obtenidos con esta técnica en la figura 3.17.

Figura 3.17: *Ejemplos de caras (izquierda) y de no caras (derecha) obtenidos mediante la aplicación de una técnica de bootstrapping. Las imágenes han sido tomadas de la página web [204]:* `http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html`

Otras técnicas de proyección en subespacios Yang y otros [203], proponen dos métodos basados en subespacios alternativos al uso de PCA. El primero consisten en aplicar la idea de los *discriminantes lineales de Fisher* (LDA). Se argumenta que mientras PCA está orientado a reconstrucción, LDA es más apropiado para clasificación binaria [81]. En particular, utilizan tamaños de ventana de 20×20 píxeles. Como en [173, 174], se realiza una agrupación no supervisada de los patrones de cara y no cara; en este caso en 25 clusters de cara y otros 25 de no cara, para lo cual se utilizan los mapas auto-organizativos de Kohonen (SOM) [99]. Para cada cluster se crea un modelo de la distribución de probabilidad del mismo, a través de una

proyección que minimiza la varianza *intra-clase* y maximiza la *inter-clase*. La discriminación cara/no cara se basa en la regla de máxima verosimilitud, es decir, seleccionando la clase del cluster con mayor probabilidad.

El otro método propuesto en [203] utiliza la formulación de los conocidos como *analizadores de factor* (FA, *factor analyzer*). Esta técnica se puede entender como una extensión de PCA, donde se añade un componente para modelar el ruido. Así, los ejemplos x se suponen generados mediante un modelo del tipo: $x = \Lambda z + u$; donde u es una variable de ruido gaussiano; z es la proyección de x en el subespacio; y Λ es la base. Yang y otros proponen modelar la distribución de las caras con un modelo de mezcla basado en analizadores de factores, cuyos parámetros son entrenados con el algoritmo EM. El uso de FA tiene dos ventajas sobre PCA: es más robusto al ruido, y describe mejor la densidad de probabilidad de las clases.

Los resultados de ambos métodos presentados en [203] sobre la base de caras MIT consiguen alrededor del 90 % de detección, sin superar las 3 falsas alarmas. En la base CMU/MIT alcanzan sobre el 93 %, para un 65 % de falsas alarmas por imagen. Los ratios son siempre ligeramente mejores para el método basado en LDA.

Clasificación basada en redes neuronales

Aunque se han creado muchos algoritmos de detección de caras basados en redes neuronales, las aportaciones de Henry Rowley [153, 152], están consideradas como las más importantes dentro de esta categoría. En muchos aspectos, el trabajo de Rowley parte de los resultados de Sung y Poggio [173, 174]: preprocesamiento de las ventanas, generación de ejemplos virtuales, y uso de *bootstrapping*.

La principal novedad es que las redes neuronales se aplican directamente sobre las ventanas, de 20×20 píxeles, obtenidas en el preproceso. En concreto, se utilizan perceptrones multicapa con una capa oculta de 26 de unidades. Las conexiones de estas unidades se establecen de antemano, tal y como se puede ver en la figura 3.18: 4 unidades se conectan a subregiones de 10×10; 16 a regiones de 5×5; y 6 a bandas horizontales solapadas de 20×5 píxeles. El entrenamiento se realiza con el algoritmo de retropropagación clásico. El resultado final es una unidad, que debe valer 1 para las caras y -1 para las no caras.

Puesto que los pesos resultantes de la red pueden depender de los ejemplos de entrenamiento (e incluso del orden en el que se introducen en la red), se aplican varias redes de este tipo para cada posible ventana. Sobre sus resultados, se estable un esquema de *arbitraje* que puede ser de distintos tipos: AND - se detecta cara si todas las redes producen un valor mayor que 0; OR - alguna de las redes debe devolver mayor que 0; votación - la mayoría de las redes deben producir un valor positivo. Se plantea también la posibilidad de realizar este arbitraje mediante otra red neuronal.

En principio, el proceso está diseñado para detectar caras de frente o con una reducida inclinación. Para conseguir invarianza frente al segundo factor, Rowley propone una *red enrutadora*, que dada una ventana indica el ángulo de inclinación más probable, entre 0 y 360º,

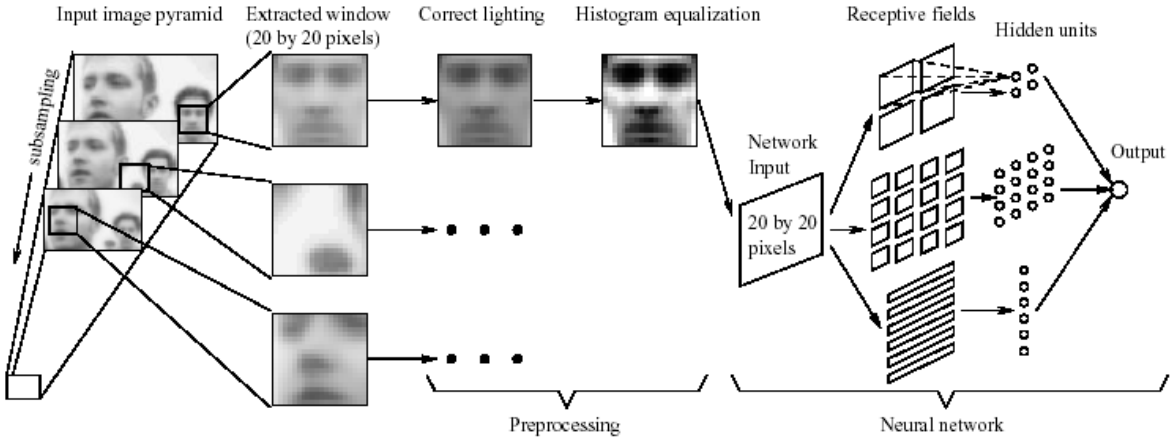


Figura 3.18: *Detección de caras mediante redes neuronales. Extraído de [152]. (c) Henry Rowley.*

con incrementos de 10^o. De esta forma, se pueden detectar caras con cualquier orientación respecto del plano de la imagen. No obstante, puesto que la red no es perfecta, el rendimiento baja ligeramente cuando se aplica sobre caras no inclinadas.

Los resultados sobre las bases de caras MIT y CMU/MIT son bastante interesantes, aunque en general adolecen de un alto número de falsos positivos. Por ejemplo, en la base MIT el ratio de detección es parecido al de los métodos de Yang y otros [203], pero con un total de 42 falsas detecciones. No obstante, una de las claves de la popularidad del detector de Rowley es haber ofrecido acceso público a su implementación en C, que se puede encontrar en:

`http://www.cs.cmu.edu/~har`

Gracias a ello, será incluido en los experimentos comparativos de la sección 3.4.

Clasificación basada en máquinas de vectores de soporte

Desde el punto de vista de la clasificación binaria cara/no cara de las ventanas, la aplicación de las máquinas de vectores de soporte (SVM) [185], a la detección facial surge de manera bastante natural. Así, la primera utilización de SVM en el problema es debida a Osuna y otros [131]. El funcionamiento toma también muchas de las propuestas de Sung y Poggio [173, 174], y utiliza ventanas de cara y no cara de 19×19 píxeles como la entrada al clasificador. La función de *kernel* utilizada es un polinomio de segundo grado.

Una cuestión que no se debe obviar en estos métodos es la eficiencia computacional. En [131] se utilizan 50.000 patrones de cara y no cara. Eso implica un enorme coste de entrenamiento y de aplicación de SVM. Para paliar el problema, los autores describen un método de entrenamiento eficiente con conjuntos de datos muy grandes. Más recientemente, Romdhani y otros [150], proponen también la reducción del número de vectores de soporte utilizados, con el fin de disminuir los tiempos de aplicación del clasificador.

Los resultados de [131] para una base propia con 313 imágenes y una cara por imagen son bastante interesantes. Se alcanza un 97 % de detección con sólo 4 falsas alarmas. El rendimien-

to disminuye para el caso más complejo de la base MIT, donde sólo llega a un 72,4 %, con un 87 % de falsos positivos por imagen. A pesar de ello, se argumenta que la detección es unas 30 veces más rápida con SVM que con la propuesta de Sung y Poggio.

Clasificación basada en redes dispersas de *winnows*

La arquitectura de redes dispersas de *winnows* (SNoW, *sparse network of winnows*), fue propuesta originalmente por Roth [151], para la resolución de problemas en los que interviene un número muy elevado de características binarias, algunas de las cuales pueden ser relevantes y otras no. El término *winnow* (en español, "aventar" o "ahechar"), sugiere la idea del método, consistente en descartar o añadir características individuales, según su contribución observada en la resolución del problema.

Dado un vector de características booleanas, $c = \{c_1, c_2, ..., c_t\}$, (fácilmente con t por encima de cien mil) existe una matriz de ponderaciones, $w = \{w_1, w_2, ..., w_t\}$, que indica la relevancia de cada característica. La clasificación se realiza mediante una simple regla multiplicativa. Si $w \cdot c$ es mayor que un umbral, τ, se declara que c es positivo y en otro caso negativo. La novedad introducida por Roth se encuentra en el mecanismo de aprendizaje, que consiste en un proceso iterativo de corrección de los pesos. Si un ejemplo de entrenamiento, c, es clasificado de forma incorrecta, entonces se procede a corregir los pesos:

- Si c debería ser declarado como positivo (y se ha clasificado erróneamente como negativo), se aumentan todos los pesos w_i para los valores de i tales que c_i valga 1.
- Si c debería ser declarado como negativo, se decrementan todos los pesos w_i para los valores de i tales que c_i valga 1.

Típicamente, el aumento/decremento del peso consiste en multiplicar o dividir por 2, respectivamente. Al final del proceso, existirá un conjunto de características activas y otras descartadas completamente, esto es, con ponderación casi nula.

Yang, Roth y otros [205], plantearon una posible aplicación de la arquitectura SNoW al problema de la detección de caras, informando de unos resultados muy positivos sobre la base CMU/MIT. En particular, crean dos clasificadores SNoW, donde las características se obtienen combinando posición, intensidad media y varianza (discretizadas a un conjunto reducido de valores), de subregiones de la ventana de 10×10, 4×4, 2×2 y 1×1 píxeles. En consecuencia, el vector de características consta de 135.424 valores booleanos.

Sobre las imágenes de la base CMU/MIT alcanzan unos resultados muy competitivos, por encima del 94 % de detección para menos del 70 % de falsas alarmas por imagen. No obstante, los autores echan en falta un mecanismo más potente de obtención de las características.

Clasificación basada en Naive Bayes y apariencia local

La regla de decisión de Bayes ha sido aplicada al problema de detección de caras por Schneiderman y Kanade [162]. Básicamente, la regla viene a indicar que la ventana x debe ser

clasificada como cara cuando se cumpla que:

$$\frac{P(x|cara)}{P(x|no-cara)} > \frac{P(no-cara)}{P(cara)} \tag{3.1}$$

Si las representaciones de $P(x|cara)$ y $P(x|no-cara)$ son precisas, el criterio de Bayes es óptimo. El método desarrollado en [162] se denomina "Naive Bayes" porque las funciones de densidad de probabilidad son modeladas por partes (en subregiones de la ventana) suponiendo que son estadísticamente independientes. Es decir, si x es descompuesto en los fragmentos $x_1, x_2, ..., x_n$, las probabilidades buscadas serían de la forma:

$$P(x|cara) = \prod_{i=1}^{n} P(x_i|cara) \; ; \; P(x|no-cara) = \prod_{i=1}^{n} P(x_i|cara-no) \tag{3.2}$$

Por ejemplo, se proponen tamaños de ventana de 64×64 píxeles, con subregiones de 16×16. Para la obtención de las probabilidades de subregión se plantean dos métodos alternativos: uno de ellos basado en la aplicación de PCA sobre los trozos; y el otro utilizando filtros de *wavelet*.

A pesar de la simplificación de suponer independencia entre las regiones, los resultados de este método sobre la base CMU/MIT se encuentran al nivel de los mejores métodos descritos previamente. No obstante, el mayor obstáculo de esta técnica es su elevada carga computacional, siendo varios órdenes de magnitud superior a los otros.

Clasificación basada en filtros de Haar y AdaBoost

Hemos dejado para el final la descripción del que es, probablemente, uno de los trabajos más populares en el campo de la detección de caras. A su buen compromiso entre ratios de detección y eficiencia computacional, se une el hecho de haber sido incorporado a las librerías de código abierto Intel OpenCV [35], de forma que está accesible de públicamente para toda la comunidad investigadora. El método fue propuesto por Viola y Jones [188], y mejorado después por Lienhart y Maydt [110]. Estos segundos son también los autores de la implementación disponible en OpenCV.

El algoritmo AdaBoost busca una combinación óptima de un conjunto de *clasificadores débiles* con el fin de crear uno más potente. El entrenamiento consiste en un proceso iterativo, en el que se van seleccionando y dando pesos a los clasificadores elementales, según los propios ejemplos de entrenamiento. Sean x_i las muestras de entrenamiento; X el conjunto de todos los ejemplos; $h_j : X \rightarrow \{+1, -1\}$ los clasificadores débiles posibles, con $j = 1, \ldots, J$; α_t la relevancia de cada clasificador débil seleccionado, con $t = 1, \ldots, T$; y T el número de clasificadores débiles que se quieren en el resultado. Se define una función, $D_t(i)$, para controlar la "dificultad" implícita de cada ejemplo x_i en el paso t. A grandes rasgos, el funcionamiento del algoritmo AdaBoost es como el siguiente:

1. Inicializar los valores $D_1(i)$ con valores constantes, para todo i.

2. Para $t = 1, \ldots, T$ hacer:

   a) Normalizar los valores de $D_t(i)$ para que la suma total sea 1.

   b) Entrenar los clasificadores débiles, h_j, para los ejemplos de X, según los pesos de $D_t(i)$. Para cada clasificador, calcular el error de h_j sobre los datos de prueba, ε_j (como la clasificación es binaria, el error siempre será menor que 0,5).

   c) Seleccionar el clasificador débil, h_t, que produzca el menor error, ε_t.

   d) El peso del clasificador débil, h_t, viene dado por: $\alpha_t = 1/2 \ln((1 - \varepsilon_t)/\varepsilon_t)$.

   e) Actualizar la función de relevancia de los ejemplos, $D_{t+1}(i)$. De forma resumida, los ejemplos que han sido bien clasificados disminuyen su relevancia, y los que han sido erróneos la aumentan.

3. El clasificador combinado resultante es: $H(x) = signo(\sum_t \alpha_t h_t(x))$.

La propuesta de Viola y Jones [188], consiste en que los clasificadores débiles se obtengan mediante la aplicación de *wavelets* inspirados en los *filtros de Haar*. Básicamente, se trata de un operador definido en una subregión rectangular de la ventana, donde una parte de los píxeles se suman y los otros se restan. Si el valor del filtro está por encima de cierto umbral, se acepta (o se rechaza) el candidato. En la figura 3.19a) se pueden ver algunos de estos filtros. La utilización de *imágenes integrales* –que veremos en el apartado 3.3.2– permite obtener muy rápidamente el resultado del filtro. Sobre el valor calculado se establece un umbral, fijando así el resultado final del clasificador débil.

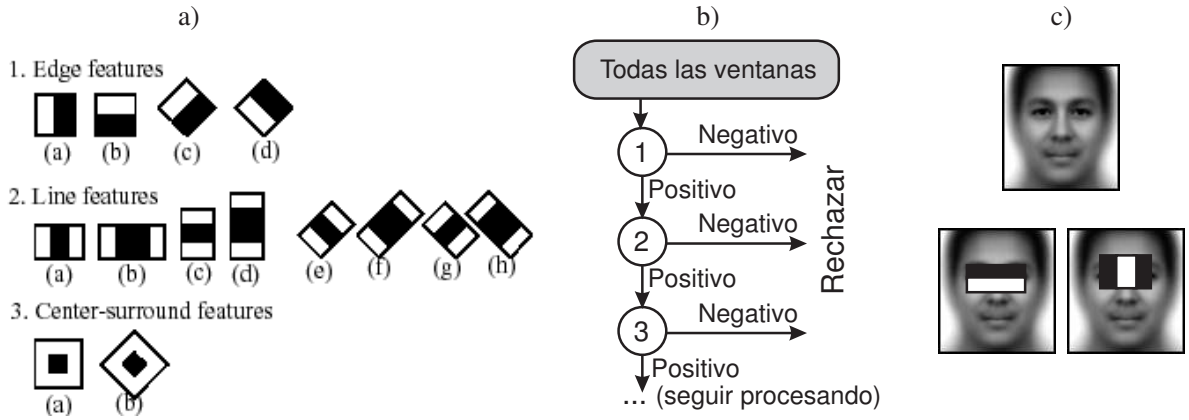


Figura 3.19: *Detección de caras mediante filtros de Haar y AdaBoost [188, 110]. a) Los tipos de características usados para los clasificadores elementales. b) Esquema del clasificador en cascada. Una ventana se clasifica como cara si es aceptada por todos los clasificadores combinados. c) Una cara media y las dos características más discriminantes seleccionadas en el entrenamiento.*

Es más, Viola y Jones utilizan varios de los clasificadores combinados –del orden de unos 32–, que son ejecutados en un *proceso en cascada*, como se muestra en la figura 3.19b). Es decir, se aplican de forma secuencial. Si alguno de ellos produce respuesta negativa (no cara), se

detiene el proceso. Se detecta una cara cuando toda la cascada de clasificadores devuelve resultado positivo. El entrenamiento de la cascada utiliza también el algoritmo AdaBoost.

En sus experimentos, los autores informan de unos ratios de detección ligeramente inferiores a los de otros métodos, como el detector de Schneiderman y Kanade [162], y el basado en SNoW de Yang y otros [205]; y ligeramente superior al de redes neuronales de Rowley y otros [152]. Sin embargo, argumentan que su algoritmo es varios órdenes de magnitud más rápido. En concreto, según los datos que aportan, es 15 veces más rápido que el primero y 600 veces más que el segundo. Estos datos coinciden, aproximadamente, con los resultados de nuestros experimentos, que presentamos en la sección 3.4.

Con posterioridad han aparecido numerosas variaciones y extensiones del método básico, por ejemplo, para detectar caras con diferentes orientaciones 3D [194], para manejar oclusiones [112], y para trabajar con otros tipos de objetos. Cabe destacar que la implementación de Intel OpenCV [35], es genérica, entrenable e incluye cascadas ya entrenadas para encontrar caras de frente, de perfil, y todo el cuerpo de un sujeto.

3.3. Detección de caras mediante integrales proyectivas

Básicamente, la técnica de detección de caras que proponemos en este capítulo se encuentra dentro de los métodos basados en apariencia. La novedad radica en trabajar exclusivamente con integrales proyectivas. De esta forma, la idea subyacente es la aplicación de un proceso de búsqueda exhaustiva multiescala, en el que cada fragmento de las imágenes se somete a un test de clasificación cara/no cara, basado en las proyecciones del modelo facial. Por lo tanto, podemos decir que se trata de un método que se fundamenta en la *apariencia 1,5D del rostro*. Los modelos de proyección utilizados por el detector son obtenidos mediante entrenamiento, como explicamos en la sección 2.2, por lo que el método puede adaptarse de forma más o menos directa a otros tipos de objetos.

En el resto de esta sección vamos a describir detenidamente el mecanismo de detección propuesto. En primer lugar mostramos la estructura global del método, en el apartado 3.3.1, para después profundizar en cada uno de los pasos principales del algoritmo, en los apartados 3.3.2, 3.3.3 y 3.3.4. Por último, el apartado 3.3.5 analiza cómo el test cara/no cara se puede utilizar en conjunción con otros detectores para mejorar los resultados de ambos.

3.3.1. Esquema global del método de detección

En la figura 3.20 se representa esquemáticamente el método propuesto para la detección de caras humanas usando integrales proyectivas, mostrando su aplicación sobre un ejemplo.

El detector recibe como entrada la imagen a analizar (en color o en escala de grises) y un par de modelos de señal: el de la proyección vertical de la cara (MV_{cara}) y el de la proyección horizontal de los ojos (MH_{ojos}). Como se puede observar, el algoritmo consta de tres grandes pasos: (i) se calculan proyecciones verticales de la imagen, por bandas y a distintas resolu-

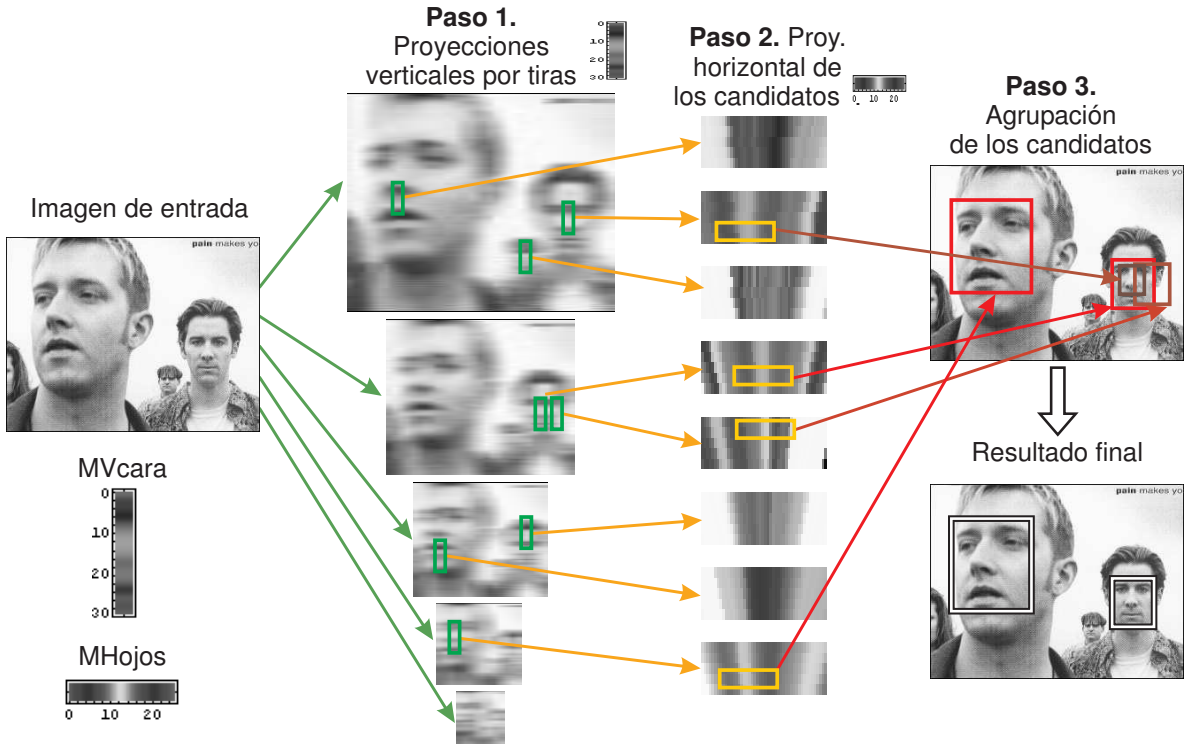


Figura 3.20: *Esquema global del detector de caras mediante integrales proyectivas. La entrada es la imagen a analizar y los modelos de PV_{cara} y PH_{ojos} (a la izquierda). Paso 1. Sobre las proyecciones verticales se buscan apariciones de MV_{cara} (en verde). Paso 2. Sobre las proyecciones horizontales de los candidatos se busca MH_{ojos} (en amarillo). Paso 3. Las posiciones resultantes (en rojo) se agrupan para obtener el resultado final.*

ciones, y se buscan las apariciones del modelo MV_{cara}; (ii) sobre los candidatos obtenidos en el paso anterior, se calculan las proyecciones horizontales y se comprueba la existencia del modelo MH_{ojos}; (iii) finalmente, el resultado de los pasos previos se traslada a la imagen, y se realiza una criba y agrupación de candidatos.

Por lo tanto, el método detectará como cara todas las regiones cuya proyección vertical se asemeje a MV_{cara} y la horizontal (en la parte superior de la región) sea parecida a MH_{ojos}. El tamaño de ambos modelos de proyección impone una limitación sobre el tamaño mínimo de las caras detectadas y, en consecuencia, debería mantenerse reducido.

Es lógico pensar que en una imagen de cierto tamaño, y analizada con alta resolución, podrán existir muchas regiones de no cara que cumplan las dos condiciones anteriores. Esta ambigüedad existe, aunque en los experimentos veremos que es mucho menor que con otros métodos de detección, como usando patrones 2D. Además, el paso (iii) del algoritmo eliminará las no caras solapadas o próximas a caras detectadas con mayor fiabilidad.

3.3.2. Búsqueda de candidatos usando proyecciones verticales

El primer paso del algoritmo de detección consiste en buscar apariciones de MV_{cara} en subimágenes de la entrada. Para ello, hacemos uso de la robustez de las proyecciones, que,

como vimos en la sección 2.3.3, mantienen su forma bajo ligeras variaciones en la posición de las regiones proyectadas. El método básico se repite en una búsqueda multiescala.

Iteración multiescala

En general, el primer reto al que se enfrenta cualquier método de detección de objetos es la indeterminación sobre la posición y el tamaño del objeto buscado dentro de las imágenes. Hemos visto que una de las soluciones más habituales es aplicar una iteración multiescala, tomando como base un método de escala fija. Es decir, un detector básico que admite sólo determinado tamaño se repite escalando la imagen con distintas resoluciones.

Concretamente, se define un factor de reducción f. De esta manera, si el tamaño original de la imagen es de $W \times H$ píxeles, en el siguiente paso será de $W/f \times H/f$, luego $W/f^2 \times H/f^2$, y así sucesivamente; en la iteración i-ésima el tamaño será de $W/f^{i-1} \times H/f^{i-1}$ píxeles. Típicamente, los métodos de detección de caras utilizan factores del orden de $f = 1,2$. En la tabla 3.1 se presenta un ejemplo de las iteraciones a las que daría lugar ese factor sobre una imagen de 320×240 píxeles.

Paso	Imagen	Cara	Paso	Imagen	Cara
1	320×240	24×30	7	107×80	72×90
2	267×200	29×36	8	89×67	86×107
3	222×167	35×43	9	74×56	103×129
4	185×139	41×52	10	62×47	124×155
5	154×116	50×62	11	51×39	149×186
6	129×96	60×75	12	43×32	178×223

Tabla 3.1: *Ejemplo de iteraciones en una búsqueda multiescala usando un factor $f = 1,2$. Se supone que la imagen de entrada es de 320×240 píxeles y el tamaño de las caras detectadas de 24×30 píxeles. En total se ejecutarían 12 iteraciones; para cada una de ellas, se indica el tamaño de la imagen procesada y el de las caras en la imagen original.*

Un factor de reducción menor podría mejorar un poco la capacidad de detección, aunque haría que el coste computacional del método aumentara muy rápidamente.

Proyecciones verticales por tiras

Supongamos una iteración concreta de una búsqueda multiescala genérica. Si la imagen actual es de $W \times H$ píxeles y el modelo de cara es de tamaño[5] $w \times h$, se deben analizar en total $(W - w + 1)(H - h + 1)$ posibles posiciones de cara.

Pero las posiciones adyacentes presentan una gran correlación, por lo que algunas comprobaciones se podrían evitar. Esto es más evidente si utilizamos integrales proyectivas: la proyección vertical de una región es prácticamente idéntica a la proyección de esa región desplazada un píxel a la izquierda o a la derecha. Por poner un ejemplo, si $w = 24$, el 92 % de los píxeles proyectados (22 de 24) son los mismos.

[5] En el modelo de caras mediante proyecciones, h sería el tamaño de MV_{cara}, y w el de MH_{ojos}.

En consecuencia, proponemos una *reducción del espacio de búsqueda* en sentido horizontal: la proyección vertical se calcula en "tiras" o "bandas" de w píxeles de ancho y separadas g píxeles. Cada "tira" es una región del ancho dado, que ocupa todo el alto de la imagen. De una tira a la siguiente se desplaza cierta cantidad de píxeles, que hemos denotado por g. En la figura 3.21 se ilustra el proceso de cálculo de integrales proyectivas verticales por tiras.

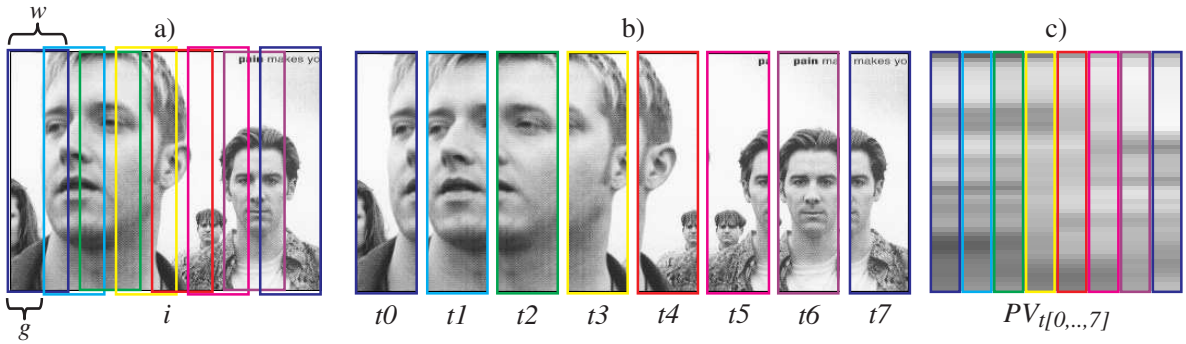


Figura 3.21: *Ejemplo de cálculo de proyecciones verticales por tiras. a) Imagen de entrada, i, de 287×220 píxeles, procesada con un factor de escala 4,2 (es decir, tamaño virtual de 68×52). b) Tiras extraídas de i, con $w = 24$ y $g = 12$. c) Proyecciones verticales de las tiras de b).*

Normalmente el salto, g, se define en función del ancho de las tiras. Un valor típico usado en los experimentos es la proporción $g = w/3$, es decir, un salto horizontal de un tercio del ancho de cara.

Cuando las imágenes de entrada son en color, existen distintas posibilidades: proyectar alguno de los canales o trabajar con el valor de intensidad. Como veremos en los experimentos, en el caso específico de las caras lo más interesante es proyectar el canal rojo, que suele ofrecer un mayor contraste para los tonos del rostro.

Implementación del cálculo de proyecciones por tiras

Con toda seguridad, este primer paso del detector será el de mayor impacto en la complejidad computacional del algoritmo. Su ejecución requiere cálculos simples pero repetidos muchas veces: reducir las imágenes, extraer las tiras, sumar filas para calcular las proyecciones, y almacenar los resultados. Pero cuidando la implementación, podemos conseguir una reducción muy significativa de los tiempos de ejecución.

En particular, resulta especialmente interesante el concepto de *imagen integral* [188], que permite simplificar tanto el cálculo de las proyecciones verticales por tiras como la búsqueda multiescala. Veamos en primer lugar su definición.

Definición 3.2 ***Imagen integral.***

Dada una imagen i de tamaño $W \times H$, la imagen integral asociada a i es otra imagen, suma, de tamaño $W+1 \times H+1$, donde:

$$suma(x,y) := \sum_{\forall x' < x} \sum_{\forall y' < y} i(x',y')$$

Esto es, cada píxel acumula los píxeles que tiene encima y a su izquierda. En la figura 3.22b) se puede ver un ejemplo típico de una imagen integral. Obviamente, el rango de valores de salida es mucho mayor que el de entrada, por lo que la imagen integral aparece saturada.

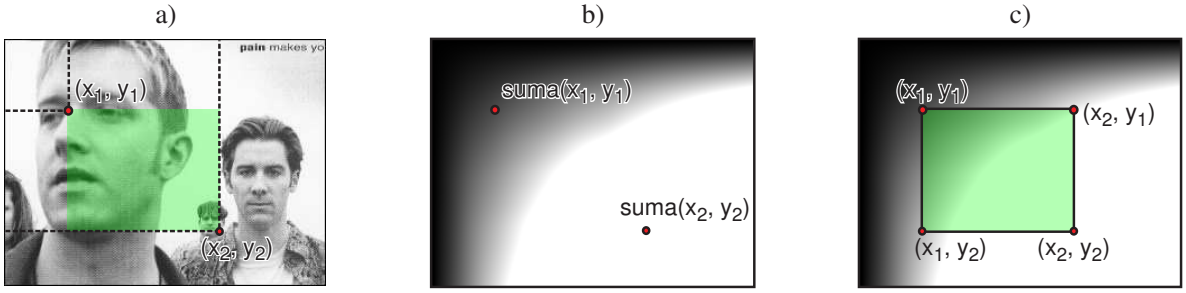


Figura 3.22: *Ejemplo de imagen integral y cálculo de una suma de píxeles. a) Imagen de entrada, i, sobre la que se quiere calcular la suma de píxeles en el rectángulo de esquinas (x_1, y_1), (x_2, y_2). b) Imagen integral de i; cada valor $suma(x_k, y_k)$ es la suma de los píxeles de i que están arriba y a la izquierda. c) Obtención de la suma del rectángulo, usando la imagen integral.*

Usando las imágenes integrales, es posible calcular la suma de los píxeles dentro de un rectángulo de i con esquinas (x_1, y_1), (x_2, y_2), con sólo tres operaciones de suma/resta:

$$\sum_{x_1 \leq x \leq x_2} \sum_{y_1 \leq y \leq y_2} i(x, y) = suma(x_2, y_2) - suma(x_1, y_2) - suma(x_2, y_1) + suma(x_1, y_1) \quad (3.3)$$

De esta forma, una vez obtenida la imagen integral, el cálculo de cada punto de las proyecciones se reduce a esas tres sumas/restas, en lugar de las w originales. Es más, el proceso de reducción multiescala se puede hacer implícitamente, sin más que "escalar" convenientemente los puntos en los que se aplica la fórmula 3.3. En definitiva, en el algoritmo 3.1 se concreta el proceso de cálculo de las proyecciones en este primer paso del detector.

Claramente, el orden de complejidad del algoritmo 3.1 es $O(H(W/f_a - w)/f_a g)$. Para la primera iteración, tenemos que $f_a = 1$. Además, podemos despreciar w frente a W, por lo que tenemos un orden de $O(WH/g)$. Es decir, el número de operaciones es g veces menor que el número de píxeles de la imagen, WH. Es más, si consideramos todas las iteraciones con las diferentes escalas posibles, nos queda un tiempo de:

$$t(W, H, g, f) = \frac{WH}{g} + \frac{WH}{gf^2} + \frac{WH}{gf^4} + \cdots = \frac{WH}{g} \sum_{i=0}^{k} \frac{1}{f^{2i}} \leq \frac{WH}{g} \frac{f^2}{f^2 - 1} \quad (3.4)$$

El factor $f^2/(f^2 - 1)$ aumenta a medida que f tiende a 1; para un ajuste típico de $f = 1,2$ vale 3,27. De esta forma, incluso considerando todas las escalas, si fijamos f el número de operaciones seguiría siendo un $O(WH/g)$. Este resultado es muy interesante, ya que vemos que el orden se encuentra por debajo del número de píxeles de la imagen. Ahora bien, como el tiempo de obtener la imagen integral es un $O(WH)$, el orden total vendría dado por esa cota; en cualquier caso, lineal respecto del número de píxeles.

PROYECCIONES VERTICALES POR TIRAS
ENTRADA:
- Imagen integral: $suma$, de tamaño $W+1 \times H+1$.
- Ancho de las tiras: w.
- Salto entre tiras: g.
- Factor de reducción actual de la imagen: f_a.

SALIDA:
- PV_t: array $[0,\ldots,(W/f_a - w)/g]$ de proyecciones de tamaño H/f_a.

ALGORITMO:
Iteración principal:
para $k := 0$ ***hasta*** $(W/f_a - w)/g$ ***hacer***
para $j := 0$ ***hasta*** $H/f_a - 1$ ***hacer***
$x_1 := kgf_a;\ x_2 := x_1 + wf_a$
$y_1 := jf_a;\ y_2 := (j+1)f_a$
$PV_{t[k]}(j) := suma(x_2, y_2) - suma(x_1, y_2) - suma(x_2, y_1) + suma(x_1, y_1)$
finpara
finpara

Algoritmo 3.1: *Cálculo de las proyecciones verticales por tiras de una imagen. Observar que se parte de la imagen integral, suma, de la imagen original, i, puesto que sólo se calcula una vez para todo el proceso multiescala. f_a indica el factor de reducción en cada momento: 1, f, f^2, f^3, etc.*

Búsqueda de caras candidatas

Una vez obtenidas las proyecciones verticales por tiras, se buscan en ellas las apariciones del modelo MV_{cara}. Se trata de una búsqueda de tamaño fijo (recordemos que el problema de la escala se resuelve en la iteración multiescala). Por lo tanto, podemos aplicar la distancia señal/modelo para obtener una medida de verosimilitud de la aparición del modelo en cada una de las posiciones posibles de las tiras.

Suponiendo un modelo de proyección media/varianza, $MV_{cara} = (M, V)$, podemos usar la fórmula de *dist* de la ecuación 2.23 (ver la página 56). No obstante, para que esta medida funcione correctamente, el trozo de la tira debe estar normalizado en el valor respecto del modelo. La señal normalizada según la ecuación 2.4 (ver la página 39) quedará:

$$PV'_{t[xx,yy]} := normal_{medvar}\left(PV_{t[xx]}(yy), PV_{t[xx]}(yy+1), ..., PV_{t[xx]}(yy+h-1)\right) \tag{3.5}$$

donde xx indica la tira (entre 0 y $(W-w+1)/g$); yy indica el tope superior del trozo de región analizado (entre 0 y $H-h+1$); med es la media de M; y var la varianza de M. De esta manera, en $PV'_{t[xx,yy]}$ tenemos un fragmento de las tiras normalizado en la intensidad, desde $PV_{t[xx]}(yy)$ hasta $PV_{t[xx]}(yy+h-1)$. Ahora podemos aplicar la ecuación 2.23 a las distintas posiciones de las tiras, PV_t, obteniendo:

$$disTiras\left((M,V), PV_t, xx, yy\right) := \frac{1}{||h||} \sum_{i=0,\ldots,h-1} \frac{\left(PV'_{t[xx,yy]}(yy+i) - M(i)\right)^2}{V(i)} \tag{3.6}$$

Se puede ver un ejemplo de aplicación de esta medida en la figura 3.23, para las tiras de la figura 3.21. En la figura 3.23d) se muestra una interpretación de las regiones de las imágenes correspondientes a algunos de los trozos de tira.

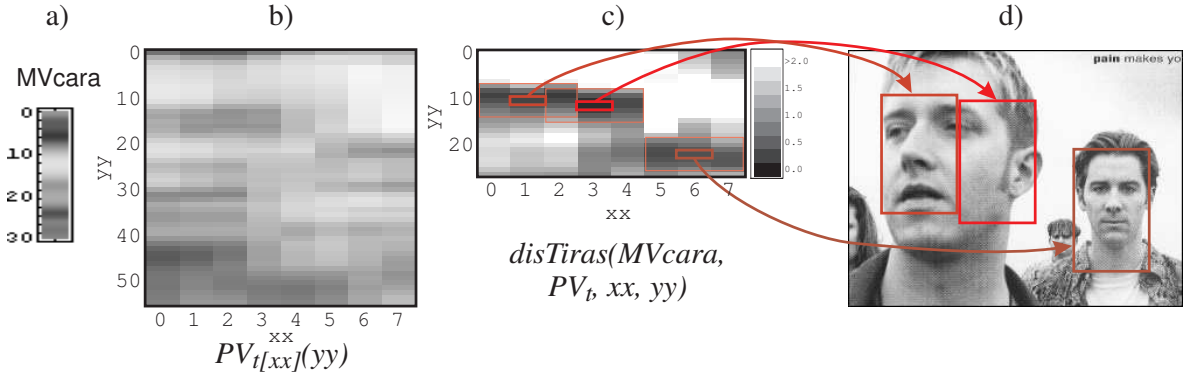


Figura 3.23: *Búsqueda de candidatos de cara en las tiras verticales. a) Modelo de cara utilizado (se muestra sólo la media) de tamaño $h = 30$, $w = 24$. b) Proyecciones verticales de las tiras (ver la figura 3.21). c) Distancias de los trozos de tira al modelo (ver ecuación 3.6). Se han señalado los tres mínimos locales por debajo de distancia 1. El tamaño de la vecindad local viene dado por $vx = 1$, $vy = 3$. d) Los mínimos locales, trasladados al espacio de la imagen original.*

Las caras candidatas son localizadas en los mínimos locales de la ecuación 3.6, para todo xx, yy, y para cada factor de escala admitido. La búsqueda de estos mínimos locales se realiza mediante un simple proceso iterativo, en el que se encuentra el mínimo global, se almacena si está por debajo de un umbral, y se anula una vecindad del mínimo antes de repetir el proceso. En el algoritmo 3.2 se detalla este proceso de selección de los candidatos.

Existe una heurística sencilla relacionada con la ecuación 3.5, donde se normalizan los trozos de tiras procesados. Si se encuentra que la varianza de un trozo de proyección cae por debajo de cierto umbral, es posible descartarlo, ya que corresponderá a una región uniforme de la imagen, o con alta uniformidad. Esta heurística no sólo es razonable sino que es muy conveniente aplicarla, puesto que algunas regiones más o menos uniformes podrían ajustarse aleatoriamente al modelo de cara al ser normalizadas. De forma parecida, podemos fijar también un umbral superior para la varianza (región de excesivo contraste), y para la media un umbral inferior (región muy oscura) y otro superior (región muy clara). No obstante, conviene ser flexibles en el ajuste de estos umbrales heurísticos.

En la figura 3.24 se muestran algunos ejemplos de candidatos resultantes del primer paso del algoritmo de detección, para varias imágenes usadas en los experimentos. Es sorprendente cómo, en algunos casos, basta con aplicar únicamente este paso del algoritmo para conseguir detectar perfectamente las caras existentes en las imágenes. No obstante, esa no será la situación más frecuente, y normalmente habrá que comprobar y eliminar los muchos candidatos espurios que suelen aparecer.

A pesar de ello, podemos concluir que el paso de búsqueda de candidatos mediante proyecciones verticales es bastante efectivo, considerando la enorme cantidad de posibles subregiones examinadas. Por poner un ejemplo, en la imagen “1091.jpg” de la figura 3.24, de

BÚSQUEDA DE CARAS CANDIDATAS
ENTRADA:
- Ancho y alto del modelo: w, h.
- Salto entre tiras: g.
- Factor de reducción actual de la imagen: f_a.
- Distancias de las tiras al modelo: *dTiras*: matriz de $[0,\ldots,(W/f_a - w + 1)/g] \times [0,\ldots,H/f_a - h]$, siendo cada: $dTiras[xx, yy] = disTiras((M,V), PV_t, xx, yy)$, según la ecuación 3.6.
- Umbral máximo de distancia: $maxDistPV$.
- Tamaño de la vecindad local en X e Y: vx, vy.

SALIDA:
- Lista de caras candidatas: *candCara*: lista de rectángulos.

ALGORITMO:
Inicialización:
candCara:= lista vacía
$(xx^*, yy^*) := \arg\min_{\forall x,y} dTiras[x,y]$ /* *Buscar mínimo global* */

Iteración principal:
mientras $dTiras[xx^*, yy^*] \leq maxDistPV$ ***hacer*** /* *Repetir mientras no supere el umbral* */
r:= rectángulo $[(gf_a xx^*, f_a yy^*), (gf_a xx^* + wf_a, f_a yy^* + hf_a)]$
Insertar *r* en la lista *candCara*
para $x := -vx$ ***hasta*** vx ***hacer*** /* *Eliminar vecinos del mínimo* */
para $y := -vy$ ***hasta*** vy ***hacer***
$dTiras[xx, yy]$:= INFINITO
finpara
finpara
$(xx^*, yy^*) := \arg\min_{\forall x,y} dTiras[x,y]$ /* *Buscar siguiente mínimo* */
finmientras

Algoritmo 3.2: *Búsqueda de caras candidatas en los mínimos locales de disTiras. Nótese que los parámetros w, h, g y f_a intervienen únicamente para generar las posiciones de los candidatos en la imagen original, asociados a los mínimos encontrados.*

395×278 píxeles, aparecen 20 falsos candidatos; sin embargo, el número total de localizaciones analizadas, para todas las escalas, asciende a 44.764. Incluso para este ejemplo aparentemente desfavorable, se ha rechazado el 99,96 % de las regiones de no cara y, lo que es más importante, sin descartar las caras existentes.

3.3.3. Verificación de candidatos con proyección horizontal

Después del primer paso, tenemos un conjunto de regiones cuya proyección vertical coincide con MV_{cara}. Algunas de ellas son caras, otras serán regiones "cercanas a caras" –como se puede observar en la figura 3.24–, pero otras muchas serán no caras con proyección vertical parecida a MV_{cara}. El segundo paso del detector consiste en verificar la validez de los candidatos usando MH_{ojos}. Además, este paso tiene la función de realizar un ajuste fino de la escala y la posición horizontal de las caras, de manera que las regiones cercanas a caras se sitúen sobre los rostros existentes.

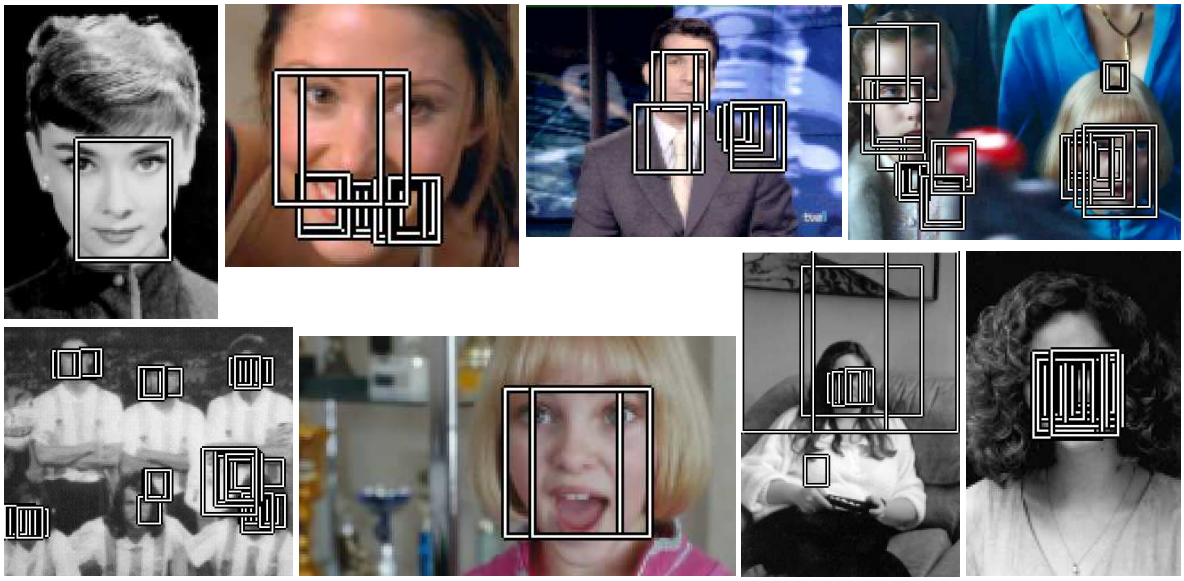


Figura 3.24: *Caras candidatas tras el primer paso del detector. De arriba abajo, de izquierda a derecha: audrey2.gif (CMU/MIT), 033.jpg (UMU), 002.jpg (UMU), 1091.jpg (UMU), fragmento de Argentina.gif (CMU/MIT), 5013.avi.jpg (UMU), tammy.gif (CMU/MIT), kaari1.gif (CMU/MIT). Se han usado distintos umbrales de $maxDistPV$, en algunos casos ajustados para seleccionar sólo el mejor candidato.*

Proyecciones horizontales de los candidatos

Si analizamos detenidamente los resultados de la figura 3.24, podemos deducir que el primer paso del algoritmo produce una alta indeterminación en sentido horizontal: aparecen muchos candidatos a derecha e izquierda de las caras existentes. Estos falsos candidatos a ambos lados de los rostros pueden, incluso, producir menor distancia para el criterio de la ecuación 3.6 que las posiciones reales de cara. Podemos sacar varias conclusiones:

- La incertidumbre horizontal se debe reducir aplicando proyecciones horizontales, usando para ello el modelo MH_{ojos}.
- Se debe proyectar un trozo mayor que el ancho de la región (w), añadiendo un margen que incluya el desplazamiento (a derecha y a izquierda) que puede ocurrir típicamente en el primer paso del algoritmo.
- Se debe refinar el factor de escala, para permitir un rango de tamaños mayor que el que ofrece el factor de escala (f).

En cierto sentido, podemos decir que la búsqueda de MH_{ojos} en las proyecciones horizontales será parecida a la de MV_{cara} en las verticales, pero con una sola tira, de tamaño más reducido y en una iteración multiescala limitada a los rangos del factor de incremento, esto es, desde escala f hasta $1/f$. Así, tendremos un array de proyecciones, $PH_{c[j]}$, igual que antes, aunque ahora el índice j del array se refiere a distintas escalas, y no a las tiras. El tamaño del array indicará la resolución usada para las escalas.

Debemos recordar, por otro lado, que la proyección PH_{ojos} se restringe a una parte de la región de cara, limitada verticalmente por los parámetros $y_{ojosmin}$ e $y_{ojosmax}$ del modelo, en proporción a la altura de la región (ver la definición del modelo de cara en la página 61). En la figura 3.25 se muestra un ejemplo de la región proyectada para una cara candidata.

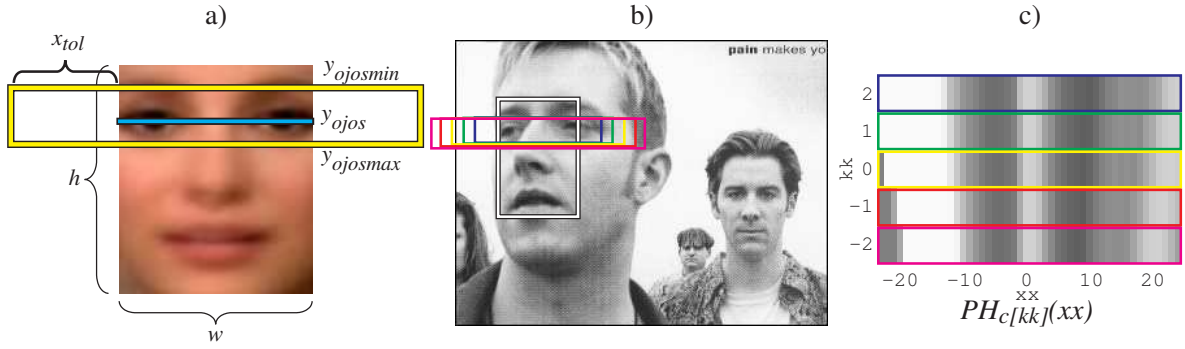


Figura 3.25: *Ejemplo de cálculo de proyecciones horizontales de caras candidatas. a) Parámetros del modelo de cara que definen la región proyectada del candidato. Se añade un ancho adicional, dado por x_{tol}. b) Cara candidata analizada (en recuadro blanco) y las regiones proyectadas para cada resolución definida. c) Array de proyecciones horizontales del candidato, PH_c. En este caso, $k = 2$ y $f = 1, 2$.*

Aparte de los valores definidos en el modelo de cara, aparecen dos parámetros adicionales en la creación de las proyecciones horizontales: el tamaño adicional tomado a lo ancho, x_{tol}; y el número de saltos en la resolución, k (que determina el tamaño del array PH_c). De esta manera, la posición $PH_{c[0]}$ contiene la proyección horizontal de la región esperada de ojos con escala 1 (la misma escala del candidato); $PH_{c[1]}$ con escala $f^{1/k}$; $PH_{c[2]}$ con $f^{2/k}$; y así hasta $PH_{c[k]}$ que será f. Igualmente, para los índices negativos $PH_{c[-v]}$ es $f^{-v/k}$. Debemos aclarar que x_{tol} es el ancho adicional para $PH_{c[0]}$; lógicamente, al cambiar de escala se tomará más o menos ancho adicional, como se puede comprobar en la figura 3.25.

En definitiva, en el algoritmo 3.3 se describe el proceso de obtención de las proyecciones horizontales asociadas a cada candidato, $PH_{c[-k..k]}$. Hemos considerado que conocemos la escala actual del candidato, f_a, aunque por simplicidad no se almacena explícitamente en el algoritmo 3.2. La figura 3.25c) contiene un ejemplo del resultado de este proceso.

Localización y verificación del candidato

El proceso de búsqueda de MH_{ojos} en las proyecciones horizontales de cada candidato es análogo a la búsqueda de MV_{cara} en el conjunto de proyecciones PV_t. Así que podemos resolverlo aplicando el mismo proceso: suponemos una normalización, PH'_c, de los trozos de PH_c equivalente a la ecuación 3.5; definimos una medida de distancia para cada posición del patrón buscado como en la ecuación 3.6; y realizamos una búsqueda del mínimo como en el algoritmo 3.2. Sea la medida de distancia:

$$disCand\left((M,V),PH_c,kk,xx\right) := \frac{1}{||w||}\sum_{i=0,\ldots,w-1}\frac{\left(PH'_{c[kk]}(xx+i)-M(i)\right)^2}{V(i)} \tag{3.7}$$

PROYECCIONES HORIZONTALES DE CARAS CANDIDATAS
ENTRADA:
- Imagen integral: $suma$, de tamaño $W+1 \times H+1$.
- Esquina superior izquierda del candidato: xx, yy.
- Factor de escala actual del candidato: f_a.
- Ancho y alto del modelo de cara: w, h.
- Parámetros del modelo de cara: $y_{ojosmin}$, $y_{ojosmax}$.
- Ancho adicional de la región proyectada: x_{tol}.
- Factor de incremento de escala y número de pasos: f, k.

SALIDA:
- PH_c: array $[-k, \ldots, k]$ de proyecciones con dominio desde $-x_{tol}$ hasta $w + x_{tol}$.

ALGORITMO:
Iteración principal:
$y_1 := yy + y_{ojosmin} f_a$
$y_2 := yy + y_{ojosmax} f_a$
para $v := -k$ ***hasta*** k ***hacer***
$f' := f_a f^{v/k}$
para $j := -x_{tol}$ ***hasta*** $w + x_{tol}$ ***hacer***
$x_1 := xx + f'j$
$x_2 := xx + f'(j+1)$
$PH_{c[v]}(j) := suma(x_2, y_2) - suma(x_1, y_2) - suma(x_2, y_1) + suma(x_1, y_1)$
finpara
finpara

Algoritmo 3.3: *Cálculo de las proyecciones horizontales de las caras candidatas. El algoritmo trabaja con la imagen integral, suma, de la imagen original, i. El proceso se repetiría para cada candidato.*

donde ahora (M, V) es el modelo media/varianza de PH_{ojos}; kk son los factores de escala considerados, desde $-k$ hasta k; y xx es el desplazamiento en el eje X del trozo de proyección horizontal.

Puesto que se trata de verificar un candidato dado, realmente sólo es necesario localizar el mínimo global de la función *disCand* y comprobar si está por debajo de cierto umbral, que denotaremos por $maxDistPH$:

$$\{kk^*, xx^*\} = \arg \min_{\forall kk, xx} disCand\left((M, V), PH_c, kk, xx\right) \tag{3.8}$$

El umbral $maxDistPH$ desempeña un papel importante, ya que es el que marca el límite de la decisión de aceptación/rechazo del candidato. Si la distancia mínima de la ecuación 3.8 es mayor que $maxDistPH$, la verificación del candidato ha resultado negativa. En otro caso, se reajusta la escala y la posición en X del candidato dado. El desplazamiento en X sería de $xx^* \cdot f^{kk^*/k}$ píxeles, mientras que la escala –es decir, el ancho y el alto de la región de cara–, debe ajustarse también con el factor $f^{kk^*/k}$.

En la figura 3.26 se muestran sendos ejemplos del proceso de verificación de candidatos mediante proyecciones horizontales. En el ejemplo superior, el mínimo de *disCand* está por debajo del umbral, por lo que el candidato es aceptado y se reajusta la posición del rectángulo contenedor. En el inferior, la distancia está por encima del umbral, de manera que se rechaza

el candidato y no pasa a la siguiente fase del detector.

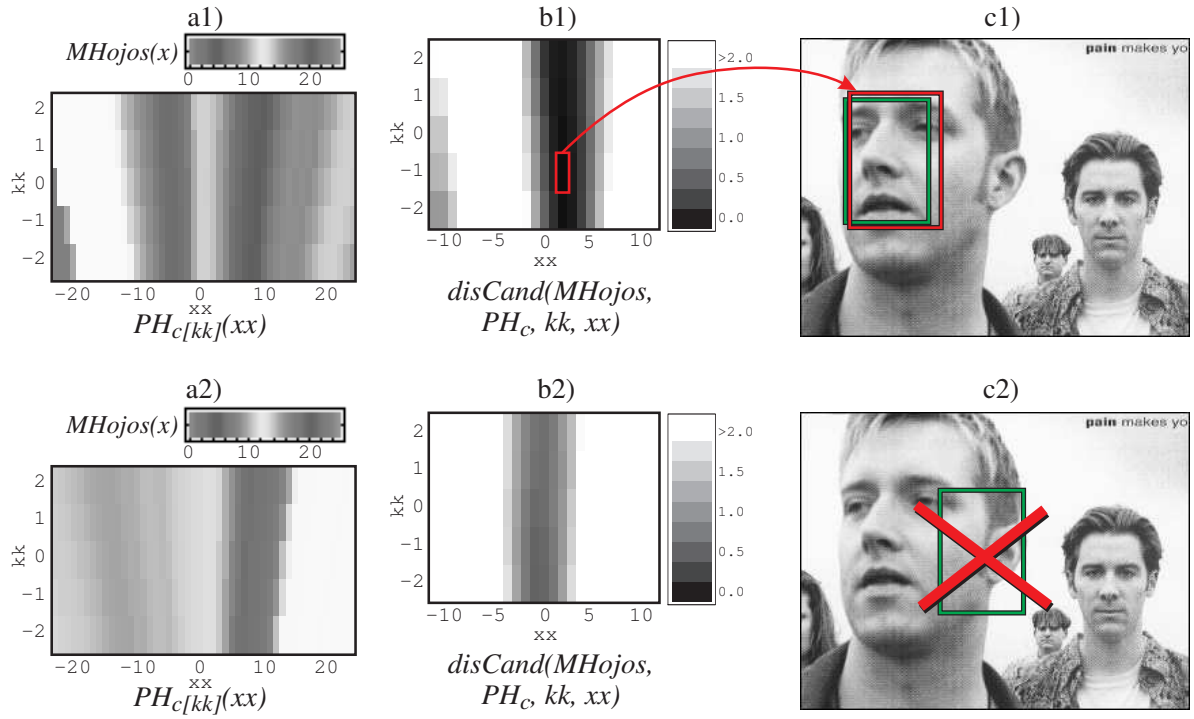


Figura 3.26: *Verificación de caras candidatas mediante proyecciones horizontales. a1,2) Modelo de proyección horizontal de los ojos, MH_{ojos}, y proyecciones horizontales de las regiones candidatas. b1,2) Distancias de los trozos de las proyecciones horizontales al modelo. c1) Como el mínimo está por debajo del umbral, el candidato (en verde) se acepta y se reajusta (en rojo) según la posición del mínimo. c2) El mínimo supera el umbral, por lo que el candidato (en verde) se rechaza.*

Se pueden ver algunos resultados del proceso de verificación de candidatos en la figura 3.27. Los candidatos entrantes son los mismos de la figura 3.24. Es interesante apreciar las dos contribuciones de este paso al proceso de detección: por un lado, eliminar muchos de los falsos positivos del primer paso; por otro lado, relocalizar la posición horizontal de los candidatos verdaderos.

La verificación de candidatos mediante la proyección horizontal, PH_{ojos}, demuestra ser bastante efectiva, como se puede ver comparando las imágenes de las figuras 3.24 y 3.27. No obstante, por sí sola, su efectividad es menor que la que ofrece el modelo MV_{cara}, como explicamos en el apartado 2.2.5 del anterior capítulo (y cuantificamos en la tabla 2.2). Esa es la justificación de que en el algoritmo la búsqueda de MV_{cara} vaya en primer lugar, y la de MH_{ojos} vaya a continuación, y no al contrario.

3.3.4. Agrupación y selección de candidatos

La última etapa del algoritmo de detección de caras es común a la mayoría de las técnicas existentes, especialmente a los métodos basados en apariencia. Después de haber seleccionado un conjunto de candidatos, procesados hasta este punto de forma independiente, se debe tomar la decisión final sobre las caras resultantes. Son principalmente tres las cuestiones a re-

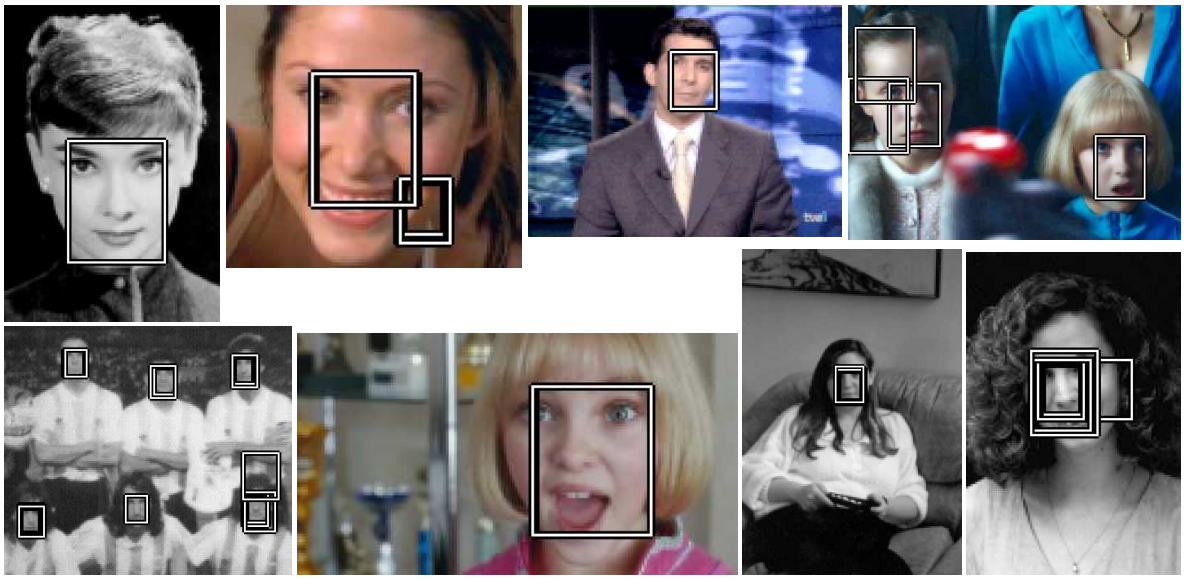


Figura 3.27: *Candidatos verificados tras el segundo paso del detector. De arriba abajo, de izquierda a derecha: audrey2.gif (CMU/MIT), 033.jpg (UMU), 002.jpg (UMU), 1091.jpg (UMU), fragmento de Argentina.gif (CMU/MIT), 5013.avi.jpg (UMU), tammy.gif (CMU/MIT), kaari1.gif (CMU/MIT).*

solver: (1) agrupar caras candidatas con posiciones parecidas; (2) buscar candidatos solapados y decidir cuál dejar y cuál quitar; y (3) aplicar heurísticas adicionales, en caso necesario.

Vamos a abordar cada uno de los puntos, concluyendo así la definición del algoritmo de detección de caras. En adelante supondremos que tenemos una lista de n caras candidatas, numeradas de 1 a n. De cada cara candidata tenemos su rectángulo contenedor, r_i, la distancia mínima para la proyección vertical, dpv_i (dada por el valor de *disTiras*, de la ecuación 3.23), y para la horizontal dph_i (según *disCand* de la fórmula 3.8). Los dos últimos valores nos darán una medida de prioridad o preferencia entre dos candidatos distintos.

Agrupación de regiones candidatas

En la figura 3.27 se pueden ver algunos ejemplos típicos de las salidas que suele producir el segundo paso del detector de caras. Las posibles situaciones se resumen de forma esquemática en la figura 3.28.

El caso más favorable es el de la figura 3.28a); un mismo rostro es detectado mediante dos o más regiones candidatas situadas en posiciones muy próximas. El hecho de que existan muchos candidatos en posiciones comunes no es negativo, sino más bien todo lo contrario: es una confirmación de que la cara ha sido detectada con alta fiabilidad. De hecho, en muchos métodos basados en apariencia se utiliza el número de candidatos solapados como un criterio para seleccionar o descartar la región correspondiente.

Supongamos que las esquinas opuestas de un candidato están dadas por (x_i^a, y_i^a) y (x_i^b, y_i^b), para todo i desde 1 hasta n, como se muestra en la figura 3.28d). El *porcentaje de solapamiento*

Figura 3.28: *Clasificación de situaciones de solapamiento de candidatos. a) Candidatos con solapamiento muy grande. b) Un candidato está contenido dentro de otro. c) Candidatos solapados ligeramente, pero incompatibles entre sí (sólo uno puede corresponder a una cara). d) Parámetros que definen las regiones de los candidatos i y j, para el cálculo del solapamiento.*

del candidato i respecto del j (que denominaremos $solap$) viene dado por:

$$solap(i,j) := \frac{solap_{base}(i,j) \cdot solap_{altura}(i,j)}{(x_i^b - x_i^a)(y_i^b - y_i^a)} \tag{3.9}$$

con:

$$solap_{base}(i,j) := \text{máx}\left(0, min(x_i^b, x_j^b) - max(x_i^a, x_j^a)\right)$$

$$solap_{altura}(i,j) := \text{máx}\left(0, min(y_i^b, y_j^b) - max(y_i^a, y_j^a)\right)$$

La proximidad entre dos candidatos se puede medir en función del solapamiento; la única salvedad es que $solap(i,j)$ será distinto de $solap(j,i)$ si i y j son de diferente tamaño. Se puede ver un ejemplo claro de esa diferencia en la figura 3.28b), donde un candidato pequeño está contenido dentro de otro mayor. El solapamiento desde el punto de vista del pequeño será del 100 %, pero para el grande no llega al 15 %. En consecuencia, la función de proximidad (que denotamos por $proxim$), se debería basar en el menor de ambos ratios:

$$proxim(i,j) := \text{mín}\{solap(i,j), solap(j,i)\} \tag{3.10}$$

El funcionamiento del proceso de agrupación de candidatos es sencillo. Para todas las parejas de candidatos, i, j, comprueba si la medida $proxim(i,j)$ está por encima de cierto umbral[6]. En caso afirmativo, existen dos posibilidades: combinar los rectángulos correspondientes (obteniendo un nuevo rectángulo en posiciones intermedias), o seleccionar el candidato detectado con más fiabilidad (esto es, con menor distancia a los modelos).

En la práctica, la segunda forma suele funcionar ligeramente mejor. Una posible definición

[6]Por ejemplo, el usado normalmente en las pruebas es del 65 %.

de la *fiabilidad de un candidato*, i, a partir de las distancias, pdv_i y dph_i, podría ser:

$$fiabilidad(i) := \frac{1}{1 + (0,5dpv_i + 0,5dph_i)^2} \tag{3.11}$$

No obstante, en el siguiente capítulo veremos que estas pequeñas diferencias de posición son irrelevantes tras ejecutar el paso de localización de componentes faciales.

Eliminación de regiones solapadas

La justificación para eliminar regiones candidatas que se solapan con otras es fundamentalmente heurística: en una imagen natural las caras humanas muy raramente aparecen solapadas, sino que ocupan posiciones diferenciadas. Pueden ocurrir dos situaciones de solapamiento, aunque ambas se tratan de la misma forma: un candidato que aparece dentro de otro más grande; o bien, candidatos de tamaño similar pero adyacentes entre sí.

La mayoría de las veces, las posiciones correctas de cara producen menores distancias al modelo; de esta manera, se puede usar el criterio de distancia para eliminar falsos candidatos que aparecen junto a otras caras detectadas con mayor fiabilidad. Obviamente, la técnica no es perfecta, por lo que en algún caso se pueden eliminar caras correctas en favor de falsos candidatos.

La comprobación y eliminación de regiones solapadas es posterior a la agrupación de candidatos, y tiene también la forma de un simple proceso iterativo. Para cada grupo resultante, i, se busca el máximo valor de $solap(i, j)$, según la ecuación 3.9, para todos los demás grupos j. Si ese solapamiento máximo es mayor que cierto umbral, se elimina el candidato que tenga menor fiabilidad (calculada con la ecuación 3.11). Normalmente el umbral utilizado es 0, es decir, no se permite nada de solapamiento.

Ajuste de umbrales y heurísticas adicionales

El proceso de detección descrito hasta ahora funciona bien usando umbrales adecuados para las distancias señal/modelo –esto es, $maxDistPV$ y $maxDistPH$–, que son los parámetros del algoritmo con mayor influencia en el resultado final. Es posible encontrar un ajuste fijo que funciona más o menos bien para la mayoría de las imágenes. Sin embargo, en algunos casos, el umbral que detecta correctamente las caras de una imagen, en otra imagen produce un elevado número de falsas alarmas. En otros casos, no se detectan las caras existentes, pero subiendo el umbral se encuentran perfectamente.

Hemos podido comprobar en los experimentos que este problema es común a la mayoría de las técnicas de detección de caras existentes. Un detector será mejor cuanto más "universal" sea el ajuste de los parámetros, es decir, cuanto mejor funcionen unos valores por defecto. En última instancia, la cuestión está estrechamente relacionada con la forma de la curva ROC, como veremos en la siguiente sección.

Alternativamente, se pueden definir algunas heurísticas equivalentes a ajustar los umbrales del detector a posteriori. Por ejemplo, si los requisitos de la aplicación imponen que sólo aparece una cara en las imágenes (como en una aplicación de videoconferencia), se pueden reducir los umbrales hasta que sólo se detecte un rostro. En nuestro caso, esto es equivalente a seleccionar el candidato con mayor valor de fiabilidad.

Si la anterior suposición no es factible, se pueden aplicar otras heurísticas, algunas de las cuales funcionarán mejor que otras. Por ejemplo, se puede fijar un umbral mínimo de verosimilitud a posteriori, en función de la verosimilitud del máximo candidato. Por otro lado, normalmente las caras que aparecen en una imagen suelen tener tamaños parecidos. De esta forma, uniendo ambas heurísticas, podemos eliminar caras que sean significativamente más pequeñas y mucho menos fiables que otras candidatas encontradas en las imágenes. Estos criterios han sido aplicados en los experimentos, fijando el tamaño y la verosimilitud mínima al 60 % y 70 %, respectivamente, en relación al candidato más fiable.

En cualquier caso, se deben poder anular los anteriores criterios heurísticos, que en ciertos casos podrían ser contraproducentes. En la figura 3.29 se muestran algunos resultados finales del proceso de detección de caras desarrollado a lo largo de esta sección.

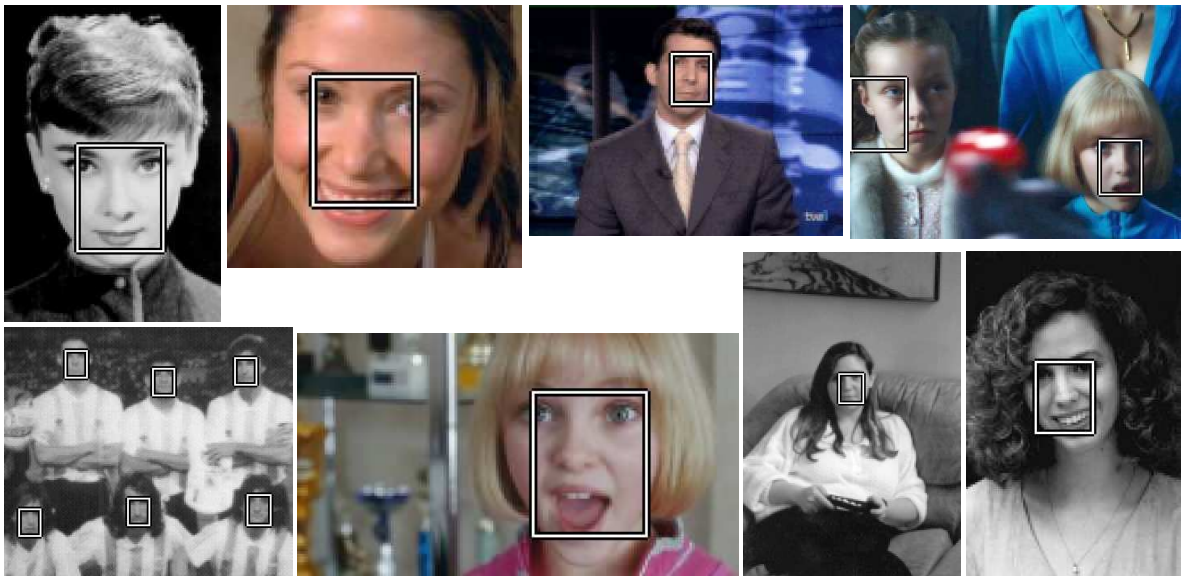


Figura 3.29: *Caras resultantes tras los tres pasos del detector. De arriba abajo, de izquierda a derecha: audrey2.gif (CMU/MIT), 033.jpg (UMU), 002.jpg (UMU), 1091.jpg (UMU), fragmento de Argentina.gif (CMU/MIT), 5013.avi.jpg (UMU), tammy.gif (CMU/MIT), kaari1.gif (CMU/MIT).*

Comparando los resultados de la figura 3.29 con los de la figura 3.27, podemos observar que los procesos de agrupación de candidatos y eliminación de solapados han funcionado en general bastante bien. Sólo en la imagen "1091.jpg" se ha seleccionado un candidato solapado incorrecto. Por otro lado, en ninguno de los ejemplos ha sido necesario aplicar las heurísticas adicionales. Presentaremos más ejemplos del detector y haremos una discusión en profundidad de los resultados de la técnica dentro de la sección 3.4.

3.3.5. Combinación de detectores

Hasta ahora hemos abordado el problema de detección de caras utilizando exclusivamente integrales proyectivas. En los experimentos veremos que el método ofrece buenos resultados, especialmente interesantes en aplicaciones que requieran una buena relación entre tiempo de ejecución y ratio de detección.

No obstante, también es posible combinar las integrales proyectivas con otras técnicas de detección de caras, pudiendo lograr, con un diseño adecuado, una mejora significativa en la efectividad de ambas técnicas por separado. Para ello, en principio, cualquiera de los detectores de caras existentes se puede utilizar como método alternativo. A continuación vamos a proponer dos esquemas de combinación de métodos, que serán utilizados en los experimentos de la sección 3.4.

Esquema genérico de combinación de detectores

Prácticamente todos los detectores admiten diversos modos de funcionamiento dentro de su curva ROC, desde el modo más "permisivo" (con pocas caras perdidas pero con muchas falsas alarmas), hasta el más "restrictivo" (que reduce las falsas detecciones, a costa de disminuir el ratio de detección). Pero es interesante observar que distintos detectores incurren, con frecuencia, en diferentes tipos de falsos positivos. Por lo tanto, es factible un esquema de combinación mediante votación: aplicar los detectores elementales en un modo de operación "permisivo", y seleccionar los candidatos más votados. Lógicamente, el proceso acumularía el tiempo de ejecución de todos métodos subyacentes.

Para evitar el aumento del tiempo, proponemos un **esquema de combinación secuencial**: se aplican los detectores uno tras otro, y los candidatos rechazados en una etapa se eliminan para la siguiente. De esta forma, los sucesivos detectores no trabajan con toda la imagen sino con los candidatos supervivientes. En caso de usar sólo dos detectores, uno de ellos tiene el papel de generador de candidatos, y el otro de verificador. Esta idea se representa gráficamente en la figura 3.30.

Por ejemplo, suponiendo que la detección con integrales proyectivas se utiliza como verificador, el proceso sería como el siguiente:

1. Ejecutar el método alternativo de detección de caras, ajustando el algoritmo a un modo de funcionamiento que produzca un bajo número de falsos negativos, aun cuando el número de falsos positivos sea grande.

2. Para cada región resultante del paso 1, hacer:

   a) Ajustar el rectángulo contenedor de cara a la posición y proporción del modelo de caras usado en el detector mediante proyecciones.

   b) Calcular la proyección vertical del rectángulo y alinearla respecto del modelo MV_{cara}.

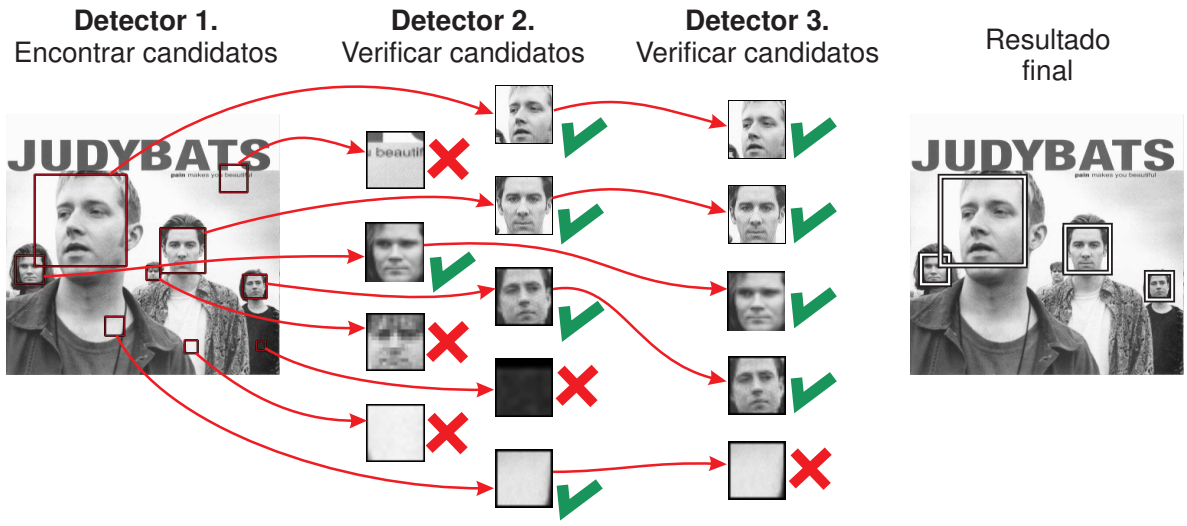


Figura 3.30: *Esquema global del método de combinación de detectores. El primer detector se aplica sobre toda la imagen, produciendo caras candidatas. Los sucesivos detectores se encargan de verificar los candidatos. Sólo los candidatos que pasan todas las etapas se obtienen en el resultado final.*

c) Si la distancia resultante del alineamiento vertical es mayor que cierto umbral, descartar el candidato.

d) En otro caso, calcular la proyección horizontal de la región de ojos y alinearla respecto del modelo MH_{ojos}.

e) Si la distancia de alineamiento horizontal es mayor que cierto umbral, descartar el candidato. En otro caso, aceptarlo.

3. Agrupar los candidatos resultantes y eliminar los solapados.

El alineamiento al que se refieren los pasos 2.b) y 2.d), es el algoritmo 2.4 para el alineamiento rápido de integrales proyectivas. Obsérvese que la distancia resultante de ese proceso se utiliza aquí como el criterio de verificación del candidato.

Combinación con verificador alternativo

En contraposición al método anterior, también es posible que el detector de integrales proyectivas sea el ejecutado en primer lugar, y el método alternativo desempeñe el papel de verificador de candidatos. Esta estrategia es más prometedora cuando el tiempo de ejecución resulta crítico –teniendo en cuenta que el método aplicado en primer lugar será, normalmente, el responsable de la mayor carga computacional–. Pero, evidentemente, cualquier cara no detectada en la fase inicial tampoco lo será posteriormente.

El esquema de combinación de detectores es análogo al presentado en el punto anterior:

1. Ejecutar el detector de caras mediante integrales proyectivas, en un modo de funcionamiento “permisivo”.

2. Para cada región resultante del paso 1, hacer:

   a) Seleccionar un rectángulo contenedor de la cara, con un margen suficiente para el detector alternativo.

   b) Aplicar el método alternativo de detección de caras al rectángulo seleccionado.

   c) Si se encuentra una cara, aceptar el candidato; en otro caso, descartarlo.

3. Agrupar los candidatos resultantes y eliminar los solapados.

En la figura 3.31 se muestra un ejemplo de ejecución de los dos métodos combinados, usando el detector de caras de Viola y Jones [110], como técnica alternativa.

Figura 3.31: *Comparación de resultados de los métodos de detección combinados, sobre la imagen Argentina.gif (CMU/MIT). a) Detector de Haar verificado con integrales proyectivas (tiempo de ejecución: 585 ms). b) Detector de proyecciones verificado con detector de Haar (tiempo de ejecución: 212 ms). Tiempos medidos sobre un Pentium IV a 2,6GHz.*

Podemos adelantar un hecho que se contrastará ampliamente en los experimentos: el primer método combinado ofrece mejores resultados de detección, aunque el segundo reduce muy significativamente los tiempos de ejecución (en el ejemplo, es casi 3 veces más rápido).

3.4. Resultados experimentales

El propósito de los experimentos descritos en esta sección es múltiple. El objetivo principal es demostrar la viabilidad práctica de las técnicas de detección propuestas, contrastándolas con los resultados que ofrecen algunas de las principales alternativas disponibles.

Hay, fundamentalmente, dos formas de llevar a cabo la comparación: (1) ejecutando distintos detectores sobre los mismos datos; o (2) utilizando bases de caras estándar. Ambas opciones tienen sus inconvenientes y limitaciones. Por un lado, la primera está limitada por la disponibilidad de los métodos alternativos. La comparativa debería incluir los mejores detectores del estado del arte, pero no todos están accesibles de forma pública y gratuita.

Por otro lado, la segunda se basa en trabajos previos publicados sobre esa base estándar. Sin embargo, la comparación no es todo lo precisa que cabría esperar, debido a la variedad

de consideraciones introducidas por los autores: diferentes definiciones de cuándo la cara está detectada o no; falta de datos relevantes en las publicaciones; ajustes específicos para cada base concreta; etc. Además, si repasamos el estado del arte en la sección 3.2, vemos que la mayoría de los trabajos previos utilizan sus propios conjuntos de imágenes.

Aparte del estudio comparativo, otro gran objetivo de las pruebas es analizar el comportamiento de las técnicas propuestas frente a distintos factores de entrada y ajustes de los algoritmos. De esta manera, queremos encontrar los puntos fuertes y débiles propios de las técnicas, y las circunstancias bajo las cuales resulta más adecuada su aplicación.

Desarrollo de los experimentos

Vamos a describir y presentar los resultados de dos experimentos comparativos sobre sendas bases de caras. En el primero usamos la base de caras propia (UMU) y varias implementaciones disponibles de detectores propuestos por otros autores. En el segundo manejamos la base estándar CMU/MIT [153], y los resultados expuestos en algunos artículos.

Conjuntamente con el código de detección de caras propiamente dicho, se han programado diversas aplicaciones orientadas al uso de los detectores en diferentes escenarios: ejecución interactiva sobre imágenes fijas; detección sobre entrada de vídeo; y procesamiento por lotes de conjuntos de imágenes etiquetadas. Este último es el usado en los experimentos descritos en esta sección. La forma gráfica del interface de usuario se muestra en la figura 3.32.

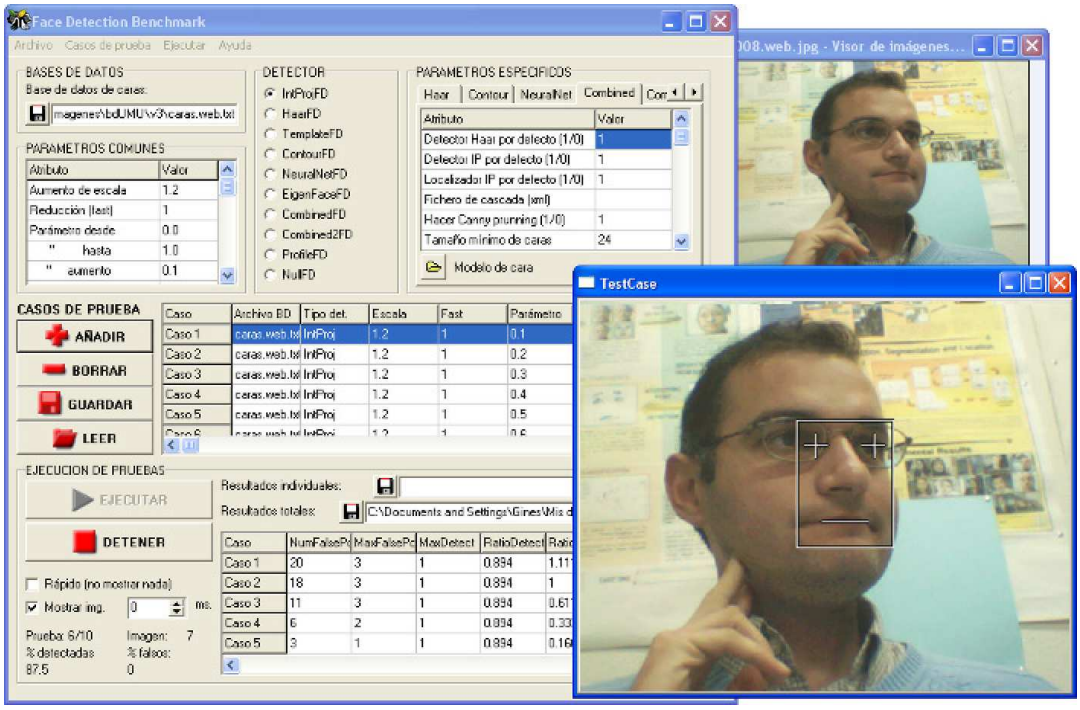


Figura 3.32: *Aplicación creada para la ejecución de los experimentos de detección. En la parte superior de la ventana, los datos que definen cada caso de prueba (base de datos de caras, detector, parámetros comunes, etc.). En medio, el conjunto de casos de prueba añadidos. Abajo, los resultados totales de los casos. A la derecha, visualización de la detección actual.*

Dentro de las pruebas estudiaremos tanto el algoritmo que usa exclusivamente integrales proyectivas como los dos métodos combinados. Obviamente, se espera que los segundos produzcan mejores resultados en cuanto a los ratios de detección. Sin embargo, pretendemos comprobar que la técnica básica es capaz de ofrecer una eficiencia computacional muy superior a los restantes métodos, manteniendo buenas tasas de detección.

Evaluación de los algoritmos y métricas de detección

Un análisis completo debe tener en cuenta todos los criterios que expusimos en el apartado 3.1.2. En particular, recordemos que no sólo es importante el porcentaje de detección, sino también su proporción con el número de falsas alarmas, y la relación de ambos con la eficiencia computacional.

El criterio para decidir si una cara ha sido *detectada correctamente* es el siguiente. Dada una cara etiquetada –por un operador humano– y el rectángulo contenedor devuelto por el algoritmo, se calcula la máxima distancia euclídea entre las esquinas correspondientes de ambos cuadriláteros, dos a dos. Si esa distancia máxima es menor que el 30 % del ancho etiquetado, decimos que la cara ha sido detectada. En otro caso, decimos que ha ocurrido un *falso positivo*, o una falsa alarma.

El *ratio de falsos positivos* se mide en función del número total de imágenes procesadas, mientras que el *ratio de detección* está, lógicamente, en proporción al total de caras etiquetadas por el humano.

Más específicamente, en los experimentos que detallamos a continuación, los parámetros medidos para cada técnica sobre cada conjunto de imágenes son los siguientes:

- **FP=1 %**, **FP=5 %**, **FP=10 %**, **FP=20 %**, **FP=50 %**: ratio de detección del método cuando el porcentaje de falsos positivos es del 1 %, 5 %, 10 %, 20 % o del 50 %, respectivamente[7].
- **Máx. det.**: máximo porcentaje de detección del algoritmo, independientemente del ratio de falsas alarmas, dentro de los modos de operación con los que se ejecuta el detector.
- **eer**: ratio de error igual. Porcentaje de error del detector cuando el ratio de falsos positivos es igual al porcentaje de caras no detectadas. Igual que antes, el valor puede ser obtenido, en caso necesario, mediante interpolaciones en la curva ROC.
- **Tiempo**: tiempo medio por imagen de ejecución del detector, incluyendo desde la lectura del fichero hasta la obtención del resultado.

En relación al último parámetro, la tabla 3.2 resume las principales características del ordenador en el que se han llevado a cabo las pruebas. Estos datos resultan relevantes a efectos

[7] Evidentemente, no todos los detectores pueden ser ajustados para producir exactamente un 1 % o un 10 %, por ejemplo, de falsos positivos. En consecuencia, estos valores son obtenidos mediante una interpolación lineal de los puntos más próximos de la curva ROC correspondiente. Por ejemplo, sea un método con un 77,7 % de detección para un 1,3 % de falsos positivos, y 74,5 % para 0,9 % falsas alarmas; decimos que su ratio de detección para FP=1 % es del 75,3 %.

de analizar la eficiencia computacional de los distintos detectores, y estudiar la posibilidad de aplicarlos en tiempo real. Como se puede observar, se trata de un ordenador personal medio.

Procesador	Intel (R) Pentium IV
Velocidad del procesador	2,60 GHz
Memoria caché	8 Kb (1^{er} nivel) + 512 Kb (2^{o} nivel)
Memoria RAM	1024 Mb (DDR)

Tabla 3.2: *Características del sistema informático usado en la ejecución de las pruebas.*

La mayor parte los experimentos realizados maneja las imágenes de la base de caras propia, que hemos denominado "base UMU". Los resultados de estas pruebas son presentados en el apartado 3.4.2, desglosando diversos aspectos del conjunto de imágenes y del funcionamiento de los detectores. A continuación, en el apartado 3.4.3 abordamos la cuestión del coste computacional de los diferentes métodos. Por último, en el apartado 3.4.4 exponemos los resultados obtenidos con la base CMU/MIT, contrastándolos con los de otros trabajos previos que constituyen el estado del arte en la disciplina. En la sección 3.5 se destacan las conclusiones más relevantes de estos experimentos. Pero antes, vamos a detallar los métodos de detección alternativos usados en la comparativa.

3.4.1. Métodos alternativos de detección

Tanto la técnica de detección de caras mediante proyecciones como los dos métodos combinados han sido implementados en el entorno de trabajo descrito en el capítulo 1, es decir, en lenguaje C++, con el compilador Borland C++ Builder 6, bajo Windows XP, y haciendo uso de las librerías de procesamiento de imágenes y visión artificial Intel OpenCV beta 5, e Intel IPL 2.5 [35]. En general, se han cuidado los aspectos de eficiencia computacional en el diseño del código. Por ejemplo, el cálculo de las proyecciones utiliza imágenes integrales [188], tal y como vimos en el apartado 3.3.2.

Además de los algoritmos basados en proyecciones, disponemos de cuatro métodos alternativos de detección de caras con los que realizar las comparativas: dos técnicas incluidas en las librerías OpenCV [35], una técnica de código libre adaptada a nuestro entorno [152], y un método básico implementado desde cero. Vamos a describirlos muy brevemente.

IntProy - Detección de caras mediante proyecciones

La implementación del detector mediante integrales proyectivas sigue las pautas expuestas en el desarrollo del método en la sección 3.3. Por omisión, se utiliza un factor de reducción $f = 1,2$ y en el caso de imágenes a color se proyecta el canal rojo.

El tamaño de los modelos de proyección es de 30 puntos para MV_{cara} y 24 para MH_{ojos}. En principio, estos valores determinan el tamaño mínimo de las caras detectables; más adelante discutiremos cómo superar esta limitación. Los modelos son de tipo media/varianza y han

sido obtenidos a partir de un conjunto de 374 caras capturadas de TDT y no incluidas en la base UMU. La forma concreta de los modelos se puede ver en la figura 3.33.

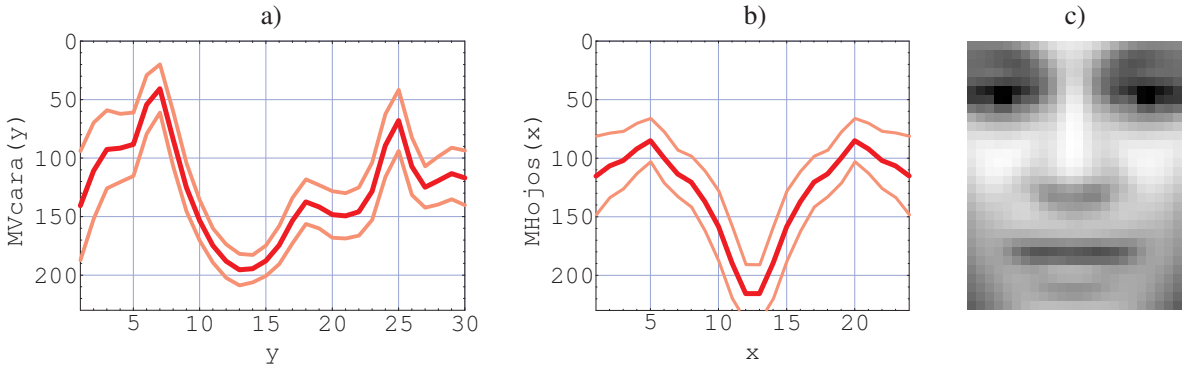


Figura 3.33: *Modelos utilizados en el detector mediante integrales proyectivas, y en los métodos combinados. a) Modelo de proyección vertical de la cara, PV_{cara}. b) Modelo de proyección horizontal de los ojos, PH_{ojos}. c) Imagen media de cara asociada a las regiones proyectadas, de 24 × 30 píxeles.*

Haar - Detección mediante filtros de Haar y AdaBoost

Se trata del método de detección de objetos desarrollado en [188] y mejorado en [110]. Como ya describimos en la sección 3.2, se basa en la detección de características puntuales, lineales, o de bordes, inspiradas en los filtros de Haar. Esas características son la entrada para un conjunto de *clasificadores débiles* mediante árboles de decisión. Componiendo muchos de ellos con el algoritmo AdaBoost, se obtienen *clasificadores combinados*. Por último, existe una *cascada* de estos clasificadores combinados, es decir, se aplica secuencialmente un número de ellos, resultando positiva la detección si todos ellos son ciertos.

La implementación disponible en las librerías Intel OpenCV [35], es debida a Rainer Lienhart, uno de los impulsores de la técnica. Se ofrecen varios detectores de caras ya entrenados, uno de ellos específico para rostros de perfil. El código es bastante eficiente y, aunque se describe como un detector genérico, hay algunas optimizaciones y ajustes específicos para el caso de las caras humanas. Por ejemplo, se puede utilizar el operador de bordes de Canny [22], para descartar regiones uniformes –y, por ello, poco probables para contener caras–, haciendo el proceso más rápido.

Por otro lado, se puede modificar el factor de escala (que por defecto, vale 1,1), y el tamaño mínimo de las caras detectadas (hasta un mínimo de 20 × 20 píxeles). En nuestros experimentos, todos los parámetros han sido ajustados para permitir una comparación justa con los restantes detectores. En particular, el tamaño mínimo de cara se establece a 24 × 24 píxeles, y el factor de escala es ajustado –como en todos los restantes métodos– a valor 1,2.

Desafortunadamente, la operación disponible no permite controlar directamente el parámetro que determina la frontera aceptación/rechazo de los candidatos. No obstante, se puede establecer el número mínimo de regiones solapadas para aceptar un candidato. Por omisión vale 3, es decir, el test cara/no cara debe ser positivo en al menos 3 regiones próximas para

detectar una cara. Modificando este valor conseguimos movernos por distintos puntos de la curva ROC.

Haar+IP - Detección combinada con Haar y verificación con proyecciones

Como describimos en el apartado 3.3.5, a partir de varios detectores de caras es posible construir un método combinado, intentando aprovechar las ventajas de los primeros. En concreto, esta idea fue desarrollada usando las proyecciones como uno de esos detectores elementales.

Este primer método combinado aplica la detección de Haar para obtener las caras candidatas; después, las integrales proyectivas se usan para verificar los candidatos. El detector de Haar es aplicado normalmente en un modo "permisivo", con alto número de detecciones y de falsos positivos, esperando que los segundos sean eliminados en el proceso de verificación. Además, se aplica una heurística adicional: en caso de que no se encuentre ninguna cara, se aplica el detector basado en proyecciones.

IP+Haar - Detección combinada con proyecciones y verificación con Haar

En este segundo mecanismo combinado se utiliza el detector de integrales proyectivas en primer lugar, ejecutado también en un modo "permisivo". Los candidatos resultantes son extraídos a un tamaño de 60×60 píxeles, y después son verificados con el detector de Haar. Igual que antes, se aplica la heurística de ejecutar el segundo método si no se consigue encontrar ninguna cara.

En los dos métodos combinados, los algoritmos elementales utilizan los ajustes típicos que han sido indicados para IntProy. Por ejemplo, los modelos de proyección asociados a la cara son los mismos que aparecen en la figura 3.33.

NeuralNet - Detección de caras mediante redes neuronales

A pesar de su relativa antigüedad –la implementación data de 1999–, la técnica subyacente es uno de los métodos clásicos y más exitosos para la detección de caras humanas. La implementación es debida a Henry Rowley [152], uno de los pioneros en la detección mediante redes neuronales. El autor ofrece gratuitamente su código para uso no comercial, en el que aparecen varias funciones para detectar rostros en imágenes en color o en escala de grises. Algunas funciones trabajan con caras rotadas (respecto del plano de la imagen), y con distintos giros laterales (perfil izquierdo, perfil derecho, etc.).

Todo el código disponible ha sido trasladado a nuestro entorno de pruebas. Además, aunque las funciones originales no permiten ajustar la frontera de decisión cara/no cara, se han modificado para que ese valor se pueda pasar como parámetro de los procedimientos de detección[8]. La eficiencia computacional de las funciones disponibles no es muy alta, como

[8]Debemos aclarar que ese umbral de aceptación/rechazo existía ya en el código original, aunque tomaba un valor constante que no se podía modificar con los parámetros de las funciones disponibles. Por lo tanto, el cambio

veremos más adelante. No obstante, pensamos que no se debe a la implementación en sí, sino que es implícita al propio funcionamiento del método.

El paquete ofrece también funciones para la localización de componentes faciales, que utilizaremos en el siguiente capítulo.

TemMatch - Detección de caras mediante *matching* de patrones

Para tener un rendimiento base con el que poder contrastar los métodos más avanzados, hemos implementado desde cero una técnica de detección de caras basada en búsqueda de patrones (en inglés, *template matching*). El detector simplemente realiza una búsqueda multiescala, aplicando para cada resolución la función de OpenCV que realiza el *matching* de patrones. Todos los resultados que superan un umbral se toman como caras candidatos. Finalmente, se lleva a acabo un proceso de agrupación y eliminación de candidatos solapados, como el definido en el apartado 3.3.4.

La operación permite trabajar con imágenes y patrones en color o en escala de grises, y con diferentes medidas de *matching*: suma de diferencias al cuadrado, producto escalar y correlación. Usaremos la última, que es la que suele ofrecer mejores resultados. El patrón utilizado ha sido entrenado con el mismo conjunto usado para los modelos de proyección. Su tamaño es de 24×30 píxeles, y su forma se muestra en la figura 3.33c).

Aunque la técnica no sea muy prometedora, hay dos aspectos que la hacen interesante para esta comparativa. Por un lado, es un método capaz de funcionar bien con casos sencillos, donde aparezca una cara destacada sobre un fondo uniforme. Por ello, puede servir como una base para medir la complejidad implícita del conjunto de imágenes de prueba. Por otro lado, el modo de funcionamiento es conceptualmente parecido a la detección con integrales proyectivas, en cuanto a que no se utilizan clasificadores complejos, sino una simple medida de distancia al modelo –1D o 2D, según el caso–. Sin embargo, veremos que las integrales proyectivas son capaces de mejorar sustancialmente los resultados del método 2D.

Cont - Detección de caras mediante agrupación de contornos

Esta técnica de detección facial está incluida dentro de las librerías Intel OpenCV [35], como una funcionalidad experimental. Implementa una estrategia ascendente de detección (ver el apartado 3.2.2) basada en contornos. En primer lugar, se aplica una serie de binarizaciones de la entrada según un conjunto de umbrales predefinidos. Para cada umbral, se obtienen los contornos en la imagen binaria correspondiente. Finalmente, se buscan agrupaciones de contornos que coincidan más o menos con un patrón predefinido de cara, según unas restricciones geométricas dadas.

Debemos aclarar que, como los demás métodos basados en invariantes, está más orientado a localización que a detección; es decir, cuando existe una sola cara en la imagen. Por ello, no se espera que los resultados obtenidos estén entre los mejores. Sin embargo, lo incluimos en la

no supone una merma en la eficiencia o la capacidad de detección del algoritmo.

comparativa por ser el único de los métodos disponibles no basado en aspecto. En el siguiente capítulo veremos que estas funciones pueden aplicarse también en problemas de localización de componentes faciales.

3.4.2. Comparación de resultados sobre la base UMU

Como ya mencionamos en el capítulo 1, la base de caras UMU está orientada fundamentalmente hacia aplicaciones que manejan entrada de vídeo –capturas de televisión digital o analógica, de cámara web o de DVD–. No obstante, para aumentar la variabilidad del conjunto se añaden 34 imágenes tomadas de la base CMU/MIT, en su mayor parte escaneos de fotografías analógicas, algunas de ellas de periódicos o revistas.

En total, la base UMU está compuesta por 737 imágenes, con una gran variedad de fuentes de adquisición, resoluciones, número de personas por imagen, expresiones faciales, etc. Algunas han sido ya mostradas previamente. En estas imágenes aparecen 853 caras, que han sido localizadas manualmente en las posiciones de ojos, nariz y boca, calculando a partir de las mismas los rectángulos contenedores.

En la figura 3.34 se muestran las curvas ROC obtenidas para los 3 métodos propuestos en esta tesis, más los 4 detectores de caras incluidos en la comparativa. Recordemos que estas curvas representan los porcentajes de detección frente al ratio de falsos positivos, para los distintos modos de funcionamiento de cada detector.

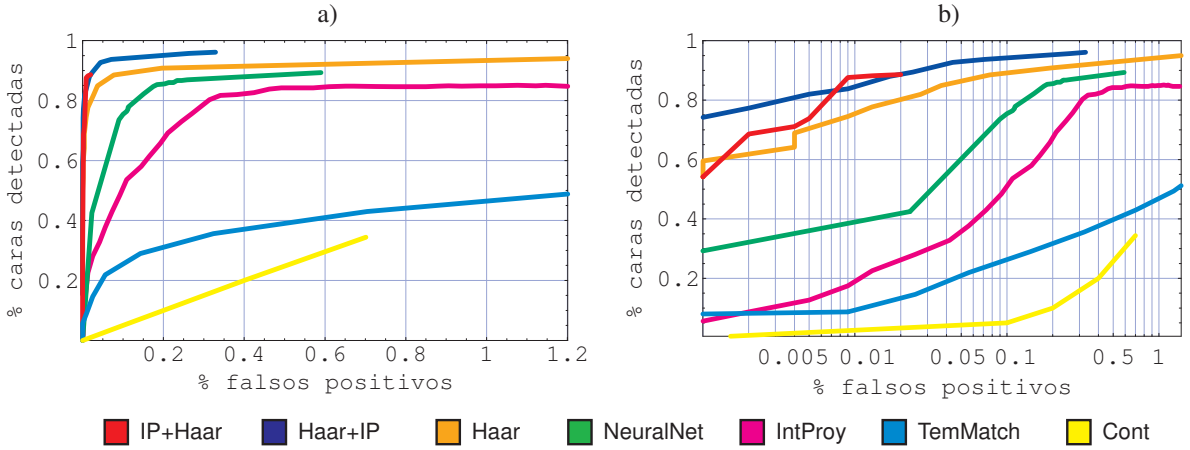


Figura 3.34: *Curvas ROC de los diferentes detectores analizados sobre la base de caras UMU. a) Curvas ROC de los 7 detectores, con escala lineal. b) Las mismas curvas pero usando una escala logarítmica en el eje horizontal. Abajo se muestra la leyenda con los colores usados para cada detector.*

La gráfica de la figura 3.34b) utiliza una escala logarítmica para los falsos positivos, con el objetivo de hacer hincapié en los valores bajos de este parámetro.

Hay que notar que algunos detectores pueden trabajar en más modos de operación que otros. Por ejemplo, la implementación del método basado en contornos no admite parámetros de control, de manera que sólo ha sido ejecutado en un único modo de trabajo[9].

[9]Se traza una línea desde el punto (0,0) para este método, por ser un modo de operación trivial y presente en

En relación con lo anterior, vemos que no todas las curvas se mueven en todo el rango de falsos positivos. El método combinado IP+Haar es un ejemplo extremo de esta situación, ya que no sobrepasa el límite del 2 % de falsos positivos, para un 89 % de detecciones. En principio, esto no es un hecho negativo: si el máximo ratio de detección de cierto método es del 89 %, no es un inconveniente –sino más bien todo lo contrario– que no se generen muchas falsas alarmas. Distintos motivos (de implementación o implícitos al método) pueden dar lugar a que el número de falsas detecciones no aumente indefinidamente, aun cuando se relajen los parámetros del detector.

Los resultados numéricos pormenorizados se exponen en la tabla 3.3. Se indican algunos puntos concretos de las curvas ROC, según el ratio de falsos positivos.

Método detección	**Ratios de detección**					**Máx. det.**	**eer**	**Tiempo (ms)**
	FP=1 %	**FP=5 %**	**FP=10 %**	**FP=20 %**	**FP=50 %**			
IntProy	18,8	35,6	50,8	67,2	84,2	85,1	25,7	**85,2**
Haar	75,3	86,1	88,9	90,8	91,8	95,3	10,9	292,5
NeuralNet	18,5	55,0	75,4	85,5	88,6	89,3	16,4	2337,7
Haar+IP	84,3	**92,7**	**94,0**	**95,0**	**96,1**	**96,1**	**6,6**	295,6
IP+Haar	**87,8**	88,6	88,6	88,6	88,6	88,6	11,4	97,0
TemMatch	9,1	20,5	25,5	31,1	39,0	74,7	59,2	389,4
Cont	0,5	2,5	5,0	10,0	24,8	34,4	67,1	120,3

Tabla 3.3: *Resultados de los distintos detectores de caras sobre la base UMU. La entrada son 737 imágenes que contienen en total 853 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

Las figuras 3.35, 3.36 y 3.37 muestran algunos de los resultados más representativos de los mecanismos propuestos para IntProy, Haar+IP e IP+Haar, respectivamente. Se han incluido tanto ejemplos de funcionamiento correcto como algunos errores típicos de cada método. Se pueden ver resultados de los métodos alternativos en la figura 3.38.

A continuación vamos a discutir las conclusiones más importantes de los resultados presentados en la tabla 3.3 y en la figura 3.34. Empezaremos con una valoración global de los datos, para desgranar después diversos aspectos de interés, como la influencia de la resolución, la fuente de adquisición, la inclinación de las caras o el canal de color. A medida que profundicemos en estos puntos, iremos añadiendo resultados adicionales.

Valoración global de los resultados

Existe un criterio inmediato para juzgar la bondad de un detector: cuanto más alta sea su curva ROC, para todo el rango de falsos positivos, mejor será. Si nos fijamos en la figura 3.34, son muy pocos los casos en los que las curvas se cortan, de manera que es fácil establecer una clasificación de los métodos. De acuerdo con esta regla, el **mejor resultado** lo obtiene el detector combinado **Haar+IP**. Su comportamiento es muy bueno para todos los modos de operación. No sobrepasa el 32,9 % de falsos positivos (en términos absolutos, 243 fallos),

cualquier detector.

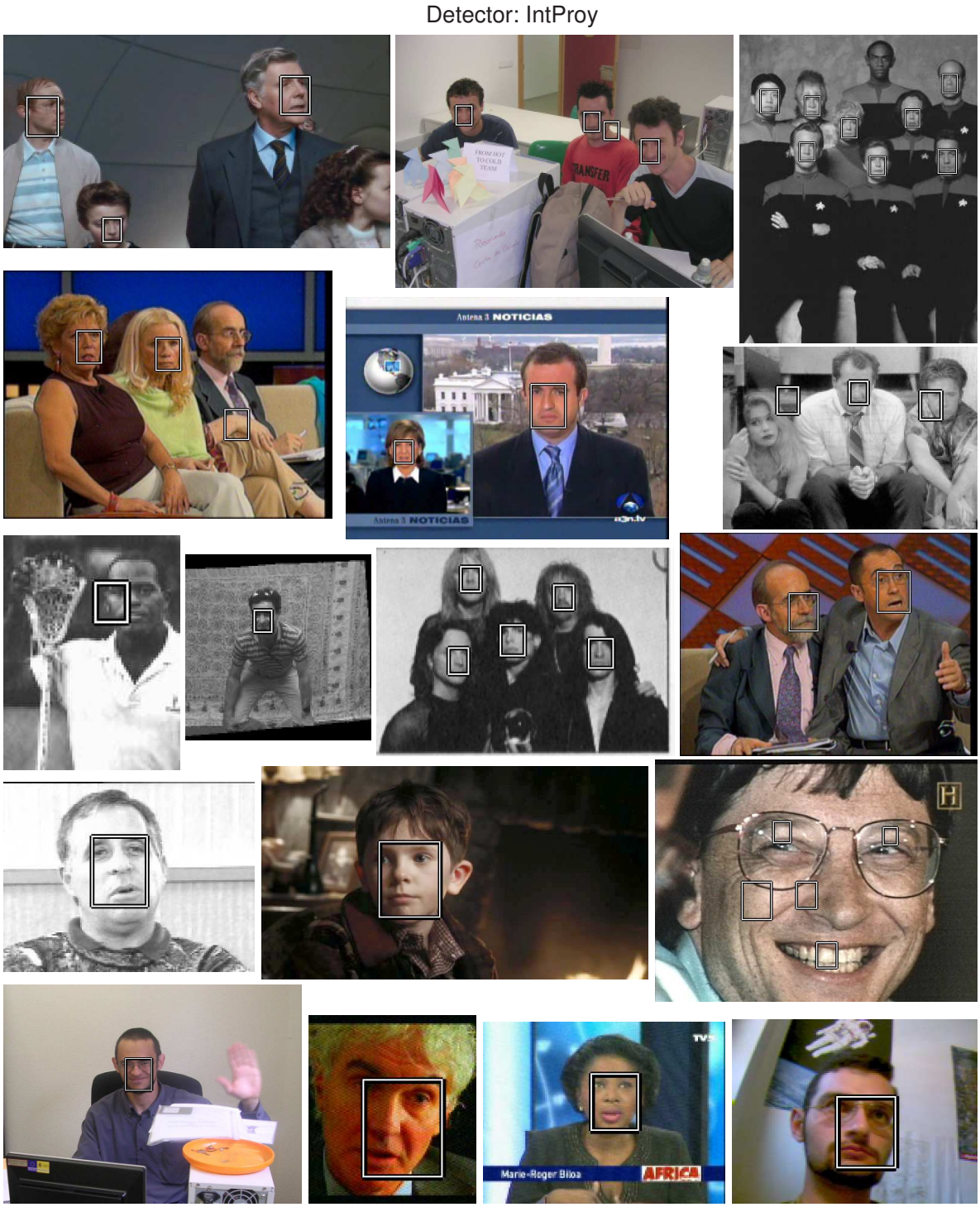


Figura 3.35: *Algunos ejemplos de resultados del detector de caras IntProy sobre la base UMU. En todos los ejemplos mostrados se han usado los parámetros por omisión del detector.*

logrando para ese punto el máximo de detección, un 96,1 % (lo que supone perder sólo 33 rostros). Es más, para un 74 % de caras detectadas genera únicamente una falsa alarma. Este resultado viene a demostrar la gran utilidad de las proyecciones como un mecanismo de

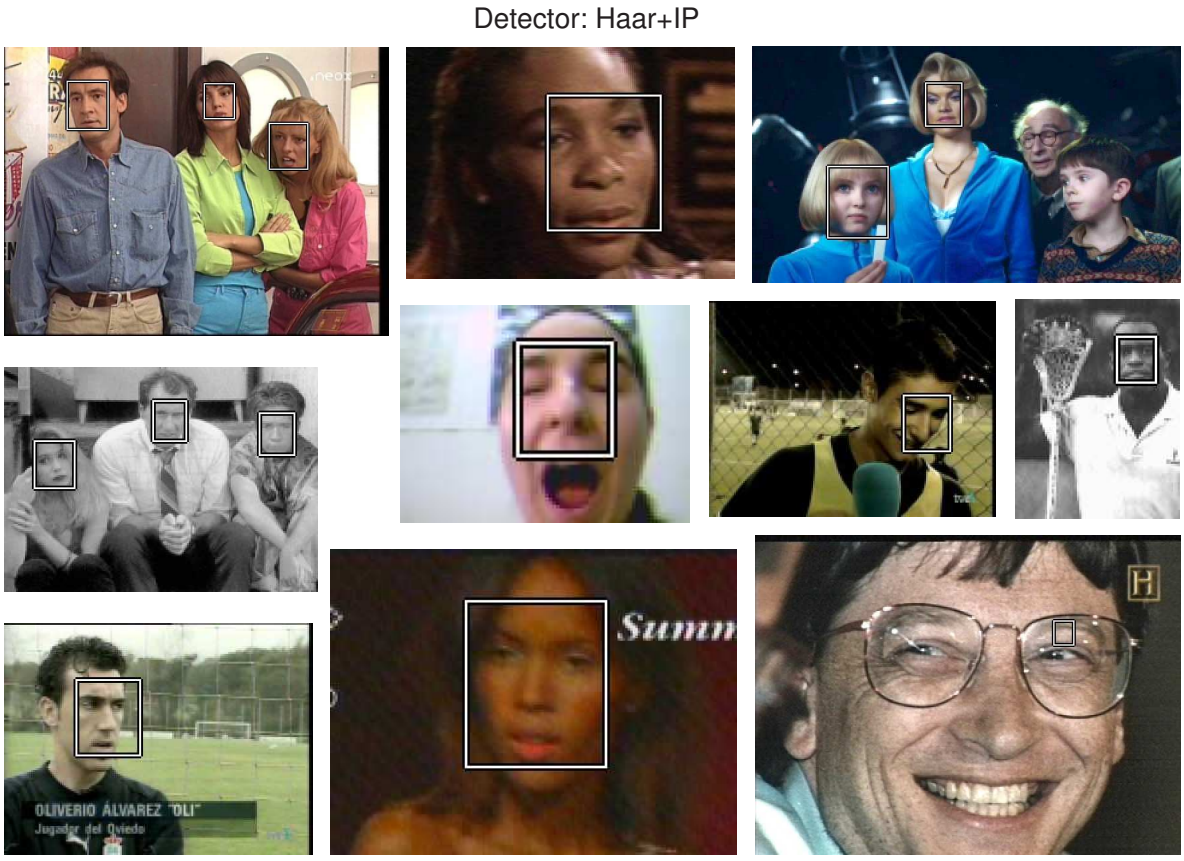


Figura 3.36: *Algunos ejemplos de resultados del detector combinado Haar+IP sobre la base UMU. En todos los ejemplos mostrados se han usado los parámetros por omisión del detector.*

verificación de los candidatos.

En **segundo lugar** podemos situar el método de detección **IP+Haar**, que incluso llega a superar a Haar+IP en ciertos casos. Su gran virtud es la de conseguir un buen porcentaje de detección, del 88,6 %, para un ratio extremadamente reducido de falsos positivos, de sólo el 2 % (15 casos). El primer valor está limitado por la caras encontradas por el detector basado en proyecciones, mientras que el segundo está relacionado con la fiabilidad del proceso de verificación. Junto con la conclusión obtenida para Haar+IP, podemos deducir que ambos métodos elementales incurren en falsas alarmas bajo distintas circunstancias. Ésta es la razón por la que los algoritmos combinados consiguen un rendimiento tan elevado.

El **tercer lugar** de la clasificación lo ocupa **Haar**, que es el mejor de los detectores no combinados, y con un amplio margen sobre sus competidores. En relación a Haar+IP, su curva ROC se encuentra aproximadamente unos 5 puntos por debajo, para la mayor parte de los modos de operación. Su número óptimo de detecciones, del 95,3 %, lo consigue para un total de 1923 falsos positivos (por encima del 2600 %). El mayor obstáculo de este método es la dificultad para encontrar caras que están próximas a los bordes de la imagen, aunque aparezcan con buen tamaño y definición. En la figura 3.38 se puede ver un ejemplo de esta situación.

En **cuarto puesto** estaría **NeuralNet**. Su mejor ratio de detección supera al de IP+Haar,

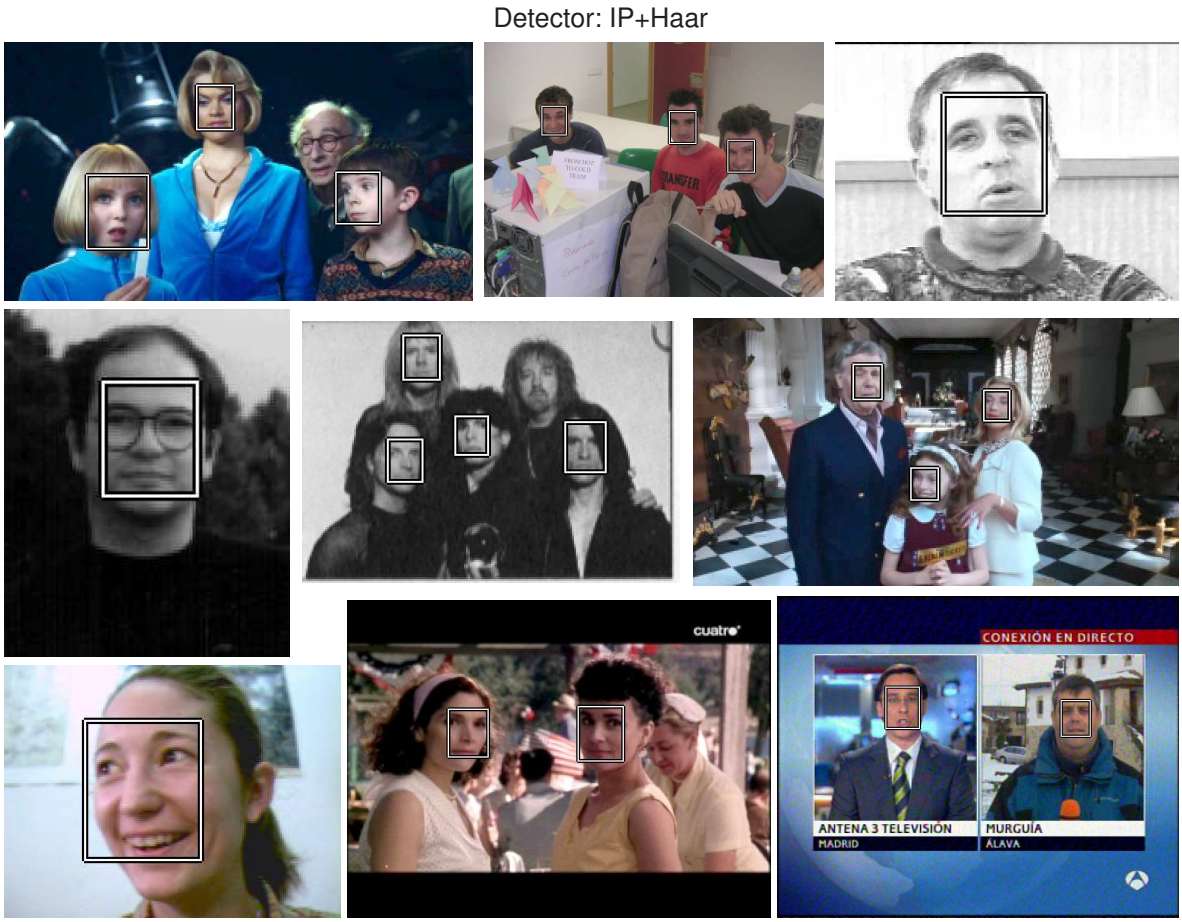


Figura 3.37: *Algunos ejemplos de resultados del detector combinado IP+Haar sobre la base UMU. En todos los ejemplos mostrados se han usado los parámetros por omisión del detector.*

pero lo hace para un 59 % de falsas caras. Además, presenta el inconveniente de que al intentar reducir ese ratio, el número de detecciones disminuye significativamente. Es difícil ajustar los modos de operación del método, que pasa muy rápidamente del 74 % de detecciones con un 9 % de falsas alarmas, a un 42,5 % para 2,3 % de no caras. La figura 3.38 muestra un caso donde el método incurre en un número muy alto de fallos, incluso para un ajuste restrictivo del parámetro aceptación/rechazo. El problema es la existencia de una textura que el algoritmo detecta repetidamente como cara. Un mejor entrenamiento podría reducir este error, aunque en general el problema siempre estará presente en mayor o menor medida.

El comportamiento de **IntProy** es análogo al de NeuralNet: buenos ratios máximos de detección, pero con dificultades al bajar el número de falsos positivos. En la gráfica de la figura 3.34b) se aprecia esta forma común, aunque siempre con mejores ratios para NeuralNet. Frente a las otras alternativas, IntProy aplica un mecanismo de clasificación muy sencillo, basado en simple distancia a un modelo medio. Éste parece ser el mayor obstáculo del método –más que el uso de proyecciones en sí–, y el que origina a una alta ambigüedad en la decisión cara/no cara. En cualquier caso, los ejemplos de la figura 3.35 demuestran que las

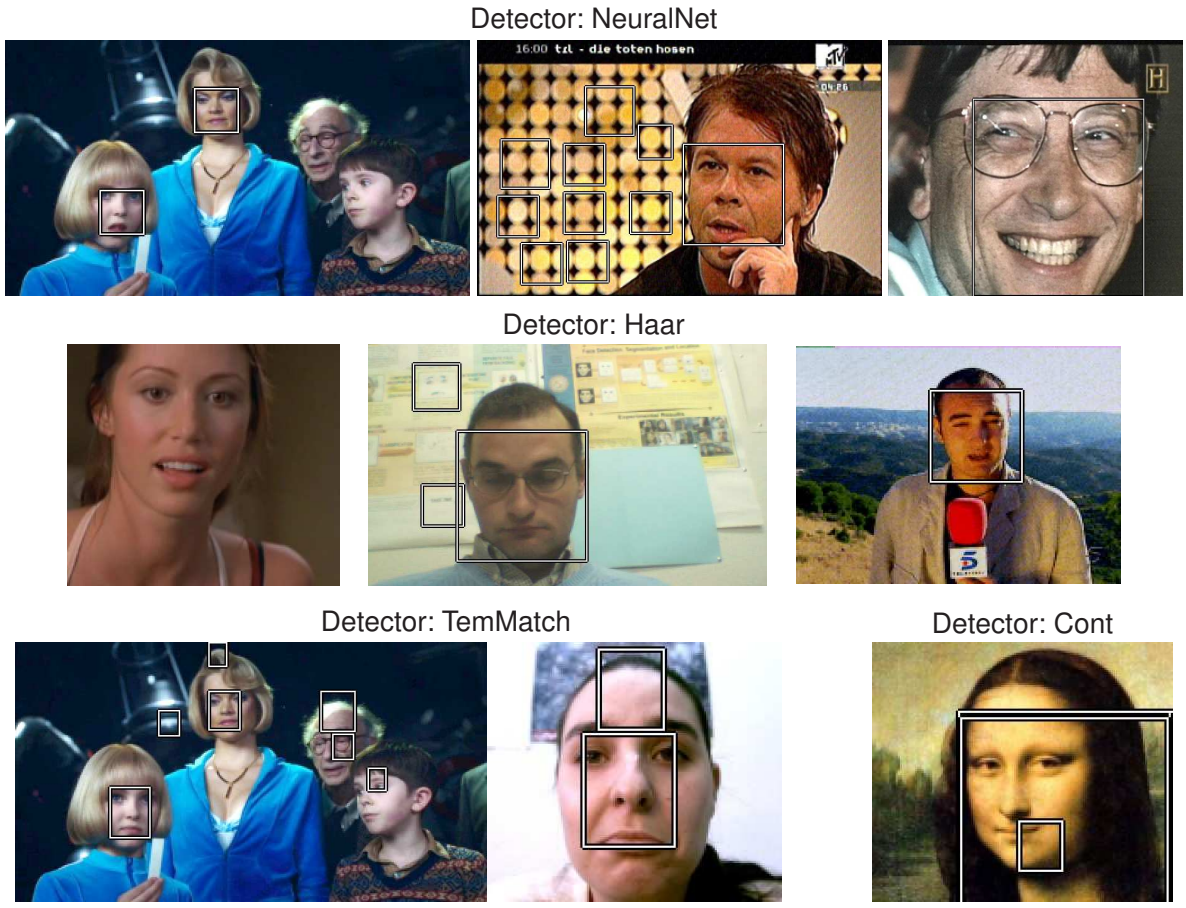


Figura 3.38: *Algunos ejemplos de resultados de los detectores alternativos sobre la base UMU. En todos los ejemplos mostrados se han usado los parámetros por omisión de los detectores.*

integrales proyectivas pueden ser usadas más allá del simple problema de localización, como se ha sugerido en trabajos previos [204]. A diferencia de esos otros acercamientos, el método propuesto es capaz de encontrar un número arbitrariamente alto de caras –y no sólo una–, independientemente de que éstas ocupen una fracción pequeña de las imágenes y tengan un fondo complejo.

A mucha diferencia de los restantes detectores se encuentran **TemMatch** y **Cont**. El primero, al igual que IntProy, se basa en una simple métrica de distancia a un patrón medio. Sin embargo, la capacidad de generalización es muy superior usando proyecciones que con patrones 2D. Así, aunque el punto de máxima detección para TemMatch es del 74,7 %, la tasa de falsas alarmas es del 2460 % (18.156 no caras) para ese modo. Con el mismo porcentaje de detección, IntProy genera únicamente 200 no caras (el 27,1 % del total). En conclusión, podemos decir que la información que se pierde en el proceso de proyección no perjudica a la clasificación cara/no cara, sino más bien todo lo contrario. Diferentes métricas de distancia en TemMatch (como suma de diferencias al cuadrado o producto escalar), o la utilización de patrones RGB, no consiguen mejorar los resultados presentados.

Finalmente, los malos datos del detector basado en contornos sólo pueden ser aprovecha-

dos como una medida de la complejidad implícita del conjunto de imágenes. Grosso modo, podemos establecer que en aproximadamente un 1/3 de las imágenes aparece una sola cara, en la que se pueden distinguir claramente los ojos y la boca por niveles de gris. De hecho, se ha implementado un "detector trivial" que encuentra siempre una cara en la posición promedio del conjunto de caras de UMU[10], respecto del tamaño de la imagen. El método trivial alcanza el mismo porcentaje de detección que Cont para un 60 % de falsas alarmas.

Resultados de la detección en función de la resolución de las caras

Previsiblemente, uno de los factores que más pueden afectar al rendimiento de los detectores es el tamaño de las caras en las imágenes, en cuanto que está relacionado con la calidad y definición disponible. Para analizar su efecto hemos realizado una partición de la base UMU en tres grupos de imágenes, en función del tamaño de las caras presentes. En concreto, distinguimos tres tamaños: pequeño, mediano y grande. El criterio de clasificación se basa en la distancia interocular observada. Los márgenes para cada grupo y el número de muestras contenidas son los siguientes:

- **Pequeño**: de 12 a 30 píxeles, 133 imágenes con 207 caras.
- **Mediano**: de 31 a 60 píxeles, 370 imágenes con 403 caras.
- **Grande**: de 61 a 184 píxeles, 234 imágenes con 243 caras.

Algunas imágenes contienen ejemplos de diferentes grupos, por lo que han sido clasificadas según los casos más predominantes. Para cada grupo definido, se repiten los experimentos de detección con los 6 métodos comparados (se omite Cont). Las curvas ROC resultantes se pueden consultar en la figura 3.39. La tabla 3.4 resume los principales parámetros obtenidos de cada técnica.

Método detección	**Tam. pequeño, FP=**			**Tam. medio, FP=**			**Tam. grande, FP=**		
	5 %	**20 %**	**50 %**	**5 %**	**20 %**	**50 %**	**5 %**	**20 %**	**50 %**
IntProy	26,2	60,2	76,5	36,9	67,9	84,6	40,2	69,7	89,1
Haar	78,6	89,0	90,6	91,0	92,8	93,5	76,4	88,4	90,2
NeuralNet	31,7	82,6	84,7	57,5	88,4	89,6	47,5	73,1	90,6
Haar+IP	**86,6**	**92,3**	**93,7**	**94,9**	**97,7**	**98,2**	90,1	**95,9**	**96,2**
IP+Haar	80,6	80,6	80,6	91,5	91,5	91,5	**91,7**	91,7	91,7
TemMatch	26,7	38,3	44,7	16,8	32,6	40,8	16,4	25,9	32,3

Tabla 3.4: *Resultados de los detectores en la base UMU, en función del tamaño de las caras. El grupo "pequeño" contiene 133 imágenes con 207 caras; "mediano" 370 imágenes con 403 caras; y "grande" 234 imágenes con 243 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

Los ratios de detección no parecen estar asociados de una forma trivial y directa con el tamaño de las caras. Es más, en el caso de TemMatch los mejores resultados se obtienen siempre para las resoluciones menores. Aparte de ese hecho, en general sí que se aprecia una

[10]De manera orientativa, esta posición media corresponde a una cara que ocupa un 20 % del ancho de la imagen, centrada en el eje X, y los ojos están a 1/3 de la altura en Y.

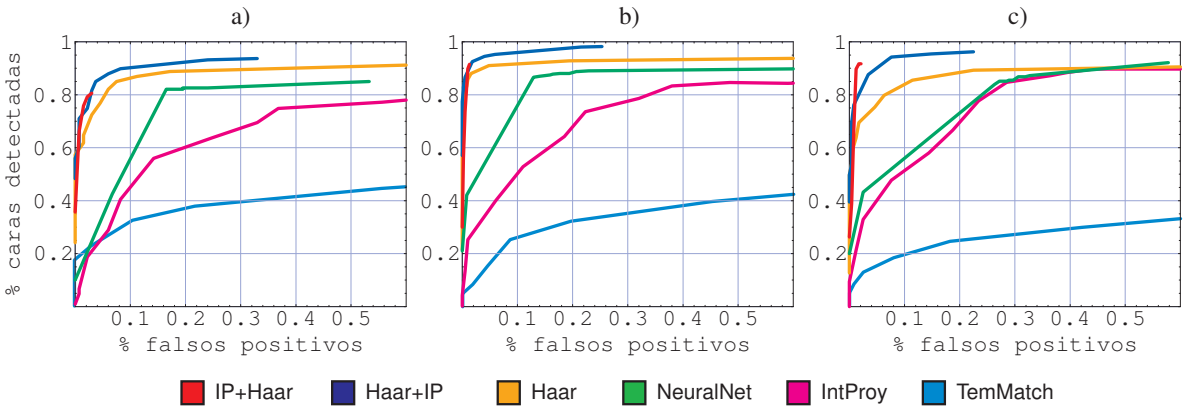


Figura 3.39: *Curvas ROC de los detectores en función de la resolución de las caras. De forma aproximada, los tamaños (distancia interocular, en píxeles) de los diferentes grupos son: a) entre 12 y 30 píxeles; b) entre 31 y 61; y c) entre 61 y 184.*

mejora al pasar del grupo "pequeño" al "mediano". El primero resulta el más problemático para la mayoría de las técnicas. Por ejemplo, IP+Haar no pasa del 80,6 % de detección. Para el segundo grupo, destaca el buen dato de Haar+IP, que encuentra el 98,2 % de las caras con sólo un 25 % de falsos positivos.

Sin embargo, las caras con una alta resolución no siempre son detectadas mejor que las medianas. De hecho, sólo IntProy aprovecha la mayor resolución de las caras para mejorar sus resultados. Gracias a ello, consigue igualar e incluso superar a NeuralNet y, para un alto número de falsas detecciones, también a Haar. Los demás algoritmos bajan ligeramente su rendimiento respecto del grupo intermedio. Sólo los métodos combinados consiguen mantener unos buenos porcentajes. En este fenómeno pueden influir otros hechos colaterales, como la mayor dificultad implícita del tercer grupo, en el que aparecen ejemplos de sombras, giros y expresiones faciales.

Otra conclusión interesante es que los detectores combinados son siempre más fiables que los métodos que los componen. La mejora parece estar en función de la efectividad relativa de ambos. Así, en "pequeño", donde IntProy obtiene los peores resultados, el método Haar+IP aumenta únicamente 3 puntos de detección respecto de Haar. En "grande" las proyecciones funcionan mejor, y el incremento está por encima de los 6 puntos porcentuales.

Fiabilidad de los detectores frente a la inclinación

A excepción del detector basado en contornos, las demás técnicas analizadas parten de la suposición de que las caras aparecen de frente y sin inclinación. A pesar de ello, todos los métodos admiten un cierto margen de tolerancia, detectando rostros con ángulos moderados respecto del plano de imagen. El propósito de esta prueba es medir y comparar la efectividad de los diferentes algoritmos frente a la inclinación. Para ello, se utilizan las imágenes de la base UMU, que son rotadas en varios ángulos entre -40º y 40º, con saltos de 4 en 4 grados. Para cada caso, se rotan también las posiciones etiquetadas a mano, y se repiten las pruebas

de detección con los métodos disponibles. En este caso, los algoritmos se ajustan a un modo de operación fijo. Además, la distancia máxima para declarar que una cara es detectada se aumenta al 40 % del ancho del rostro.

La figura 3.40 muestra los porcentajes de detección resultantes en función de las inclinaciones, para los distintos métodos. Se omiten los falsos positivos, por ser irrelevantes para esta prueba.

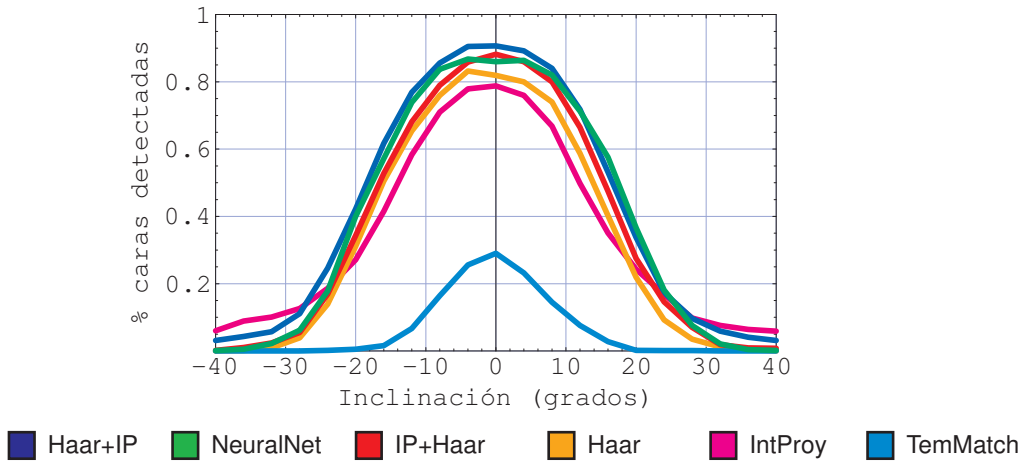


Figura 3.40: *Ratios de detección de los métodos analizados sobre la base UMU, en función de la inclinación de las caras. Las imágenes de entrada son rotadas en la cantidad indicada en el eje horizontal.*

Si bien las curvas se mueven en porcentajes diferentes, todas ellas muestran una forma muy similar, parecida a una campana de Gauss aunque más ancha por el centro. En un intervalo de $\pm 10^o$ la mayoría de los detectores conservan hasta un 80 % de su efectividad; los más robustos son NeuralNet y Haar+IP, que mantienen un 90 % sobre el valor para inclinación 0. El descenso para $\pm 4^o$ es prácticamente inapreciable. Incluso, los detectores Haar y NeuralNet consiguen mejores ratios para -4º que para 0º (por ejemplo, Haar logra un 83,2 % para el primero frente al 81,9 % para el segundo). Esto puede explicarse por la inclinación original de algunas caras, que queda compensada al rotar toda la imagen.

Para el rango de $\pm 20^o$ la degradación de los diferentes mecanismos resulta mucho más drástica. Los ratios de detección difícilmente llegan al 40 %; y esto a pesar de que algunos métodos, como Haar+IP, aumenta ligeramente su ratio de falsos positivos. Si las caras están rotadas más de 20º, los detectores analizados se vuelven prácticamente inutilizables.

Una posible forma de abordar el problema de la inclinación –cuando se quiere admitir un grado de giro arbitrario de las caras–, consiste en repetir el proceso básico del detector para diferentes ángulos de rotación. La conclusión de este experimento es que un incremento adecuado podría estar en torno a los 20º. Es decir, se debería aplicar el detector para 0º, 20º, 40º, etc. En total, se repetiría 18 veces, garantizando una mínima pérdida para los métodos más avanzados.

Influencia de los canales de color

Todos los algoritmos de detección incluidos en la comparativa trabajan con imágenes en escala de grises, sin hacer un uso explícito de la información de color[11]. No obstante, en caso de disponer de imágenes en color –típicamente en el modelo RGB–, es posible utilizar diferentes estrategias en cuanto a cómo debe ser reducida la entrada a un solo canal. En principio, existen muchas posibilidades. La elección básica es seleccionar qué canal, R, G o B, se aprovecha.

Por ejemplo, en el caso del detector basado en integrales proyectivas la decisión consiste en establecer el canal que se proyecta. Para valorar la influencia de este factor, hemos repetido el experimento de detección sobre IntProy, proyectando uno u otro canal. En la prueba se incluye también la luminosidad de los píxeles (referida como "canal gris") que se calcula como: 0,3 R + 0,59 G + 0,11 B. Los resultados del experimento se presentan gráficamente en la figura 3.41 y se detallan en la tabla 3.5.

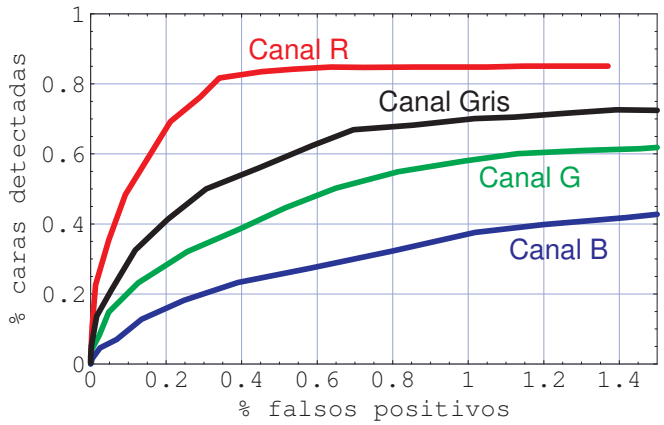


Figura 3.41: *Curvas ROC del detector de caras basado en proyecciones sobre la base UMU, usando distintos canales de color. El algoritmo y los modelos aplicados son los mismos, pero modificando el canal (en el modelo RGB) que se proyecta. El caso "gris" corresponde a la luminosidad de los píxeles.*

Canal de color	Ratio de detección					Máx.		Tiempo
	FP=1 %	FP=5 %	FP=10 %	FP=20 %	FP=50 %	det.	eer	(ms)
Rojo	**18,8**	**35,6**	**50,8**	**67,2**	**84,2**	**85,1**	**25,7**	85,2
Verde	5,5	15,0	20,4	28,3	43,8	63,3	54,3	81,5
Azul	2,4	6,0	9,7	15,9	25,6	47,4	70,0	**77,2**
Gris	10,1	20,2	29,2	40,9	58,4	73,2	44,2	84,1

Tabla 3.5: *Resultados del detector mediante integrales proyectivas sobre la base UMU, usando distintos canales de color. La entrada son 737 imágenes que contienen en total 853 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

Ciertamente, el resultado obtenido es el más previsible: el canal rojo es el que permite una

[11]Sólo el detector basado en comparación de patrones permite usar modelos de cara en color. Sin embargo, los resultados son peores que partiendo de un patrón en escala de grises. Por este motivo, tampoco en los experimentos de este método hemos hecho uso del color.

mejor distinción cara/no cara. Sin embargo, son bastante llamativos los altos márgenes entre el mejor y el peor caso. Por ejemplo, para un ratio de 40 % falsos positivos, el uso de R supone casi 30 puntos porcentuales más de detección respecto al gris, 40 respecto a G, y casi 60 en relación al canal B.

La cuestión subyacente es la cantidad de información que aporta cada canal, y más específicamente el nivel de contraste que ofrece para las caras humanas. De los tres colores primarios, (R, G, B), el predominante en el color de piel es el primero de ellos. Esto significa que produce mayores contrastes entre los tonos claros y los oscuros, lo que favorece una detección más fiable.

Esta conclusión se puede aplicar también a los restantes detectores. Por ejemplo, en un conjunto reducido de pruebas sobre 294 imágenes de la base UMU, el método NeuralNet produce unos ratios de detección de 80,3 %, 71,1 % y 59,6 %, para un número similar de falsos positivos, usando los canales R, G y B, respectivamente. En el caso del detector de Haar, los porcentajes son de 93,6 % para R, 90,1 % para G y 81,2 % para B. En definitiva, la preferencia del canal rojo no es exclusiva del método basado en proyecciones, sino que resulta más adecuado en la mayoría de los acercamientos.

Resultado de los detectores según la fuente de adquisición

Otro de los factores que influyen en el rendimiento de los mecanismos de detección es el origen de las imágenes. La fuente de captura no sólo está relacionada con el ruido y la calidad de una imagen, sino también con el tipo de contenidos y las variaciones más frecuentes. Por ejemplo, si la entrada es de una cámara de videoconferencia, el escenario típico será el de un único usuario situado en primer plano frente a la cámara y mirando de frente. En las imágenes capturadas de televisión, los personajes aparecen normalmente de medio cuerpo hacia arriba, de manera que la cabeza ocupa una menor fracción de las imágenes. Por su parte, en las que proceden de fotografías analógicas, y en especial las tomadas de la base CMU/MIT, pueden aparecer muchas personas, de manera que la definición y calidad de las imágenes es menor. Finalmente, en extractos de películas, pueden ser más frecuentes los casos de iluminación deficiente y no uniforme, maquillaje, giros y expresiones faciales, como sucede con las imágenes de la base UMU.

El análisis de este aspecto se ha centrado en los mecanismos de detección que usan proyecciones: IntProy, Haar+IP e IP+Haar. Como en el estudio de la resolución, establecemos varias particiones del conjunto UMU en función del origen de las imágenes. Los grupos definidos son los siguientes:

- **CMU/MIT**: tomadas de la base CMU/MIT, principalmente procedentes de fotografías analógicas; 34 imágenes con 64 caras.
- **TV analóg.**: capturadas de televisión analógica, fundamentalmente de programas de noticias, reportajes y series; 381 imágenes con 450 caras.

- **TDT**: procedentes de televisión digital terrestre, extraídas de series, noticias y publicidad; 93 imágenes con 120 caras.
- **Webcam**: obtenidas con varias cámaras de videoconferencia en condiciones de interior; 56 imágenes con 57 caras.
- **DVD**: extractos de películas en formato DVD; 140 imágenes con 162 caras.

Algunas imágenes de origen desconocido o de fuentes minoritarias –por ejemplo, de cámara fotográfica digital–, han sido incluidas dentro del grupo más similar.

Las curvas ROC obtenidas para los métodos IntProy y Haar+IP se representan gráficamente en la figura 3.42. Los datos concretos se pueden consultar en la tabla 3.6, que incluye también los ratios de detección de IP+Haar.

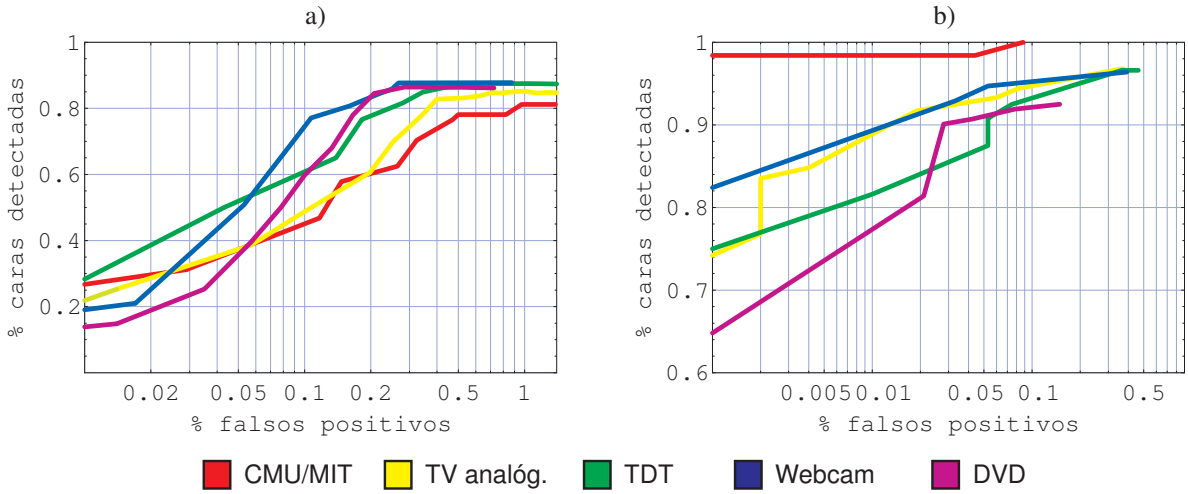


Figura 3.42: *Curvas ROC de los detectores propuestos sobre la base de caras UMU, según el origen de las imágenes. Los datos representados se detallan en la tabla 3.6. a) Curva ROC del detector basado en integrales proyectivas. b) Curva ROC del método combinado Haar+IP.*

Método detección	CMU/MIT, FP= 5 %	CMU/MIT, FP= 20 %	TV analóg., FP= 5 %	TV analóg., FP= 20 %	TDT, FP= 5 %	TDT, FP= 20 %	Webcam, FP= 5 %	Webcam, FP= 20 %	DVD, FP= 5 %	DVD, FP= 20 %
IntProy	34,9	59,9	36,4	60,9	51,1	77,5	47,9	83,3	35,0	83,5
Haar+IP	**98,4**	**100**	**92,9**	**95,5**	87,1	**94,3**	92,3	**94,7**	**91,0**	**92,5**
IP+Haar	84,3	84,3	90,6	90,6	**87,5**	87,5	**92,9**	92,9	85,1	85,1

Tabla 3.6: *Resultados de los detectores basados en proyecciones sobre la base UMU, en función de la fuente de adquisición. El grupo "CMU/MIT" contiene 34 imágenes con 64 caras; "TV analog." 381 imágenes con 450 caras; "TDT" 93 imágenes con 120 caras; "Webcam" 56 imágenes con 57 caras; y "DVD" 140 imágenes con 162 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

En relación a IntProy, los grupos más favorables son Webcam y DVD. El método es especialmente satisfactorio en el primero, alcanzando un ratio de detección del 88 % para un 26 % de falsas alarmas. La mayoría de las caras perdidas ocurren en situaciones complejas, como oclusión parcial, giros o expresiones exageradas. Añadiendo algunas heurísticas de posición

y tamaño, sería posible mejorar los resultados de la detección. Por ejemplo, de las 16 no caras que aparecen en el modo de máxima detección de Webcam, aproximadamente unas 12 pueden ser eliminadas quitando los candidatos de tamaño muy reducido –que pueden descartarse fácilmente en aplicaciones de videoconferencia–. En consecuencia, las proyecciones pueden producir por sí solas buenos resultados en una aplicación de este tipo. Evidentemente, la fiabilidad se ve incrementada con el uso de un método combinado.

Los mayores problemas para IntProy se encuentran en el grupo CMU/MIT, en el que los máximos ratios sobrepasan ligeramente el 80 %. Profundizaremos en las dificultades de este conjunto dentro del apartado 3.4.4, dedicado exclusivamente a esa base de caras. Curiosamente, sobresale el hecho de que Haar+IP alcanza el 100 % de detección en ese grupo, para un 9 % de falsas alarmas (3 fallos). El método subyacente, el detector Haar, lo consigue para un 56 % de falsos positivos (19 no caras). Esto reafirma la efectividad de las proyecciones como mecanismo de verificación de candidatos.

3.4.3. Medidas de eficiencia computacional

Con mucha frecuencia, los métodos de detección más avanzados incurren en un coste computacional elevado. Este factor ha sido obviado en muchos trabajos, que omiten toda referencia al compromiso entre tiempo de ejecución y capacidad de detección. Sin embargo, el coste es importante, porque puede limitar las aplicaciones prácticas de una técnica.

Para cuantificar la eficiencia de los distintos detectores, utilizamos las imágenes de la base UMU. Se toma siempre el promedio de los tiempos de ejecución del conjunto, siendo el tamaño medio de las imágenes de 534×393 píxeles. La tabla 3.7 contiene los tiempos de ejecución obtenidos. Estos mismos datos se representan gráficamente en la figura 3.43. Las características del ordenador usado se pueden consultar en la tabla 3.2.

Tiempo (ms)	IntProy	Haar	NeuralNet	Haar+IP	IP+Haar	TemMatch	Cont
Mínimo	**64,2**	292,1	1575,5	291,0	66,6	269,4	–
Medio	**85,2**	292,5	2337,7	295,6	97,0	389,4	120,3
Máximo	**131,0**	294,0	3277,9	323,5	153,5	646,0	–

Tabla 3.7: *Tiempos de ejecución de los detectores de caras sobre la base UMU. Para cada método de detección, se muestra el máximo, el promedio y el mínimo de los tiempos de ejecución para los distintos ajustes de la técnica. La medida del tiempo es el número de milisegundos por imagen.*

Debemos aclarar que las variaciones entre los casos máximo y mínimo no corresponden a diferencias en el tamaño de las imágenes –puesto que se toman siempre los promedios–, sino al mayor o menor coste según la posición de la curva ROC en la que opera el detector. Así, normalmente la ejecución de los algoritmos es más rápida a medida que se reducen las detecciones y los falsos positivos, mientras que suele ser más lenta cuando aumentan ambos ratios. Este diferencia es mayor en algunas técnicas que en otras, como se puede apreciar con más claridad en la figura 3.43.

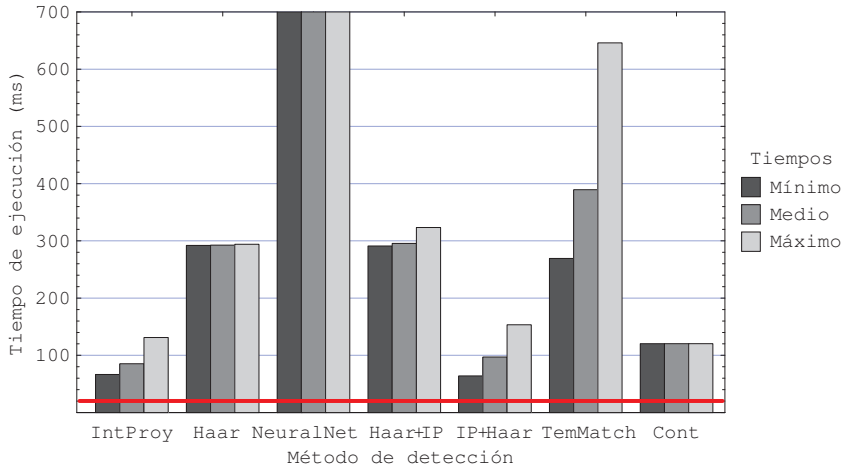


Figura 3.43: *Tiempos medios de ejecución de los distintos detectores sobre la base de caras UMU. Los datos representados se detallan en la tabla 3.7. La línea roja inferior indica el tiempo promedio en leer los ficheros del conjunto.*

Valoración de tiempos y clasificación entre métodos

Por encima de otros resultados, sobresale la elevada complejidad computacional del detector basado en redes neuronales. Cuando el proceso es ajustado para producir un alto número de detecciones, se pueden requerir hasta 3 segundos para analizar una imagen media. Este inconveniente es implícito al propio método, que debe aplicar varias redes sobre cada región candidata. En [153] se proponen algunas soluciones para paliar el problema, básicamente reduciendo el número de regiones sobre las que se aplica la clasificación cara/no cara. Pero, obviamente, esta modificación tendrá también un efecto negativo en los resultados de la detección.

Frente al elevado coste de NeuralNet, el detector basado en integrales proyectivas consigue los menores tiempos de ejecución. El caso promedio permite, aproximadamente, procesar unas 10 imágenes por segundo. A escasa diferencia se encuentra el método combinado IP+Haar, que no pasa de ser un 17 % más lento que el primero. Teniendo en cuenta los buenos ratios de detección del segundo, podría ser un buen compromiso para muchas aplicaciones que requieren un procesamiento en tiempo real.

Por otro lado tenemos los algoritmos Haar, Haar+IP y TemMatch, que se mueven en torno a los 0,3 segundos; es decir, unas 10 veces más rápidos que NeuralNet, pero 3 veces más lentos que IntProy e IP+Haar. Es interesante el hecho de que los dos primeros, y especialmente el detector Haar, son prácticamente insensibles al modo de operación, produciendo los mismos tiempos para todos los ajustes.

Un caso aparte es el detector mediante contornos, en órdenes de la décima de segundo, pero con las grandes limitaciones que hemos visto para conseguir buenos resultados.

Resultados de los métodos de detección combinados

Como ya avanzamos en el apartado 3.3.5, en los algoritmos combinados la mayor parte del coste computacional es debido al método aplicado en primer lugar. Podemos comprobar ahora que no sólo ocurre así, sino que el tiempo añadido por el segundo detector es prácticamente despreciable. De esta forma, observamos que los valores de Haar y Haar+IP son casi iguales, y lo mismo ocurre para IntProy e IP+Haar. Esto es debido a que el segundo método se ejecuta sobre un número reducido de regiones pequeñas, de manera que el tiempo de ejecución requerido es mínimo.

Teniendo en cuenta los buenos ratios de detección de los métodos combinados, el escaso aumento del coste que introducen parece más que justificado. Por lo tanto, la conclusión es que normalmente resultará preferible aplicar este tipo de estrategias, donde los diferentes métodos son ejecutados de forma secuencial, partiendo cada uno de ellos de los resultados del anterior. Aunque no profundizaremos más en este aspecto, podría ser interesante analizar la combinación de más de dos detectores.

Complejidad computacional en función del tamaño de las imágenes

En sentido estricto, el orden de complejidad de un algoritmo viene dado por el aumento del coste en función del tamaño del problema. En nuestro caso, el tamaño de la entrada lo determina el número de píxeles de la imagen y, en consecuencia, el máximo número de caras posible.

En esta prueba usamos la imagen "rot-mei-family.gif" de la base CMU/MIT. Su resolución original es de 2615×1986 píxeles, con un total de 135 caras. La imagen ha sido rotada (pues originalmente está girada), escalada y después tomamos diversos fragmentos de 650 píxeles de alto. Los fragmentos son de ancho 100, 200, 300, y así hasta 2200 píxeles. Sobre cada uno de ellos se aplican los algoritmos de detección en un modo de operación típico, obteniendo los tiempos individuales por imagen. En este caso, no se incluye la lectura de las imágenes.

El resultado del experimento son los tiempos de ejecución en relación al tamaño de la imagen, que se presentan en la figura 3.44. Obsérvese que las medidas del detector NeuralNet están divididas por 10, para encajar su curva dentro de la gráfica.

La principal conclusión de este estudio es que en todos los algoritmos el coste crece linealmente con el número de píxeles de la imagen. En algunos casos, como en NeuralNet y en TemMatch, ocurren mayores oscilaciones respecto a la tendencia media; mientras que en los restantes el crecimiento es más uniforme. Es poco probable que la oscilación sea debida a un error de medición ya que, por ejemplo, NeuralNet tarda unos 40 segundos para el caso 1600. Podría atribuirse más bien al contenido de la imagen para ese tamaño.

En relación a la comparación entre métodos, hay que tener en cuenta que las curvas sólo representan un modo de ejecución del detector. Por este motivo, las diferencias pueden cambiar respecto a las expuestas en la tabla 3.7.

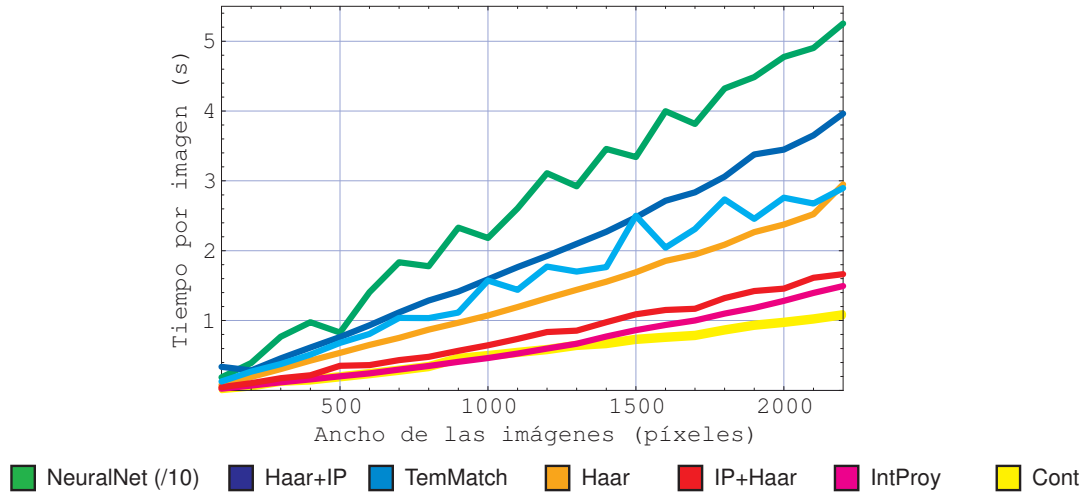


Figura 3.44: *Tiempos de ejecución de los detectores en función del tamaño de la imagen. La altura es siempre de 650 píxeles, mientras que la anchura va aumentando de 100 en 100. Las imágenes son extractos de "rot-mei-family.gif" (CMU/MIT). El tiempo de NeuralNet está dividido por 10.*

El tiempo de ejecución en función del rendimiento

Hasta ahora, la discusión de este apartado se ha centrado básicamente en los tiempos de ejecución. Pero la comparación entre métodos no puede ser independiente de los resultados del detector. Un método no se puede decir mejor o peor que otro, si no se tienen en cuenta los dos factores. Es más, en muchos casos la velocidad del proceso está relacionada inversamente con el tamaño mínimo de las caras detectadas. Aumentando ese tamaño se pueden bajar los tiempos de ejecución, a costa de reducir también los ratios de detección.

Esta idea se puede aplicar, en general, sobre cualquier algoritmo de detección de caras. El modo de conseguirlo sería como el siguiente: (1) reducir la imagen de entrada por un factor, n; (2) aplicar el detector sobre la imagen reducida; y (3) multiplicar las posiciones resultantes por n. Variando el factor de escala, n, conseguimos diferentes compromisos entre el tiempo de ejecución y el orden de complejidad.

Para valorar la viabilidad práctica de esta idea, se ha implementado el mecanismo descrito, comprobando los resultados sobre la base UMU. En particular, nos centramos en el método Haar+IP por ser el más prometedor de los analizados. Recordemos que el proceso incluye la heurística de ejecutar el detector mediante proyecciones si el primero no encuentra ninguna cara. En ese caso, además, el segundo se aplica con la resolución original de la imagen.

La relación obtenida entre tiempos de ejecución y ratios de detección, para distintos ajustes del factor de escala n, se muestran en la figura 3.45.

En la tabla 3.8 se pueden consultar los datos concretos de este experimento.

La tendencia global es hacia un descenso muy ligero de los porcentajes de detección para los primeros factores de reducción. Por ejemplo, al reducir la imagen a la mitad, sólo se pierde sobre 1 punto en los ratios de detección. Sin embargo, el tiempo de ejecución para estos tamaños disminuye de manera mucho más brusca. Así, el tiempo para $n = 2$ en 0,4 veces

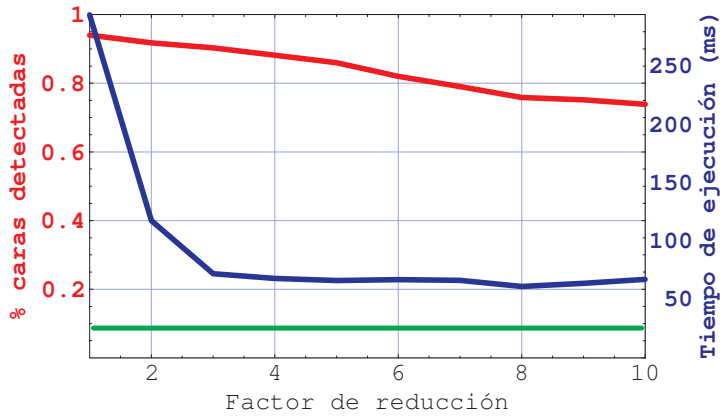


Figura 3.45: *Tiempos de ejecución y ratios de detección del método Haar+IP sobre la base UMU, en función del factor de reducción de las imágenes. La línea verde inferior indica el tiempo promedio en leer los ficheros de la base.*

Factor de reducción	Ratio de detección FP=1 %	FP=5 %	FP=10 %	FP=20 %	FP=50 %	Máx. det.	eer	Tiempo (ms)
1	**84,3**	**92,7**	**94,0**	**95,0**	**96,1**	**96,1**	**6,6**	295,6
2	80,2	91,2	91,8	94,9	94,9	94,9	8,4	126,9
4	55,9	73,1	88,2	90,1	90,7	90,7	11,5	54,5
6	37,0	54,8	82,0	83,5	87,9	88,2	17,0	48,4
8	24,3	41,5	61,6	76,8	81,2	86,5	22,8	47,7
10	10,1	20,2	29,2	40,9	58,4	73,2	44,2	64,1

Tabla 3.8: *Resultados del detector Haar+IP sobre la base UMU, según el factor de reducción de las imágenes. La entrada son 737 imágenes, escaladas según el factor indicado, que contienen en total 853 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

el tiempo original, y para $n = 3$ se divide por $1/4$.

Los primeros valores de reducción ofrecen una relación muy interesante entre tiempo de ejecución y fiabilidad de la detección. Al considerar mayores reducciones, los tiempos prácticamente se estabilizan mientras que los ratios de caras encontradas disminuyen más rápidamente, y sobre todo si nos fijamos en los modos de operación con bajos falsos positivos. No obstante, la heurística introducida consigue mantener altos los ratios de detección alcanzados. Por ejemplo, si consideramos sólo el detector Haar, el máximo número de detecciones para $n = 10$, es del 25,6 %, aunque con sólo 27 ms por imagen.

3.4.4. Comparación de resultados sobre la base CMU/MIT

La falta de conjuntos y protocolos estándar de evaluación de los sistemas de detección facial ha sido ya señalada por algunos investigadores [204]. La base CMU/MIT se puede considerar como un estándar *de facto* en ciertos ámbitos, a pesar de presentar algunas claras limitaciones. Básicamente, su principal debilidad es que no está orientada hacia el tipo de aplicaciones más habituales del procesamiento de caras, haciendo un énfasis excesivo en fuentes poco usadas en la práctica, como la digitalización de fotografías de revistas y periódi-

cos. Además, diferentes trabajos no siempre mantienen las mismas condiciones de experimentación. Así, algunos autores añaden caras no etiquetadas en el conjunto original, otros suprimen las imágenes que contienen dibujos o caricaturas de caras, y el criterio para decidir cuándo una cara está detectada no siempre coincide. Aun así, consideramos interesante presentar los resultados de los métodos de detección analizados sobre esta base.

En nuestro caso, pensamos que las caricaturas de caras no deberían ser incluidas en las pruebas, por lo que han sido suprimidas del conjunto; esto sucede con unas 21 imágenes. En total, el subconjunto utilizado consta de 109 imágenes con 482 caras. Todas las imágenes disponibles están en escala de grises.

Los resultados de los 7 métodos de detección disponibles se encuentran en la tabla 3.9. Como en la base UMU, se señalan algunos puntos de las curvas ROC correspondientes. En este caso no se indica el ratio de error igual, sino el número absoluto de falsas alarmas para el punto de máximas detecciones.

Método detección	Ratio de detección					Máx. det.	Núm. f.pos.	Tiempo (ms)
	FP=1 %	FP=5 %	FP=10 %	FP=50 %	FP=100 %			
IntProy	3,2	6,6	10,7	22,1	27,9	62,6	558	**161,6**
Haar	75,9	82,7	84,3	87,7	89,4	**93,3**	272	404,4
NeuralNet	40,4	76,2	79,8	83,4	86,1	92,5	862	2950,2
Haar+IP	**81,9**	**83,4**	**85,3**	**88,3**	**90,8**	92,7	156	385,7
IP+Haar	42,5	42,9	42,9	42,9	42,9	42,9	**2**	287,4
TemMatch	3,2	6,2	8,3	14,7	16,5	46,2	2020	402,1
Cont	0,0	0,1	0,3	1,3	2,4	2,4	97	345,6

Tabla 3.9: *Resultados de los distintos detectores sobre la base CMU/MIT. La entrada son 109 imágenes que contienen en total 482 caras. Se señala en negrita el mejor resultado obtenido para cada medida estudiada.*

Las discrepancias entre los diferentes métodos se acrecientan, por lo que la gráfica de la figura 3.46 se centra exclusivamente en las técnicas más exitosas. Obsérvese que se utiliza una escala logarítmica para el ratio de falsos positivos.

En la figura 3.46 se han añadido algunos resultados publicados por otros investigadores, y recopilados en [204]. Desafortunadamente, sólo se indica el porcentaje de detección para un número de falsos negativos; en consecuencia, estos métodos aparecen en la gráfica como simples puntos. Los datos concretos de estos detectores se resumen en la tabla 3.10.

Se pueden ver algunos de los resultados del detector mediante proyecciones, para un ajuste estándar de los parámetros, en la figura 3.47.

Valoración global de los resultados

Es evidente la mayor complejidad implícita de esta base de caras, que hace que todos los ratios de detección disminuyan en relación a los valores obtenidos para UMU. Es más, puesto que las imágenes son mayores y existe un número muy alto de caras por imagen, los porcentajes de falsas alarmas se ven incrementados proporcionalmente. Entre los métodos

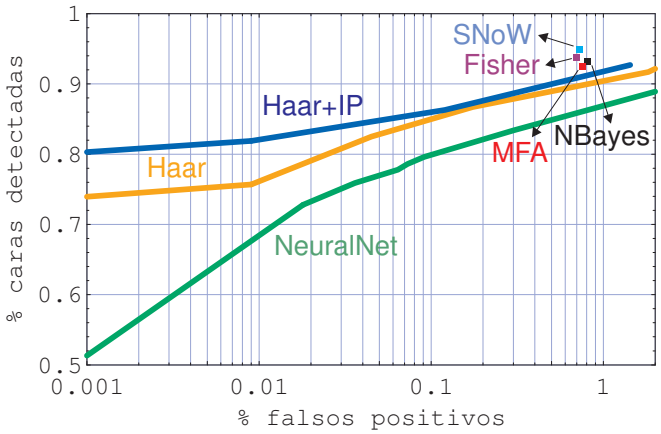


Figura 3.46: *Curvas ROC de los detectores analizados sobre la base CMU/MIT. El conjunto de prueba consta de 109 imágenes con 482 caras. Se representan algunos resultados de trabajos previos, documentados en la tabla 3.10.*

	Método de detección	Ratio de detección	Falsos positivos Total	Falsos positivos Ratio
NeuralNet	Redes neuronales [153]	92,5 %	862	689,6 %
NBayes	Clasificador Naive Bayes [162]	93,0 %	88	70,4 %
	Información relativa de Kullback [30]	98,0 %	12758	10206,4 %
MFA	Mezcla de analizadores de factor [203]	92,3 %	82	65,6 %
Fisher	Discriminante lineal de Fisher [203]	93,6 %	74	59,2 %
SNoW	SNoW con caract. multiescala [205]	94,8 %	78	62,4 %
	SNoW con caract. primitivas [205]	94,2 %	84	67,2 %
	Aprendizaje inductivo [46]	90 %	–	–
	Máquinas Vectores de Soporte [150]	80,7 %	–	–

Tabla 3.10: *Resultados de algunos detectores basados en apariencia sobre la base CMU/MIT, recopilados en [204]. De acuerdo con este artículo, el conjunto de imágenes usado consta 125 imágenes con 483 caras. Los ratios de falsos positivos están en proporción al número de imágenes.*

básicos, sólo Haar y NeuralNet consiguen unos resultados razonables. Igual que en UMU, el segundo ve degradado su rendimiento al intentar reducir el número de falsas detecciones.

El descenso del rendimiento es más acusado para el detector basado en proyecciones. En el punto más alto, alcanza casi un 63 % de detecciones, aunque para un total de 558 no caras (un 512 %). En el siguiente punto vamos a discutir un aspecto clave para comprender este mal funcionamiento del método. A pesar de ello, IntProy sigue demostrando una capacidad de detección muy superior al método basado en comparación de patrones. No sólo está por encima en su curva ROC, sino que el máximo ratio de detección alcanzado es claramente superior. Además, para ese punto máximo del 46 % de caras encontradas en TemMatch, el número de falsas caras es más del triple que en IntProy. Es decir, aunque IntProy presenta problemas para detectar las caras, genera una menor ambigüedad en cuanto a las no caras.

Teniendo en cuenta los pobres resultados de IntProy, es comprensible que el método combinado IP+Haar se encuentre en unos rangos de detección tan reducidos. No obstante, sigue conservando la propiedad de producir muy pocos falsos positivos (en términos absolutos,

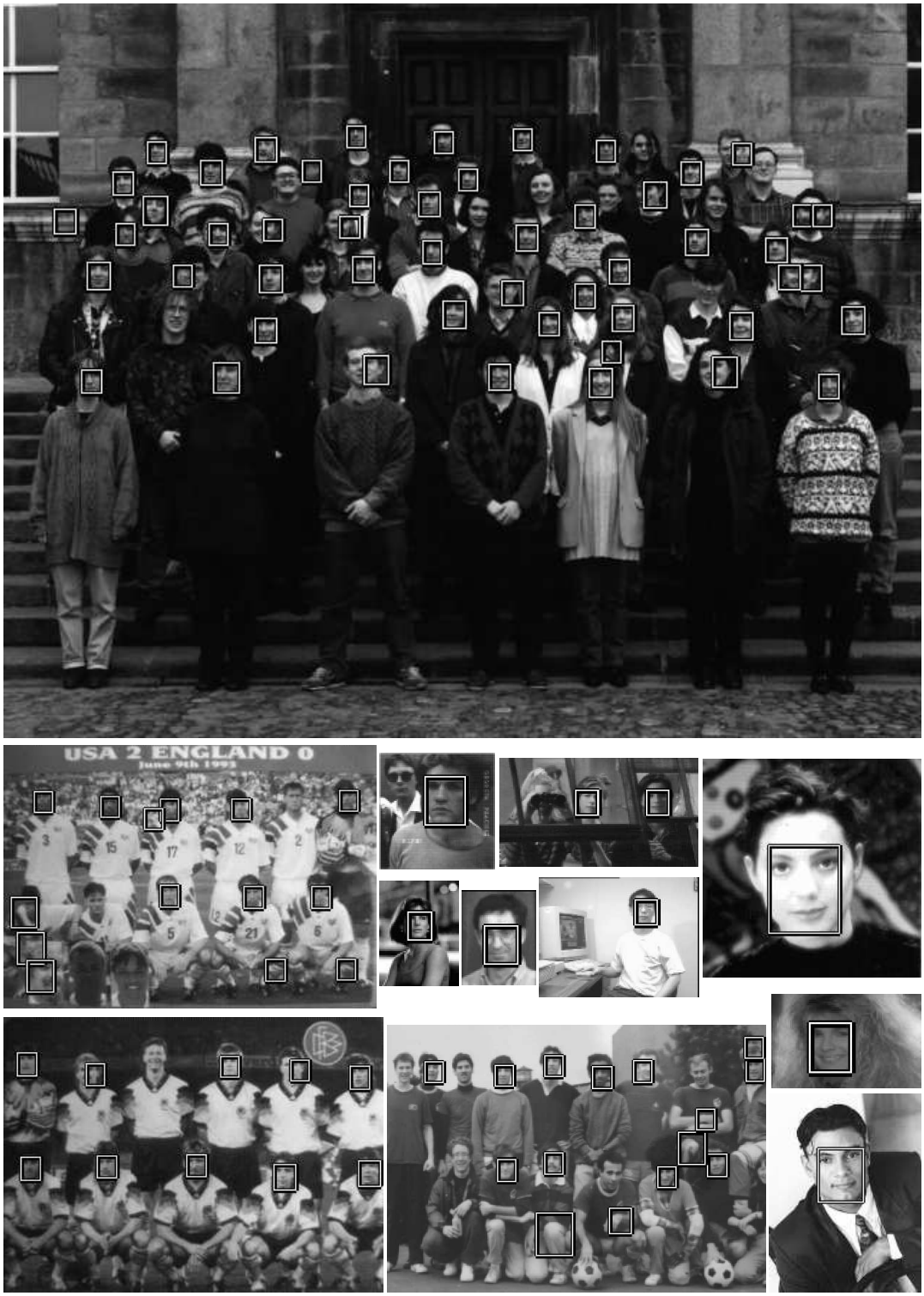


Figura 3.47: *Ejemplos de resultados del detector de caras mediante proyecciones sobre la base CMU/MIT. En los ejemplos mostrados se ha aplicado una ampliación previa de las imágenes del 33 %.*

no pasa de las 2 no caras). También se puede apreciar que la mejora de Haar+IP sobre Haar es menor que en la base UMU. Esta mejora es más significativa cuando trabajamos en un

modo con un reducido número de falsas alarmas, donde la combinación de métodos llega a aumentar hasta 6 puntos los ratios de detección de Haar.

Los resultados alcanzados para Haar+IP son comparables con los otros métodos del estado del arte presentados en la tabla 3.10. En todos ellos, el número de falsas alarmas es relativamente alto, y es difícil adivinar su comportamiento para los modos de operación más restrictivos. Por ejemplo, aunque en NeuralNet el rendimiento máximo es del 92,5 %, para un número comparable de no caras se encuentra por debajo de Haar y de Haar+IP –como se puede ver en la gráfica de la figura 3.46–. En cualquier caso, las diferencias para modos de operación similares no superan los 3 puntos porcentuales. El método basado en aprendizaje inductivo de Duta y Jain [46], podría encontrarse por debajo, aunque no se dispone de información sobre su tasa de falsos positivos.

Algo parecido ocurre con el detector facial mediante SVM [150], donde los autores mencionan un 0,001 % de falsos positivos; pero este porcentaje es tomado en relación al número de ventanas analizadas, de manera que el número real de no caras por imagen es mucho mayor[12]. También es difícil comparar con los resultados de [30], que alcanza un 98 % de detección pero para un número desorbitado de falsas alarmas.

Por otro lado, no se deben olvidar los aspectos de eficiencia computacional. De hecho, la mayoría de los métodos expuestos en la tabla 3.10 resultan extremadamente costosos. Así, por ejemplo, el tiempo del detector mediante SVM [150], sobre la primera imagen de la figura 3.47, de 1280 $\times$ 1024 píxeles, es de unos 16 segundos[13]. Sin embargo, para IntProy el tiempo no sobrepasa los 0,5 segundos.

Tamaño mínimo de las caras detectadas

Uno de los mayores inconvenientes del detector basado en proyecciones sobre la base CMU/MIT es el tamaño de las caras en las imágenes. Teniendo en cuenta que el modelo MH_{ojos} es de 24 puntos y que los ojos ocupan un 60 % de su ancho, la distancia interocular mínima de las caras detectables sería de unos 14,4 píxeles. Sin embargo, en 182 de las caras existentes (un 38 % del total) la distancia es menor de 14 píxeles; y en un 11 % adicional es de sólo 15 píxeles. Con toda probabilidad, el método propuesto fracasará en estos casos, como se puede ver en el ejemplo de la figura 3.48a). Este hecho explica, en buena parte, los bajos ratios de detección de IntProy y también de TemMatch.

Una manera de superar la limitación del tamaño, sin necesidad de modificar los modelos usados, consiste en aplicar un escalado previo a las imágenes. La idea es idéntica a la propuesta en la página 150, pero en este caso realizando un aumento en lugar de una reducción del tamaño de la imagen de entrada.

En la figura 3.48b) se muestra el resultado de IntProy sobre el mismo caso de la figura

[12] Es imposible deducir el número absoluto de falsas detecciones en [150], ya que no indican el número total de ventanas analizadas. Estimativamente, el valor podría estar por encima de las 100 falsas detecciones. De hecho, de tres imágenes mostradas en el artículo, en las tres aparece una falsa alarma.

[13] Valor estimado para un Pentium IV a 2,60GHz. El dato original es 80,1 s en un Pentium a 500Mhz.

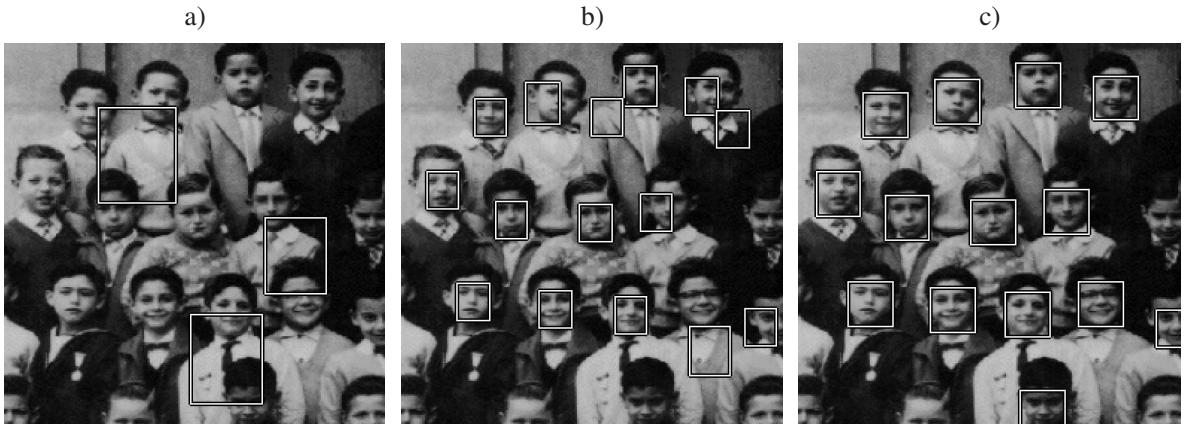


Figura 3.48: *Comparación de resultados de detección sobre la imagen "nens.gif" de la base CMU/MIT. a) Resultado de IntProy sobre la imagen original (27,5 ms). b) Resultado de IntProy sobre la imagen ampliada un 60 % (97,2 ms). c) Resultado de NeuralNet sobre la imagen original (1435 ms).*

3.48a), pero con una ampliación de la entrada a 1,6 veces el tamaño original. La mejora es muy notable, y el resultado se acerca bastante al de NeuralNet. Se encuentran 8 caras y otras 4 regiones están muy próximas a caras existentes. Lógicamente, el inconveniente de esta técnica es el aumento del coste computacional; el tiempo de IntProy se multiplica por 3. Aun así, sigue siendo casi 15 veces más rápido que NeuralNet. Por su parte, el tiempo es prácticamente idéntico al de Haar (con 103,2 ms), el cual pierde 2 caras –las de los bordes– respecto de NeuralNet.

Esta modificación ha sido aplicada sobre otras imágenes de CMU/MIT en las que IntProy falla. La mejora general es bastante sustancial, aumentando los ratios de detección el doble o el triple. Pero siguen existiendo otros obstáculos, como la falta de contraste y resolución, la saturación del brillo, la baja calidad por artefactos de compresión, o los casos de iluminación muy deficiente. En cualquier caso, debemos recordar que nuestro principal objetivo ha sido la detección de caras en fuentes de vídeo (cámaras web, televisión, DVD, etc.), frente al uso de imágenes escaneadas de fotografías analógicas y de periódico, frecuentes en esta base.

3.5. Conclusiones y valoraciones finales

Los resultados de los experimentos demuestran sobradamente que las integrales proyectivas pueden ser utilizadas **por sí solas** para resolver el problema de detección de objetos en general, y de caras humanas en particular. A diferencia de la estrategia usada por otros autores, el método propuesto es capaz de encontrar un número arbitrario de caras en imágenes con fondos complejos. La utilización de modelos, frente al análisis heurístico de las proyecciones, es una de las grandes novedades del mecanismo diseñado.

En comparación con el uso de patrones bidimensionales, las proyecciones ofrecen una **capacidad de generalización** muy superior. En el dominio de las caras humanas, un número reducido de proyecciones permite conservar la mayor parte de la información relevante, que

posibilita la distinción de una instancia de la clase de otra que no lo es. El proceso desarrollado es computacionalmente **muy eficiente**, y resulta especialmente viable en aplicaciones que manejen entrada de vídeo.

Pero la gran potencia de las proyecciones se encuentra en **combinación con otras técnicas** de detección. Con un diseño cuidadoso y un ajuste adecuado del proceso de combinación, es posible mejorar los resultados de los métodos constituyentes y equipararlos con los detectores más avanzados del estado del arte. Y todo esto con un mínimo aumento de la carga computacional. Las integrales proyectivas pueden ser usadas tanto como un proceso para la obtención rápida de los candidatos como para la verificación fiable de los mismos.

Posiblemente, la principal limitación del acercamiento propuesto es la que surge de utilizar un clasificador basado en una simple **distancia a un modelo medio**. Esto supone que todas las caras se ajustan en mayor o menor medida a ese modelo, algo que no siempre está justificado. Aunque el esquema se ha probado robusto frente a muchas fuentes de variabilidad, creemos que podría mejorarse sustancialmente con la introducción de otros mecanismos de clasificación, que admitan un entrenamiento más orientado por los ejemplos –tanto positivos como negativos–, y permitan una variación no unimodal de la clase *cara*. Existen, por lo menos, dos vías muy prometedoras para una posible investigación futura:

- En primer lugar, la clasificación cara/no cara podría basarse en criterios de separabilidad entre clases, usando técnicas de **análisis de discriminantes lineales** [203, 9], sobre las proyecciones. Estos mecanismos buscan espacios de reducida dimensionalidad donde se maximiza la *varianza inter-clase*, minimizando la *intra-clase*. Aplicado sobre las integrales proyectivas –junto con una técnica de *boosting* para seleccionar un conjunto significativo de falsas caras [173, 174]–, se podrían mejorar ampliamente las fronteras de decisión del clasificador de proyecciones usado.

- En segundo lugar, la combinación de diferentes proyecciones podría basarse en la aplicación del **algoritmo AdaBoost** [188]. De esta manera, en lugar de la comprobación secuencial PV_{cara} + PH_{ojos}, se podrían construir muchos *clasificadores débiles* usando diversas proyecciones de las ventanas analizadas, con distintas regiones y ángulos de proyección. El proceso de entrenamiento sería el encargado de crear la combinación óptima de los clasificadores elementales.

Partiendo de los interesantes resultados obtenidos en los experimentos –sobre todo en comparación con el uso de patrones 2D, que utiliza también un sencillo criterio de clasificación–, la mejora que supondría aplicar los anteriores mecanismos resulta muy atractiva. Lógicamente, por las evidentes limitaciones de espacio y tiempo, no se han podido llevar a cabo en el contexto de esta tesis. Pero creemos que se han establecido las bases para el desarrollo futuro de estas extensiones.

3.6. Resumen

El problema de detección de caras humanas presenta enormes desafíos, debidos a la infinidad de apariencias que exhibe el rostro humano bajo distintas condiciones de iluminación, posición 3D, expresión y elementos faciales. El reto que supone modelar esta compleja variedad de aspectos, unido al indudable interés práctico de disponer de detectores fiables y eficientes, ha hecho que la detección facial se haya convertido en uno de los ámbitos de investigación más activos y complejos dentro de la visión artificial. Sin pretender superar a otros métodos más complejos del estado del arte, hemos intentado explotar el potencial de las integrales proyectivas como una técnica que puede aportar nuevas ideas en la resolución del problema. Podemos destacar los siguientes aspectos del método desarrollado:

- El detector mediante proyecciones sigue la filosofía de los **métodos basados en apariencia**: realizar una búsqueda exhaustiva multiescala, en la que para cada región de la imagen se lleva a cabo una clasificación cara/no cara. El proceso se repite con diferentes resoluciones, de acuerdo con un factor de aumento de escala. La diferencia entre unos métodos y otros se limita al mecanismo de clasificación subyacente.

- En nuestro caso, haciendo uso de las herramientas de **modelado** y **manejo de integrales proyectivas**, expuestas en el capítulo 2, la clasificación cara/no cara de las subregiones se basa en las distancias a un modelo de proyección vertical de la cara, PV_{cara}, y a uno de proyección horizontal asociada a la zona de los ojos, PH_{ojos}.

- Más concretamente, el proceso desarrollado se compone de **tres grandes pasos**: (1) búsqueda de candidatos en las proyecciones verticales por tiras, usando el modelo de PV_{cara}; (2) verificación de candidatos mediante proyección horizontal de los ojos, con PH_{ojos}; y (3) agrupación de los candidatos resultantes. En definitiva, podríamos interpretar el algoritmo como una técnica de detección 1,5D, esto es, basada en la composición de patrones 1D.

- Los **experimentos** llevados a cabo demuestran que la técnica propuesta puede ser utilizada en muchos ámbitos de aplicación, siendo especialmente adecuada en aquellos que manejan entrada de vídeo, como los sistemas de videoconferencia o de análisis de contenido de vídeo. Aunque no llegue a mejorar otros métodos basados clasificadores más complejos –y, por lo tanto, mucho más costosos–, las proyecciones presentan una capacidad de **generalización** muy superior a los propios patrones 2D, usando mecanismos de clasificación análogos. Además, el uso de proyecciones supone una **reducción** significativa en el **coste computacional** del proceso.

- Las proyecciones también han sido utilizadas en **combinación con otros métodos**. En concreto, hemos planteado dos posibilidades: usar las proyecciones para obtener un conjunto de candidatos, que después son verificados con un método alternativo; y aplicar las proyecciones como verificador de candidatos, obtenidos con el otro método. El

hecho de que detectores diferentes incurran en falsas alarmas distintas, posibilita que este esquema de combinación produzca normalmente excelentes resultados, equiparables a los de las técnicas más avanzadas.

CAPÍTULO 4

"El rostro de Mae West", Salvador Dalí, 1935

Localización de Componentes Faciales

"Érase un hombre a una nariz pegado,
érase una nariz superlativa [...]
Érase un naricísimo infinito,
muchísimo nariz, nariz tan fiera
que en la cara de Anás fuera delito."
FRANCISCO DE QUEVEDO, *A una nariz.*

El rectángulo contenedor de la cara, resultante del paso de detección, puede ser suficiente en ciertas aplicaciones de visión. Por ejemplo, para un sistema de indexación automática de contenido multimedia podría bastar con etiquetar, *grosso modo*, las posiciones de los individuos existentes en las imágenes. Sin embargo, en la mayoría de los casos resultará necesario refinar la localización de la cara y ofrecer información adicional sobre la situación concreta de los componentes del rostro por separado. El análisis de expresiones faciales y la mayoría de las técnicas de reconocimiento de personas, por ejemplo, requieren que los ojos y la boca estén alineados, de forma más o menos precisa, a unas posiciones conocidas.

En consecuencia, la localización precisa de los componentes faciales se convierte en un problema preliminar para una gran cantidad de sistemas de análisis y procesamiento de caras humanas. La efectividad de estos sistemas, incluso, podría verse seriamente afectada por la inexactitud en las localizaciones obtenidas. Así, por ejemplo, varios estudios [119, 204], han comprobado la pérdida de rendimiento de los reconocedores como consecuencia de las imprecisiones de localización.

A lo largo de este capítulo vamos a tratar la localización facial desde el punto de vista de las integrales proyectivas. En primer lugar, empezamos analizando el problema, los objetivos y las dificultades que plantea en la sección 4.1. En la sección 4.2 hacemos un resumen del estado del arte en localización de componentes faciales, usando una clasificación similar a la de los métodos de detección. Después desarrollamos, dentro de la sección 4.3, el método

propuesto de localización mediante proyecciones. Esencialmente, las proyecciones verticales ayudarán al ajuste vertical de las caras, mientras que las horizontales resolverán la posición en el sentido horizontal. Los experimentos realizados, contrastando el método propuesto con algunas otras técnicas alternativas, se detallan en la sección 4.4. La sección 4.5 hace una valoración global de estas pruebas, señalando las conclusiones más destacadas. Finalmente, se resumen las principales aportaciones del capítulo en la sección 4.6.

4.1. El problema de localización de componentes faciales

Existen algunos indicios neurológicos para pensar que los elementos faciales desempeñan un papel muy relevante en la percepción humana de las caras. Obsérvese, por ejemplo, la figura 4.1; es una muestra del conocido como "efecto Thatcher", descrito por primera vez por P. Thompson [177].

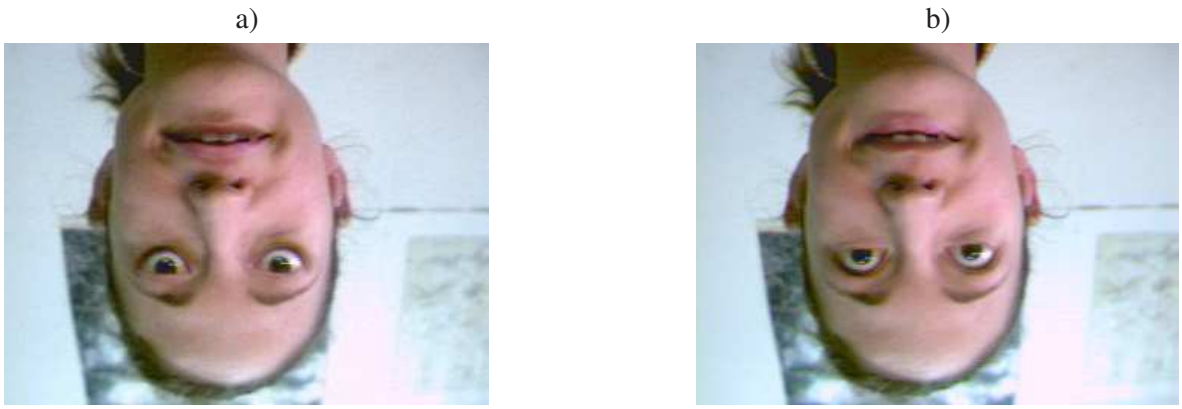


Figura 4.1: *El efecto Thatcher. Tanto la imagen a) como la b) parecen caras humanas normales invertidas. Pero sólo una de ellas lo es. ¿Cuál?*

A simple vista, ambas imágenes parecen ser inversiones de caras de apariencia bastante natural. No da la sensación de existir ningún retoque artificioso en las imágenes, aparte de la rotación de 180º. Sin embargo, si le damos la vuelta al papel, podemos apreciar inmediatamente que uno de los rostros es completamente antinatural. Los ojos y la boca están invertidos, y un poco descolocados de su posición. Pero nuestra percepción inicial era que ambas caras eran correctas, simplemente porque aparecían dos ojos, una nariz y una boca.

Esta ilusión óptica ha sido propuesta como una evidencia de que existen partes de nuestro cerebro adaptadas específicamente a la percepción de caras humanas "derechas", mientras que en las caras invertidas se usaría un mecanismo más genérico. Pero la impresión de "apariencia normal" en la cara retocada artificiosamente, es también un indicio de que otras secciones del cerebro están especializadas en cada uno de los elementos del rostro humano: la cara parece correcta porque los ojos y la boca están derechos, aunque las demás partes de la cara están invertidas.

4.1.1. Elementos faciales y objetivos de la localización

Aparte de las consideraciones neurológicas, es indudable que la localización precisa de los distintos componentes del rostro resulta imprescindible en la mayoría de los sistemas de interpretación de expresiones faciales, reconocimiento de personas, estimación de pose 3D, codificación de vídeo, y, en general, de cualquier aplicación que extraiga información a partir de las caras detectadas. Es, precisamente, la aplicación final la que determina los dos requisitos básicos de un localizador: los elementos faciales a detectar, y la forma en la que se deben describir las posiciones resultantes.

Descripción de las posiciones localizadas

No existe un consenso en cuanto a la mejor forma de describir la posición de los componentes localizados. En muchos trabajos, el objetivo es determinar un *punto medio* para cada componente de interés [169, 211, 213, 59, 199, 38, 128, 89]. Otros autores delimitan los elementos con una serie de puntos que fijan su *extensión horizontal y vertical* [66, 50, 132, 76] –por ejemplo, las esquinas de los ojos–. En otros casos, el resultado de la localización es una *forma geométrica* (rectángulo, elipse) [84, 58], o un *contorno* (por ejemplo, con *snakes*) que envuelve al componente [208, 73, 51]. Finalmente, existen métodos donde el propósito de la localización es el ajuste de un *modelo deformable* [105, 34, 210, 171], por lo que no se lleva a cabo una búsqueda específica para cada parte de la cara, sino de forma global.

Si el destino final del localizador es conseguir una *normalización de la cara* a una forma estándar –como es nuestro caso–, la simple determinación de puntos medios puede ser suficiente. En concreto, con tres puntos –por ejemplo, los ojos y la boca– queda definida unívocamente una *transformación afín* (6 incógnitas, 6 variables). Incluso con sólo dos puntos –por ejemplo, ambos ojos–, es posible definir una *transformación similar* de traslación, escala y rotación (4 incógnitas, 4 variables), para normalizar convenientemente el rostro detectado. El reconocimiento facial de personas es un ámbito donde se aplican este tipo de operaciones, de manera que es innecesaria una localización más detallada del rostro.

Por otra parte, las descripciones más refinadas mediante contornos o puntos delimitadores resultan interesantes en aplicaciones de análisis de la expresión facial, lectura de los labios y en estimación de pose. Por ejemplo, el grado de apertura de la boca se puede deducir de manera más o menos inmediata a partir de su forma geométrica observada [66].

Elementos faciales de interés

Por *componentes faciales* entendemos los órganos –en sentido biológico–, o elementos constituyentes –en sentido más general– de una cara humana normal: cejas, ojos, boca, nariz, etc. Tampoco existe una elección única y universal de los componentes faciales de interés, sino que diferentes métodos trabajan con distintos elementos, descartando otros. Cuanto mayor nivel de detalle requiera una aplicación, más componentes pueden considerarse. Por ejemplo, para un interface perceptual puede ser suficiente con conocer la posición media de ambos ojos;

sin embargo, un sistema de análisis del punto de mirada debe distinguir entre: globo ocular, párpado, iris y pupila.

Entre los elementos faciales con mayor relevancia en la literatura, podemos señalar los siguientes por orden de frecuencia de aparición:

- **Ojos.** Son los componentes básicos, y a veces los únicos que son tratados, [152, 211, 50]. La localización suele darse con la posición media; pero es difícil encontrar una definición precisa, y no siempre está claro si se refiere al centro del globo ocular o de la pupila[1]. En la figura 4.2 se muestran algunas situaciones típicas donde esta imprecisión puede ser significativa. Por ejemplo, ¿cuál es la posición teórica cuando el ojo está cerrado?

Figura 4.2: *Ejemplos de dificultades y ambigüedades de localización debidas a orientación 3D y expresión facial. La cuestión es, ¿dónde estarían los puntos medios en una localización ideal? Extractos de las imágenes de la base UMU: 5010.avi.jpg, 2024.jpg, 617.jpg, 16.jpg.*

  Cuando hablamos de "ojo izquierdo" nos referimos al que aparece más a la izquierda en la imagen, que será normalmente el ojo derecho de la persona (a menos que la imagen esté reflejada, claro), y viceversa. Otra cuestión importante relacionada con los ojos es la **orientación de la cara** en la imagen. La orientación, o inclinación, se suele definir como el ángulo de la recta que pasa por ambos ojos, respecto del eje horizontal.

- **Boca.** Existen varias formas comunes de especificar la localización de la boca. La más sencilla es mediante la posición media del conjunto boca/labios. Lógicamente, esa elección presenta las mismas ambigüedades –o incluso mayores– que para los ojos, como se puede ver en la figura 4.2, y especialmente cuando la boca está abierta. Algunos trabajos buscan también la extensión horizontal de la boca [66, 199, 132, 76]. Por su parte, la extensión vertical está relacionada con su grado de apertura.

- **Nariz.** La localización de la nariz suele tener una importancia relativa muy inferior, ya que dispone de reducida movilidad –y casi siempre asociada al movimiento de la boca– y tiene menor influencia en la expresión facial. Cuando se incluye, la posición buscada es típicamente la punta de la nariz [125, 83], o los orificios nasales [76, 170].

[1] La diferencia puede ser grande si los ojos miran hacia algún lado (ver, por ejemplo, el segundo caso de la figura 4.2). Lógicamente, tiene más sentido utilizar el centro del globo ocular; pero para un sistema de visión artificial será normalmente más sencillo localizar la pupila, al ser más oscura.

- **Otros.** Limitándonos a los otros componentes faciales que han sido objeto de investigación, podemos mencionar las cejas y la barbilla [170, 93, 76]. Podríamos añadir otros componentes, que son objeto de estudio de ámbitos más específicos, como la pupila, en la aplicación mencionada de seguimiento de la mirada, o las orejas y el iris, que son usados en algunos sistemas biométricos existentes [3]. Sin embargo, en estos casos las imágenes están centradas en tales componentes, de manera que caen fuera del procesamiento de caras propiamente dicho.

Definición del problema de localización

Una vez seleccionado un formato de descripción y un conjunto de elementos faciales de interés –lo cual, como hemos apuntado, dependerá de la aplicación final del sistema–, podemos pasar a definir el problema.

Definición 4.1 ***Localización de componentes faciales.***
Dada una imagen y una región de la misma que contiene una cara en una alta proporción, el objetivo de la localización facial es determinar la posición de cada uno de los componentes faciales de interés, refinando de esta forma la posición de la región de cara.

Se supone, por lo tanto, que el localizador recibirá como entrada la salida producida por un detector de caras. El término "alta proporción" significa que se debe admitir cierta imprecisión en los resultados de la detección, en cuanto a la posición y tamaño de las regiones. Será tarea del localizador refinar esa región de cara, a través de la búsqueda de los componentes de interés.

Debemos aclarar que algunos autores se refieren al mismo problema como *extracción o detección de componentes faciales*, en lugar de usar el término *localización*, que se deja para denotar la búsqueda de una sola cara en una imagen.

En nuestro caso, nos vamos a centrar en la **localización de los ojos y la boca**, descritos como puntos situados en sus **posiciones medias**. No obstante, una particularidad del método que vamos a desarrollar es que la boca será situada siempre equidistante entre ambos ojos. En consecuencia, aunque la operación devuelve 3 puntos, implícitamente manejamos sólo 5 grados de libertad.

4.1.2. Desafíos e inconvenientes en la localización

La localización de componentes faciales presenta similares desafíos a la detección de caras humanas. Pero surgen algunas cuestiones específicas que resulta conveniente identificar. Por un lado, el problema se simplifica al asegurarse la existencia de una cara, con sus dos ojos, una nariz y una boca. Pero, por otro lado, al trabajar con objetos de tamaño más reducido se multiplican las dificultades debidas a la variación de apariencia de los componentes. En las figuras 4.2 y 4.3 se pueden ver algunas imágenes usadas en los experimentos con situaciones que pueden complicar significativamente el proceso de localización.

Figura 4.3: *Ejemplos de situaciones complejas en la localización de componentes faciales. Se pueden ver casos de escasa resolución, sombras, oclusión, gafas y barba. De izquierda a derecha, extractos de: Argentina.gif (CMU/MIT), brian.gif (CMU/MIT), natalie1.gif (CMU/MIT), 29.jpg (UMU).*

Vamos a distinguir y clasificar los principales tipos de obstáculos que se deben abordar:

- **Escasa resolución.** Cuando el tamaño de las caras se aproxima al mínimo detectable, la distinción de los componentes faciales por separado es simplemente imposible. Por ejemplo, en una cara de 24×30 píxeles –como la primera de la figura 4.3–, los ojos no ocupan más de 3×3 píxeles. En esas circunstancias, se hace evidente que la localización sólo puede tener lugar dentro de la estructura global de la cara.

- **Expresión facial.** El efecto de las expresiones sobre la apariencia del rostro es mucho más drástico cuando analizamos los elementos faciales individuales. Además, esta situación ocurrirá con mucha frecuencia. Como se puede ver en los ejemplos de la figura 4.2, habrá variaciones si los ojos están abiertos, cerrados, entreabiertos, si las cejas están levantadas, si la pupila mira en una u otra dirección, por no mencionar la infinidad de posibles gestos, aperturas y posiciones de la boca.

- **Oclusión y elementos adicionales.** Una cara con oclusión parcial puede tener algunos componentes ocluidos total o parcialmente. La oclusión de un elemento facial provocará que la hipótesis de partida –en relación a que "existen dos ojos, una nariz y una boca"– no sea necesariamente cierta en todos los casos. El problema puede tener diferentes causas: existencia de elementos faciales (como gafas, bigote, barba), superposición de objetos externos (como la mano en el tercer caso de la figura 4.3), y la desaparición de algunos componentes con giros grandes de la cara. Los primeros, además, dificultan la localización aunque la oclusión no tenga lugar.

- **Sombras**. El efecto de las sombras puede hacer que los ojos, o la boca, no sean más que manchas indistinguibles de píxeles oscuros (como en el segundo ejemplo de la figura 4.3), incluso con resoluciones elevadas. En muchos de estos casos, la sombra es producida por la propia cara.

- **Estructura facial.** Otro aspecto que conviene tener en cuenta es la estructura propia del rostro humano. Los distintos constituyentes de la cara no aparecen en posiciones

aleatorias, sino que deben presentar una estructura coherente: los ojos están encima de la boca, las cejas están sobre los ojos, los ojos y la boca forman un triángulo isósceles, etc. Por lo tanto, el localizador debe garantizar la coherencia del resultado.

Parece claro, como resultado de esta discusión, que las dificultades inherentes al problema –debidas al aumento de variabilidad de los objetos de interés– son equiparables a, cuando no mayores que, las ventajas de partir de una posición inicial de cara. Dicho de otra forma, el problema se puede abordar gracias a la importante restricción de partida; pretender localizar ojos, narices y bocas en las imágenes originales será, con toda probabilidad, inviable, excepto en casos triviales.

Como conclusión, vemos que muchos de los obstáculos identificados apuntan a la conveniencia de realizar una localización basada en la estructura global del rostro, y no en la detección separada de cada componente facial. Este es uno de los principios subyacentes del método que proponemos y desarrollamos en la sección 4.3. El modelo de cara completo se ajustará de forma precisa a la instancia dada, y como resultado de ese ajuste se deducirán las posiciones de los componentes. De esta forma se reduce la influencia de los anteriores factores, e incluso es posible localizar de manera fiable elementos ocluidos completamente.

4.1.3. Criterios y medidas de precisión

Evidentemente, para evaluar los algoritmos de localización de componentes faciales se deben manejar medidas y criterios diferentes de los introducidos en el problema de detección. La principal métrica es la *precisión*, es decir, la distancia euclídea entre las posiciones encontradas y las reales. Sin embargo, se pueden definir otros parámetros que ayuden a cuantificar la efectividad de cada técnica. Vamos a ver los más interesantes:

- **Error medio de localización de los componentes.** Sean p_{ojo1}, p_{ojo2} y p_{boca} las posiciones resultantes de un algoritmo; y q_{ojo1}, q_{ojo2} y q_{boca} las reales. Como acabamos de decir, el error de precisión es el promedio de las distancias entre los puntos devueltos por el localizador y los reales. Este error puede expresarse de diferentes maneras:

  - **Error en píxeles:**

    $$Error^{pixel}_{comp} = \mathbf{E}(||p_{comp} - q_{comp}||) \tag{4.1}$$

    siendo $\mathbf{E}(x)$ la esperanza matemática de una variable x; $||a||$ el módulo de un vector a; y $comp \in \{ojo1, ojo2, boca\}$.

  - **Error relativo a la distancia interocular observada:**

    $$Error^{relat}_{comp} = \mathbf{E}\left(\frac{||p_{comp} - q_{comp}||}{||q_{ojo1} - q_{ojo2}||}\right) \tag{4.2}$$

  - **Error en milímetros:** suponiendo una separación típica entre ojos de unos 70 mm:

    $$Error^{mm}_{comp} = Error^{relat}_{comp} \cdot 70 \tag{4.3}$$

Cabe hacer algunas matizaciones respecto a estas medidas:

- Hablar de "posiciones reales" es una idealización. Normalmente las localizaciones que se toman como referencia han sido etiquetadas por un operador humano y, por lo tanto, están sujetas a cierto error de medición. Ese margen de error se debe tomar como el límite teórico de la precisión alcanzable.
- Posiblemente, no todos los errores de localización tienen la misma trascendencia. Por ejemplo, una desviación en sentido vertical de los ojos puede ser más grave que en sentido horizontal; y será más preocupante si un ojo está desplazado hacia arriba y el otro hacia abajo. Sin embargo, el simple criterio de distancia no lo tiene en cuenta.
- El segundo formato (ecuación 4.2) es el que se encuentra con más frecuencia en la literatura [89, 213, 76, 38, 128, 197]. No obstante, podemos argumentar que el error del etiquetado manual tiene más sentido medirlo en píxeles. Al usar la medida relativa, las imprecisiones (tanto las manuales como las del algoritmo) penalizan más con caras de tamaño pequeño que con las grandes. Por ejemplo, en una cara de sólo 24×30 píxeles, la distancia entre los ojos es de unos 14,4 píxeles, y un error de 1 píxel se transforma en un porcentaje del 7 % respecto a esa distancia.
- Puede tener sentido estudiar el error medio de cada componente (para saber si todos se encuentran con la misma precisión), y también el promedio de todos los componentes (para conocer la precisión global). De igual forma, la varianza en el error es una medida de la fiabilidad y uniformidad del método.

- **Errores de precisión global.** Para evitar algunos inconvenientes de las medidas basadas en distancias de los componentes, se pueden añadir criterios basados en el tamaño y forma global del rostro. De esta manera tenemos una visión más completa de la bondad del localizador. Definimos los siguientes:

  - **Diferencia media de tamaños.** Es la diferencia media entre el tamaño real de la cara, s_q, y el encontrado por el localizador, s_p. En concreto, valdrá: $\mathbf{E}(|s_q - s_p|/s_q)$. Por su parte, el concepto de "tamaño" puede referirse a la distancia entre los ojos (*tamaño en sentido horizontal*), o a la distancia entre el centro de los ojos y la boca (*tamaño en sentido vertical*[2]), es decir:

$$s_q = \left|\left| \frac{q_{ojo1} + q_{ojo2}}{2} - q_{boca} \right|\right| \; ; \; s_p = \left|\left| \frac{p_{ojo1} + p_{ojo2}}{2} - p_{boca} \right|\right| \tag{4.4}$$

  - **Diferencia media de ángulo.** En muchas situaciones es interesante fijarse en la precisión de la inclinación estimada del rostro. Esta medida se define como la diferencia media absoluta entre los ángulos reales y los obtenidos, es decir: $\mathbf{E}(|\alpha_q - \alpha_p|)$.

[2]En nuestro caso, esta segunda medida tiene la ventaja de usar todas las localizaciones devueltas por el método.

El ángulo de la cara se asocia típicamente con el ángulo de la recta que pasa por ambos ojos:

$$\alpha_q = \arctan \frac{q_{ojo2}.y - q_{ojo1}.y}{q_{ojo2}.x - q_{ojo1}.x} \; ; \; \alpha_p = \arctan \frac{p_{ojo2}.y - p_{ojo1}.y}{p_{ojo2}.x - p_{ojo1}.x} \tag{4.5}$$

- **Diferencia media de posición.** La definición de esta medida es equivalente al error de localización de componentes, pero considerando la posición media de los ojos y la boca: $q_{med} = (q_{ojo1} + q_{ojo2} + q_{boca})/3$, y $p_{med} = (p_{ojo1} + p_{ojo2} + p_{boca})/3$ (que normalmente estará próxima a la nariz). Cabe señalar que esta medida no se puede deducir de las anteriores: la posición media puede ser precisa pero con un alto error de los componentes, si la diferencia de tamaño es grande.

- **Número de fallos de localización.** Casi todos los métodos se pueden encontrar ante situaciones frente a las cuales no se puede completar la búsqueda de los elementos faciales. Esto puede suceder cuando la región no contiene realmente ninguna cara; pero también puede pasar con caras de apariencia problemática, debido a los factores ya analizados. Decimos, en estos casos, que el localizador ha fallado.

  Diremos, también, que ha ocurrido un fallo de localización cuando la distancia de algún componente esté por encima de cierto umbral. Esta métrica es la más usada en diferentes trabajos [169, 170, 89, 213, 128, 197], típicamente con un tope de distancia del 25 % o del 20 %, y aplicada sólo sobre los ojos. Formalmente, el ratio de localizaciones correctas es el porcentaje de casos donde se cumple: $\max(Error^{relat}_{ojo1}, Error^{relat}_{ojo2}) < \tau$, siendo τ el umbral de distancia prefijado.

- **Tiempo de ejecución.** Con toda probabilidad, el tiempo de ejecución de los algoritmos de localización no será tan crítico para las aplicaciones como los de detección. No obstante, ante dos métodos con igual fiabilidad, siempre será preferible el que se ejecute más rápidamente y, en general, requiera menos recursos de memoria o tenga menor coste de entrenamiento.

También se han propuesto y utilizado en la literatura diferentes formas de representar gráficamente los resultados de un localizador dado. Las tres más habituales, que se muestran en la figura 4.4, son las siguientes:

- **Gráfica de densidades de localizaciones.** Se trata de un histograma 2D, donde las celdas corresponden a posiciones normalizadas sobre una cara estándar. Cuanto más concentradas estén las nubes de densidad en torno a los componentes, mejor será el resultado. En la figura 4.4a) aparece un ejemplo típico, en el que se ha dibujado una cara en el fondo para poder interpretar más fácilmente el resultado.

- **Curva de distribución de las distancias.** Esta curva es un histograma 1D de las distancias euclídeas de cada elemento facial al etiquetado manual. La figura 4.4b) muestra

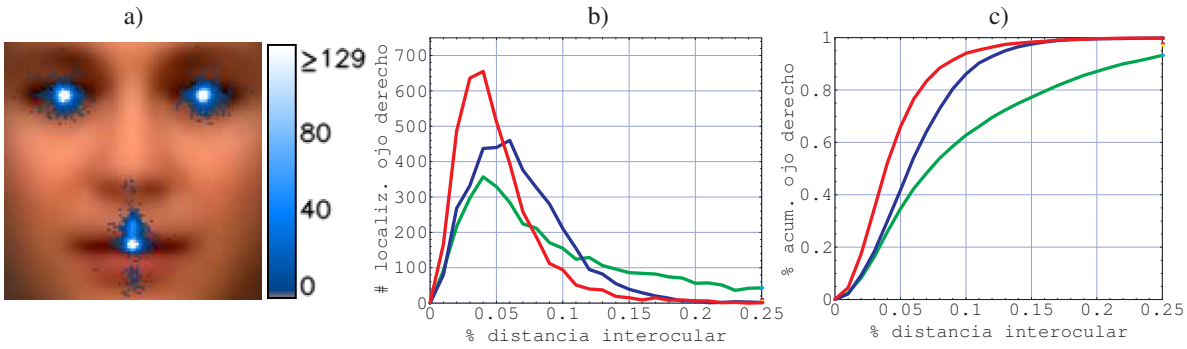


Figura 4.4: *Diferentes representaciones gráficas de los resultados de un localizador. a) Gráfica de densidad de localizaciones (frecuencia de puntos situados en cada parte de una cara estándar). b) Curva de distribución de distancias; en este caso para el ojo derecho, del método basado en proyecciones (en rojo) y otras dos técnicas. c) Curva acumulada de distancias; corresponde a los mismos datos de la gráfica b).*

un ejemplo concreto. Como ya hemos visto, el error (eje horizontal) se suele expresar en proporción a la distancia interocular. Suponiendo una separación típica de 70 mm, un valor de distancia de 0,1 corresponde a unos 7 mm en coordenadas del universo 3D.

- **Curva acumulada distancias.** Se obtiene tomando la integral de la anterior curva; es decir, el punto r de esta curva acumula todos los casos en los cuales el error del componente es menor o igual que r. La figura 4.4c) es una gráfica de este tipo. En esta representación, el número de casos (eje vertical) se expresa en relación al total, por lo que todas las curvas tienden a valor 1. Así, por ejemplo, viendo la curva roja de la figura 4.4c) podemos deducir que casi el 95 % de las localizaciones de ojo derecho tienen un error entre 0 y 7 mm para el algoritmo basado en proyecciones.

En definitiva, son muchas las métricas que se deben contrastar para realizar una evaluación completa y rigurosa de los diferentes métodos. Y, como hemos mencionado, casi todas ellas están influidas por el error de la medición manual. Cuando describamos los experimentos, en la sección 4.4, concretaremos el conjunto de medidas usadas.

4.2. El estado del arte en localización de componentes faciales

El número de trabajos que abordan de forma específica la localización de los componentes faciales es relativamente inferior al de los que tratan la detección o la localización de caras[3]. Casi siempre, porque el segundo problema conlleva explícitamente la resolución del primero. Esto ocurre típicamente en los métodos ascendentes (*bottom-up*) que localizan primero los componentes individualmente, para después agruparlos en caras según restricciones geométricas predefinidas [204].

[3]Recordemos que se ha distinguido entre *localización de caras* (detectar la única cara existente en una imagen) y *localización de componentes faciales* (encontrar los elementos constituyentes de una cara ya detectada).

Así pues, el paso de localización y extracción de componentes puede ser previo o posterior a la detección de la cara, según se use una estrategia ascendente o descendente de procesamiento. En el segundo caso, la búsqueda estará limita a ciertas regiones predefinidas sobre la cara detectada. El primero, lógicamente, se ve sujeto a una mayor incertidumbre, ya que no se conoce de antemano la posición global del rostro. No obstante, ambos casos parten de la hipótesis de localización: existe una única cara que ocupa la mayor parte de las imágenes.

Combinando las clasificaciones propuestas por Zhao y otros [212], y por Hjelmas y Low [81], así como las sugeridas en [100, 204], podemos distinguir tres grandes tipos de acercamientos a la localización de componentes faciales:

- **Basados en características de bajo nivel.** Al igual que los métodos de detección basados en invariantes, consisten en aprovechar características de bajo nivel que aparecen asociadas típicamente a las cejas, ojos, boca, y otras partes del rostro. Por ejemplo, se han usado: niveles de gris [198, 51, 193], bordes [167, 106, 211, 58], textura [79, 40], color [84, 206], simetría [148, 157], proyecciones [199, 169, 132, 59, 50, 213], etc.

- **Basados en análisis estructural.** Estas técnicas utilizan modelos matemáticos que combinan posición y forma de los elementos faciales. El ajuste de los modelos ofrece información detallada sobre la posición de los componentes o de la cara en sí. Por ejemplo, dentro de este grupo se encuentran: *snakes* [208, 73, 207], patrones deformables [210, 104], y todas las variedades de modelos de distribución de puntos (PDM) [33], modelos de forma activa (ASM) [105, 33, 34], de apariencia activa (AAM) [32, 97, 171, 121], y modelos 3D [14, 44].

- **Métodos holísticos.** El fundamento de estos sistemas consiste en plantear la búsqueda de componentes dentro de la cara como una analogía a buscar caras dentro de una imagen. Así, si la detección facial se ha resuelto con redes neuronales [156, 152, 147], AdaBoost [38, 128, 197], autocaras [137, 125], o con SVM [86, 175], la localización de componentes usará los mismos clasificadores pero entrenados específicamente para detectar ojos, narices y bocas.

Es posible encontrar también muchos trabajos que tratan la detección de caras, pero omiten cualquier referencia dirigida a concretar dónde está situada cada parte de la misma. La suposición implícita es que los ojos, la nariz y la boca ocuparán –de manera aproximada– posiciones prefijadas sobre el rectángulo o la elipse devuelta por el detector. La medida en la que esto sea más o menos correcto es otro criterio de la bondad de un detector. Pero normalmente –como veremos en los experimentos– la precisión de los detectores por sí solos no suele ser excesivamente elevada.

La taxonomía propuesta no es necesariamente disjunta, y algunos métodos pueden hacer uso de los principios aplicados en diferentes categorías. Por ejemplo, un modelo de forma activa puede ser inicializado mediante una búsqueda de propiedades de bajo nivel. En cualquier

caso, creemos que estos tres grupos representan fielmente los fundamentos que han sido aplicados hasta la fecha. Vamos a profundizar en cada uno de ellos, refiriéndonos a los trabajos más relevantes que los constituyen.

4.2.1. Métodos basados en características de bajo nivel

Los métodos de localización de componentes basados en características de bajo nivel tratan de seleccionar propiedades invariantes de los elementos faciales, esto es, que se mantienen bajo diferentes condiciones de iluminación, pose, expresión y forma de la cara. Existen muchas propiedades que han sido aprovechadas hasta la fecha. Por lo general, algunas funcionan mejor en determinadas condiciones que otras. Pero, en última instancia, la búsqueda de un invariante universal y absolutamente robusto parece condenada al fracaso.

Intensidad de gris

Es un hecho evidente que muchos de los elementos faciales –cejas, ojos, fosas nasales, boca– aparecen *normalmente* con un tono más oscuro que el resto de la cara [81]; lógicamente, tampoco esta hipótesis se puede garantizar universalmente. A pesar de ello, son muchos los trabajos basados en una simple búsqueda de mínimos locales sobre los niveles de gris [198, 51, 199]. El funcionamiento típico de estos métodos incluye la aplicación de una serie de operaciones elementales sobre las imágenes. Por orden, el esquema suele ser: (1) ecualización o estiramiento lineal del histograma, para conseguir invarianza frente a la intensidad global de las imágenes; (2) operaciones de morfología matemática, para mejorar la conectividad de las regiones oscuras; (3) umbralización fija o adaptativa del nivel de gris; y (4) agrupación de componentes conexos. Por ejemplo, en [51], el umbral se obtiene seleccionando un mínimo local en el histograma de intensidad de la región de cara.

Un ejemplo de aplicación de este enfoque es el sistema visual del robot descrito en [193], que realiza una simple umbralización de zonas oscuras, utilizada como un método de comprobación de caras candidatas, detectadas por color de piel. También los máximos locales pueden ser aprovechados para detectar zonas claras de la cara. Por ejemplo, en [83] se define un máximo como un píxel claro rodeado de 8 píxeles más oscuros. Este invariante es utilizado para localizar partes de la cara como la punta de la nariz.

Merece la pena destacar el caso del procesamiento de caras utilizando frecuencias *próximas al infrarrojo*. Algunos trabajos [211, 214], aprovechan el efecto de "pupila brillante" –similar al de "ojos rojos" en imágenes normales– para encontrar posiciones candidatas de ojos por simple diferencia de niveles de gris. En concreto, en [211] Zhao y Grigat se basan en la diferencia entre la imagen original y el resultado de un operador morfológico de apertura, que destaca los puntos claros. Posteriormente, se aplican otros pasos de selección de los candidatos.

Operadores de bordes y simetría

Ya vimos en el apartado 3.2.2 del capítulo anterior (ver la página 91) la extensa utilización de los operadores de bordes en las técnicas de detección ascendentes, de manera que no volveremos a insistir en los aspectos ya discutidos. En estos métodos, la obtención de componentes candidatos va seguida de un análisis y agrupación de los mismos en estructuras coherentes con un modelo de cara. Las operaciones más habituales en este grupo incluyen el operador de bordes de Canny [167, 106, 211], la búsqueda de componentes conexos [72], la transformada de Hough para detectar círculos [211], la descripción estadística de las relaciones geométricas de la cara [94, 117], y el análisis de constelaciones para agrupar candidatos (se puede encontrar un extenso repaso de estas técnicas en [81]).

En relación con este enfoque, en [58] propusimos un sistema de análisis de expresiones faciales partiendo de una cara ya detectada, ilustrado en la figura 4.5. El algoritmo elimina en primer lugar el exterior del rostro y la nariz –figura 4.5b)–, aplicando después un filtro de Prewitt. El resultado es umbralizado –figura 4.5c)– para obtener una imagen de bordes binaria. Los bordes generados en las posiciones de las cejas, los ojos y la boca son descritos como nubes gaussianas, acumulando los puntos en las regiones esperadas a priori para cada componente –figura 4.5d)–. Finalmente, se calculan parámetros de forma y distancia entre estas nubes para clasificar la expresión en una de varias categorías predefinidas.

Figura 4.5: *Localización y descripción de componentes faciales mediante agrupación de bordes en formas gaussianas [58]. a) Imagen original con la cara detectada. b) Cara segmentada con una máscara de forma predefinida. c) Resultado del filtro de Prewitt umbralizado, sobre la cara segmentada. d) Agrupación de los bordes en nubes de distribución gaussiana.*

Una alternativa a los operadores de bordes son los filtros de simetría local. Estas operaciones intentan aprovechar el hecho de que no sólo la cara es simétrica, sino que también sus componentes (ojos, nariz y boca) presentan simetría de forma natural [81]. Un ejemplo es la propuesta de Reisfeld y otros [148], que definen un operador genérico de simetría basado en una relación de similitud local en los valores de magnitud del gradiente.

En general, uno de los problemas de los métodos basados en operadores de bordes es que resultan muy sensibles a la aparición de diversos elementos faciales: gafas, barba, bigote, arrugas, sombras, etc. Además, la localización exclusivamente por nivel de bordes es muy imprecisa, por lo que muchas veces es sólo un paso intermedio de un proceso mayor.

Color de los componentes

Existen infinidad de trabajos que hacen uso de la información de color para resolver o ayudar en el problema de detección de caras (repasar la página 93 y sucesivas). También son muchos los que lo aprovechan en el seguimiento en secuencias de vídeo. Sin embargo, en casi todos los casos se habla del color de la piel; es sorprendente el escaso uso que se ha hecho del color de los componentes faciales, con la salvedad del color de los labios. Por ejemplo, en [206] se define la llamada *transformación de color de labios*, que destaca las diferencias de matiz entre el color de la piel y el de los labios, consiguiendo una distinción bastante efectiva. En [207], los autores aplican esta operación en combinación con la técnica de *contornos activos* (que detallaremos más adelante, en el apartado 4.2.2), para obtener una descripción del borde exterior de los labios.

Otro de los trabajos más significativos es el de Hsu y otros [84]. En el método que proponen se utiliza una variante del espacio de color YCrCb, con una compensación previa del tono, similar a un balance de blancos. La forma de encontrar el color típico de ojos y boca se deduce *ad hoc*, a través de una serie de observaciones experimentales. En concreto, el mapa de color de ojos viene dado por:

$$ColorOjos = 1/3\left(Cb^2 + (1 - Cr)^2 + Cb/Cr\right) \tag{4.6}$$

indicando un valor alto una mayor verosimilitud de color de ojo. Teniendo en cuenta que los ojos son normalmente más oscuros que el resto de la cara, la medida de color de ojos se calcula como: $ColorOjos \cdot (1 - Y')$, donde Y' es la imagen de intensidad tras aplicarle operadores morfológicos (con el fin de mejorar las zonas oscuras). Por su parte, el mapa de color de boca tiene en cuenta que ésta tiene mayores tonos de rojo; se obtiene con la fórmula:

$$ColorBoca = Cr^2\left(Cr^2 - \alpha Cb/Cr\right)^2 \tag{4.7}$$

Donde α es el promedio de Cr^2 dividido entre el promedio de Cb/Cr. Como ya hemos mencionado, no se proporciona una justificación teórica para estas fórmulas. Pero los resultados indican altos ratios de detección y una buena precisión en las posiciones de los ojos. La localización de la boca es más problemática, siendo situada muchas veces en la nariz.

Integrales proyectivas

Hasta la fecha, la utilización de integrales proyectivas en los problemas de localización de componentes faciales se suele caracterizar por dos aspectos: (1) el uso que se hace de las proyecciones es fundamentalmente heurístico; y (2) el análisis de las proyecciones es sólo un paso más dentro de un proceso mayor. Así, son muchos los trabajos que siguen –con más o menos variaciones– el esquema típico [169, 170, 199, 62, 66]: proyectar las regiones de ojos y boca; buscar mínimos locales; y devolver la posición de los mínimos en X y en Y.

El primer aspecto de los señalados se deriva de la observación de que los ojos y la boca

aparecen normalmente con un tono más oscuro que el resto de la cara. El segundo se debe a la opinión subyacente de que las proyecciones por sí solas no son suficientes para resolver el problema; aunque existen algunas excepciones, como veremos.

Vamos a profundizar en algunas de las propuestas más interesantes en el ámbito de la localización facial usando integrales proyectivas, intentando seleccionar aquellas que aportan una mayor originalidad.

- **Análisis min-max y lógica difusa.** Uno de los modos más habituales de usar las proyecciones es el método sugerido por Sobottka y Pitas [169, 170]. Como ya mencionamos en la página 3.2.2, el proceso de localización que plantean hace uso de múltiples características: color, forma, morfología. La orientación del rostro se asocia con la dirección principal del componente conexo detectado por color de piel.

  Una vez segmentada la región de cara, se obtiene la proyección vertical, $PV(y)$, y se le aplica un suavizado de 3 puntos. Luego se calculan los mínimos locales más significativos, y para cada uno de ellos se obtiene la proyección horizontal (en una región de 3 píxeles de alto), $PH_y(x)$; esta proyección es también suavizada, y se localizan sus mínimos y máximos locales. Con esta información, se obtienen posiciones candidatas para los ojos y la boca, que son aquellas que cumplen una serie de criterios *ad hoc*:

  - **Ojos:** debe aparecer un mínimo local en la mitad superior de $PV(y)$; en $PH_y(x)$ debe haber dos mínimos locales significativos; estos dos mínimos deben tener parecido valor de gris; debe existir un máximo entre ellos; la distancia entre los mínimos está en una cierta proporción respecto del tamaño de la cara.
  - **Boca:** debe aparecer un mínimo local en la mitad inferior de $PV(y)$; deben haber dos máximos significativos en $PH_y(x)$ (las esquinas de la boca) y un mínimo entre ellos; la distancia entre los máximos está en una cierta proporción respecto del tamaño de la cara.

  Los criterios cuantificables se evalúan usando *lógica difusa*. Con los candidatos resultantes, se define un proceso de agrupación (*clustering*) y selección heurística del grupo más prometedor.

  En un vídeo de prueba con 150 imágenes (estilo programa de noticias), informan de un 81 % de detección y localización correcta de los ojos, y un 64 % para la boca. No se detalla la manera de decidir cuándo las posiciones son correctas, sino que parece haberse hecho manualmente.

- **Proyección de la imagen de bordes.** Posiblemente, la aplicación de proyecciones sobre las imágenes de bordes ha sido tan popular como la proyección de los valores de intensidad. De hecho, podemos encontrar en este grupo algunos de los trabajos pioneros en el procesamiento de caras, como [93, 190, 18]. Un ejemplo más reciente es la propuesta

de Yang y otros [199], que presentan un método de localización y seguimiento utilizando color, intensidad y proyecciones. En la figura 4.6 se puede ver un ejemplo donde este tipo de proyección puede resultar más adecuado. El caso de la figura 4.6c) parece ser el más interesante, pero la obtención del umbral óptimo tampoco resulta trivial.

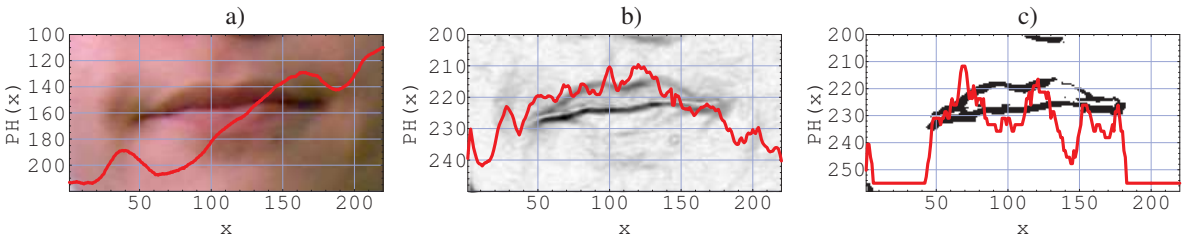


Figura 4.6: *Proyecciones de la intensidad y de las imágenes de bordes de una boca. a) Integral proyectiva horizontal del canal rojo de la imagen. b) Proyección horizontal de la imagen de bordes (operador de Sobel). c) Proyección horizontal de la imagen de bordes binarizada.*

En la propuesta de Yang y otros, en primer lugar se encuentran las pupilas mediante umbralización iterativa de la parte superior de la cabeza, detectada por color. Una vez hecho esto, se usan proyecciones para determinar las esquinas de los labios de la siguiente manera:

1. Usando las posiciones de los ojos, se extrae una región amplia donde se espera que esté situada la boca.
2. Sobre la región extraída, se calcula la proyección vertical de la imagen de intensidad. La posición de la boca se sitúa en el mínimo local de esta señal.
3. Se aplica un operador de bordes vertical (no se especifica cuál) sobre una región de unos 15 píxeles de alta en torno al resultado del paso anterior.
4. La imagen de bordes de proyecta horizontalmente. Puesto que los labios generan típicamente bordes abruptos y las mejillas no –ver el ejemplo de la figura 4.6c)–, las esquinas de los labios son localizadas mediante una simple umbralización de la proyección calculada.

Los autores presentan varias aplicaciones del sistema propuesto, como un estimador de pose, un seguidor del punto de mira, y un sistema de lectura de labios. Sin embargo, no hay una evaluación explícita de los resultados de la localización.

- **Eliminación de umbrales prefijados.** Pahor y Carrato [132], investigan también el problema de localizar las esquinas de los labios con integrales proyectivas. Como en el caso de Yang y otros [199], las proyecciones se aplican sobre las imágenes de bordes verticales, en este caso obtenidas con el operador de Prewitt. La proyección horizontal, $PH(x)$, ayuda a decidir la posición en X de las esquinas.

  La principal motivación de esta propuesta consiste en evitar el uso de umbrales, presentes en [199, 169, 170, 59] y en otros trabajos similares. Para ello, se basan en la observación de que la parte positiva del gradiente y la negativa son similares (casi simétricas)

en la zona de la boca, pero diferentes en el resto de la mitad inferior de la cara. Sean $PH_p(x)$ y $PH_n(x)$ las proyecciones horizontales de los valores positivos y negativos, respectivamente, del resultado del operador de Prewitt vertical. Se define una función que mide la diferencia entre $PH_p(x)$ y $PH_n(x)$, que llamamos $PH_d(x)$. Las esquinas se sitúan horizontalmente en las intersecciones entre $PH_p(x)$ y $PH_d(x)$.

Para determinar la posición en Y de las esquinas, se calculan dos proyecciones verticales de la imagen de gradiente: una para una franja asociada a la esquina izquierda y otra para la derecha. El resultado final en Y se sitúa en los picos máximos de las señales. El orden aquí es el contrario a los dos métodos anteriores, donde se aplicaba primero la proyección vertical y luego la horizontal.

Este sistema exhibe unos interesantes resultados sobre algunas secuencias de vídeo, donde la región inicial de boca es situada manualmente. Realmente, los ejemplos sobre los que se aplica no presentan problemas de sombras, pose o elementos faciales. No obstante, lo interesante del método es eliminar el uso de umbrales fijados a priori.

- **Función de proyección de la varianza.** Hasta ahora, hemos visto que la idea de las integrales proyectivas es calcular la media de cada fila o columna de píxeles. Feng y Yuen [50], proponen otra proyección alternativa, donde el valor asociado no es la media sino la varianza de la fila o columna correspondiente. De esta forma, la transformación deja de ser lineal. Sin embargo, justifican que el resultado de esta operación, que denominan *proyección de la varianza*, puede aportar más información en muchos casos que la proyección de la intensidad.

  Sea i una imagen en escala de grises, la *proyección vertical de la varianza*, $VV(y)$, en el rectángulo $(x_1, y_1) \leftrightarrow (x_2, y_2)$, se define como:

$$VV(y) = \frac{1}{x_2 - x_1} \sum_{x=x_1}^{x_2} \left(i(x, y) - PV(y)\right)^2 \tag{4.8}$$

  De manera análoga se define la *proyección horizontal de la varianza*, $VH(x)$. En la figura 4.7 se pueden ver dos ejemplos de estas operaciones aplicadas sobre un extracto de un ojo. En este caso, los resultados de ambos métodos adoptan formas bastante parecidas.

  Usando estos nuevos tipos de proyecciones, diseñan un algoritmo para la localización de puntos característicos (esquinas de los ojos, posición de la pupila, etc.) en imágenes de ojos ya segmentados. El proceso consta de los siguientes pasos:

  1. Calcular la proyección de varianza vertical, $VV(y)$, y horizontal, $VH(x)$, de la imagen del ojo –como las mostradas en las figuras 4.7b,c)–.
  2. Obtener las derivadas, en valor absoluto, de ambas señales:
     $dVV(y) = |VV(y) - VV(y-1)|$; $dVH(x) = |VH(x) - VH(x-1)|$.

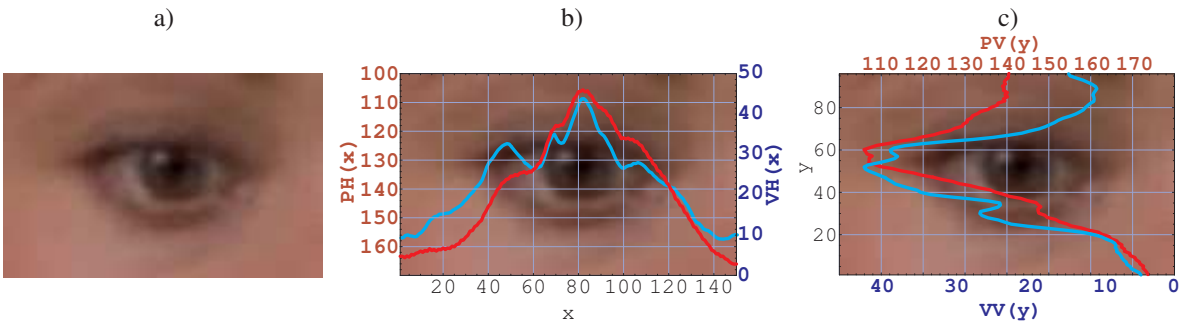


Figura 4.7: *Integrales proyectivas (en rojo) y proyecciones de la varianza (en azul) de un ojo (observar las distintas escalas para ambas señales). a) Extracto de un ojo sobre el que se aplican las proyecciones. b) Proyecciones horizontales. c) Proyecciones verticales.*

3. Extraer los máximos locales más significativos de dVV y dVH. Existe un modelo de cara predefinido, según el cual deben aparecer 2 picos en dVV y 4 en dVH.
4. Los límites verticales y horizontales del ojo y de la pupila se extraen usando los picos resultantes.

La introducción de este mecanismo de proyección alternativo supone una interesante novedad, aunque no está muy clara su robustez frente a sombras y a otros problemas habituales. Algunos aspectos quedan indefinidos en [50], y no existe una evaluación práctica del mecanismo propuesto.

- **Función de proyección generalizada.** Más recientemente, Zhou y Geng [213], sugieren combinar los conceptos de integral proyectiva (PV, PH), y proyección de varianza (VV, VH), creando las *funciones de proyección generalizadas* (GV, GH). Concretamente, se definen como una media ponderada de las dos anteriores:

$$GV(y) = \alpha VV(y) + (1-\alpha)PV(y) \tag{4.9}$$

$$GH(x) = \alpha VH(x) + (1-\alpha)PH(x)$$

Como en el caso de Feng y Yuen [50], la localización de los ojos se realiza buscando picos en las derivadas de GV y GH, que determinan la extensión en X y en Y de la pupila. En la experimentación, se utilizan tres bases de caras: BioID, JAFFE y NJUFace. La máxima distancia para declarar correcta una localización se fija en el 25 % de la distancia interocular. Los ratios de localización alcanzados son del 94,8 %, 97,2 % y 95,8 %, respectivamente.

Realmente, la mejora del método generalizado respecto de las integrales proyectivas originales es poco significativa (en algunos casos, menos de 1 punto porcentual). No obstante, los autores se fijan en el hecho de que las integrales proyectivas funcionan mejor con los occidentales, mientras que la proyección de varianza es más adecuada para los orientales. Este hecho se justifica en motivos antropológicos [213]:

> "En particular, las caras de los occidentales suelen tener las narices más grandes y las cuencas oculares más profundas, por lo que se generan muchas sombras en el rostro; las caras de los orientales suelen tener menores narices y las cuencas de los ojos menos profundas, por lo que hay pocas sombras en el rostro. [...]"

Existen otros muchos trabajos de procesamiento de caras que hacen uso de las proyecciones [93, 101, 59, 62, 24, 190, 18, 66]; algunos de ellos ya los hemos comentado en el capítulo 3. En este punto nos hemos centrado exclusivamente en la aportación específica al problema de localización de componentes faciales.

4.2.2. Métodos basados en análisis estructural

Los métodos de localización estructurales persiguen no sólo encontrar la posición de los elementos faciales, sino también describir su contorno o su forma. En su mayoría se trata de técnicas basadas en modelos, que pueden ser genéricos o específicos del dominio de las caras. De esta manera, el proceso de localización se puede entender como una búsqueda del mejor ajuste del modelo a la instancia actual, a través de la optimización de una función objetivo. Podemos distinguir dos subcategorías: las técnicas orientadas al análisis de elementos individuales (*snakes*, patrones deformables) [208, 73, 210, 104], y las que tratan la cara de manera global (modelos de forma activa y apariencia activa) [105, 33, 34, 32, 97, 171].

Snakes y patrones deformables

Los *snakes*, o *contornos activos*, son una técnica genérica de modelado de contornos, que ha sido ampliamente utilizada en la obtención del perímetro de la cara [208, 73, 51, 169]. Un *snake* es una curva cerrada, compuesta por trozos de curvas, que se sitúa inicialmente en una posición de la imagen y se adapta progresivamente al contenido de la misma. Existe un término de energía, E_{snake}, que controla la manera en la que el contorno se ajusta a la imagen; normalmente tiene la forma:

$$E_{snake} = E_{interna} + E_{externa} \tag{4.10}$$

El término de energía interna, $E_{interna}$, controla la evolución natural de la curva. La elección más habitual [208, 73], consiste en definirla proporcionalmente a la distancia entre los puntos de control del *snake*. Esto le otorga un comportamiento *elástico*, que hace que tome formas compactas. Por otro lado, la energía externa suele estar relacionada con el gradiente de la imagen [73], haciendo que la curva se sitúe próxima a los bordes más destacados. Algunos trabajos añaden también información de color de piel en el término $E_{externa}$ [208, 169, 207].

El ajuste del *snake* a la imagen actual consiste en un proceso iterativo de minimización de la energía total del sistema. Un inconveniente de este método es que la curva puede quedar

atascada en mínimos locales, produciendo un mal resultado. Además, la aplicación sobre componentes concretos, ojos, cejas y boca [51], presenta otros problemas: la posible falta de definición en el gradiente de la imagen, y la dificultad de los *snakes* para modelar contornos no convexos.

Para paliar estos obstáculos, Yuille y otros [210], definen un mecanismo de modelado de *patrones deformables*, basado también en un comportamiento elástico. Para mejorar la extracción de los ojos, incorporan información global sobre sus características más salientes: picos, valles, bordes e intensidad. La definición de la energía incluye los términos correspondientes a estos elementos:

$$E_{patron} = E_{picos} + E_{valles} + E_{bordes} + E_{intensidad} + E_{interna} \tag{4.11}$$

Como en el caso de los *snakes*, el ajuste del patrón a una imagen se realiza mediante un proceso de minimización de la energía, basado en una técnica de *gradiente descendente*. Una limitación que no consigue resolver este acercamiento es la propiedad de localidad; el patrón puede acabar situado, por ejemplo, en las cejas si no es inicializado cerca de la posición correcta. También tiene problemas de eficiencia computacional. La aplicación de una transformada de Hough circular [29], o la detección de las esquinas de los ojos [104], han sido usados como métodos para conseguir una localización inicial del modelo, intentando así reducir ambos problemas.

Modelos de distribución de puntos

Los *modelos de distribución de puntos* (PDM) son la base de la cual se derivan desarrollos posteriores como los modelos de forma activa [105, 33, 34], y apariencia activa [32, 171, 121]. Fueron propuestos por Cootes y Taylor [33], como una manera compacta de describir los posibles modos de variación de un conjunto de puntos asociados a distintas partes de la cara. En este acercamiento, la localización facial ocurre de forma global y no independientemente para cada componente.

La clave del método es la aplicación de análisis de componentes principales (PCA) sobre las variaciones observadas de una serie de puntos característicos. El proceso parte de un conjunto de puntos (normalmente entre 60 y 200) etiquetados manualmente en posiciones determinadas de una serie de imágenes, como se puede ver en la figura 4.8a).

Sea $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, un conjunto de n puntos situados sobre una imagen, donde los x_i son puntos 2D ($\mathbf{x}$ es, por lo tanto, un vector de $2n$ dimensiones). En primer lugar, se calcula la posición media de los puntos: $\overline{\mathbf{x}}$. Sea $\mathbf{X}$ una matriz de $2n \times k$ que almacena los k vectores de entrenamiento, restándoles la media: $\mathbf{x} - \overline{\mathbf{x}}$. Los modos de variación del PDM vienen dados simplemente por los autovectores de: $\mathbf{X}\mathbf{X}^{\mathrm{T}}$.

Supongamos que $\mathbf{P_s}$ es la matriz de los r autovectores con mayor autovalor asociado de $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. El modelo de distribución de puntos permite generar distintas combinaciones de la nube de puntos aplicando la fórmula generativa:

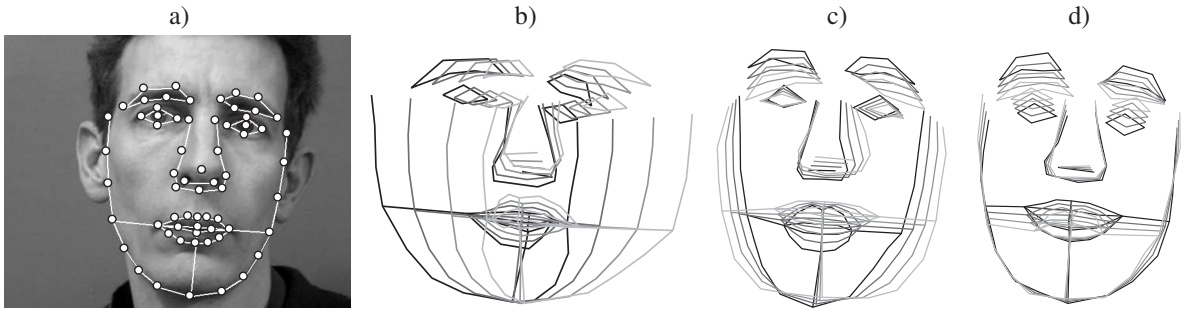


Figura 4.8: *Modos de variación en un modelo de distribución de puntos. a) Ejemplo de cara etiquetada manualmente en 66 posiciones. b,c,d) Los tres primeros modos de variación del modelo creado. Información extraída de:* `http://www.isbe.man.ac.uk/~bim/`

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P_s}\mathbf{b_s} \tag{4.12}$$

Donde $\mathbf{b_s}$ es un vector de r parámetros del modelo deformable. En la figura 4.8b,c,d) se pueden ver varios de estos ejemplos creados con la ecuación 4.12, manteniendo a 0 todos los valores de $\mathbf{b_s}$, excepto el primero, el segundo y el tercero, respectivamente.

Para permitir transformaciones de posición, escala y rotación, se añade una función $S_{\mathbf{d}}$, siendo $\mathbf{d}$ los parámetros de una transformación similar. De esta manera, los posibles puntos generados son: $S_{\mathbf{d}}(\overline{\mathbf{x}} + \mathbf{P_s}\mathbf{b_s})$.

En definitiva, el modelo PDM consta de dos tipos de parámetros: los de transformación similar, $\mathbf{d}$, y los de forma, $\mathbf{b_s}$. El ajuste de estos parámetros a un nuevo conjunto de puntos, $\mathbf{x_{nuevo}}$, se puede llevar a cabo de forma más o menos directa: (1) resolver los parámetros $\mathbf{d}$ según una modificación global de los puntos; (2) restar $\overline{\mathbf{x}}$ al resultado del paso anterior; y (3) proyectar el resultado en la base $\mathbf{P_s}$ para obtener $\mathbf{b_s}$.

Modelos de forma activa

El modelo PDM es incompleto en sí mismo, ya que para una imagen nueva no se conoce de antemano la situación de los puntos característicos, que hemos denotado por $\mathbf{x_{nuevo}}$. Los *modelos de forma activa* (ASM) [105, 33, 34, 108], definen una manera de ajustar los parámetros de posición y forma del PDM, $\mathbf{d}$ y $\mathbf{b_s}$, añadiendo información a los puntos del modelo.

En concreto, en los modelos ASM cada punto tiene una descripción del perfil de intensidades en una dirección perpendicular al contorno en ese punto. Realmente, para reducir el efecto de la intensidad global [108], se utiliza la derivada de la intensidad. El modelo asociado a cada punto es del tipo media/covarianzas. De esta manera, la descripción de ASM contiene: el número de puntos característicos, n; la posición media de cada uno, $\overline{\mathbf{x}}$; las formas de variación de los puntos, $\mathbf{P_s}$; y los modelos de intensidades en cada punto, $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n$.

En el ajuste del modelo a una imagen actual, se inicializa el ASM en una posición próxima a la cara. Entonces tiene lugar un proceso iterativo de ajuste. Para cada punto del modelo, i, se obtiene el perfil de intensidades actual, $\mathbf{h}_i$ (de la misma forma que se llevó a cabo en el

entrenamiento). Comparando $\mathbf{h}_i$ con $\mathbf{g}_i$, se calcula la nueva posición óptima de ese punto. De esta forma deducimos el conjunto de puntos, $\mathbf{x_{nuevo}}$, que nos permite resolver los parámetros, $\mathbf{d}$ y $\mathbf{b_s}$, tal y como hemos explicado en el punto anterior.

Los modelos de forma activa tienen interesantes ventajas frente a otros enfoques de localización de componentes. La más importante es que el método aprovecha a la vez forma global y propiedades locales. Además, el resultado ofrece una descripción muy detallada de las posiciones de interés (tantas como las que se hayan definido en el entrenamiento). Por contra, los inconvenientes son los comunes en este tipo de técnicas: el entrenamiento del modelo y el ajuste a una instancia nueva son procesos muy costosos, el algoritmo puede caer en mínimos locales, y la información obtenida puede ser excesiva para muchas aplicaciones.

Modelos de apariencia activa

Kirby y Sirovich [97], son los primeros en proponer el uso de PCA (también conocido como la *transformada Karhumen-Loève*) para describir las variaciones de aspecto de las caras humanas. Esta idea es aplicada posteriormente para crear los *modelos de apariencia activa* (AAM) [32, 171, 121]. En esencia, AAM se puede ver como un extensión de ASM, incorporando información de textura al modelo de puntos. Al igual que en los PDM, los distintos modos de variación de la textura se obtienen aplicando análisis de componentes principales, en este caso sobre los niveles de intensidad de los píxeles. La filosofía subyacente a estos métodos es conocida como *interpretación a través de la síntesis* [114].

En el proceso de entrenamiento, todas las imágenes de caras son transformadas a un *modelo libre de forma*, a través de una operación geométrica que desplaza todos los puntos a las posiciones medias, $\bar{\mathbf{x}}$. Después se aplica PCA sobre estas caras normalizadas, obteniendo los modos de variación de la textura, $\mathbf{P_t}$. Igual que antes, el modelo generativo consta de una textura media, $\bar{\mathbf{t}}$, y de un conjunto de parámetros, $\mathbf{b}_t$, que dan lugar a la ecuación: $\bar{\mathbf{t}} + \mathbf{P_t}\mathbf{b}_t$.

Evidentemente, el proceso de ajuste del modelo se ve sujeto a la necesidad de resolver tres tipos de parámetros: posición, $\mathbf{d}$; forma, $\mathbf{b_s}$; y textura, $\mathbf{b}_t$. De manera resumida, se define una función residual, que mide la diferencia entre el ajuste actual y el contenido de la imagen que está siendo tratada. El algoritmo realiza un proceso de gradiente descendente, buscando la minimización del residuo.

Los modelos AAM mejoran el uso de la información que se hace en ASM. No obstante, no consiguen evitar los problemas de ineficiencia y localidad ya mencionados. Actualmente, existe un gran interés en la comunidad investigadora en resolver estos inconvenientes mejorando el mecanismo básico de AAM (ver, por ejemplo, el capítulo 3 de [108]). Otro enfoque estrechamente relacionado, aunque no lo detallaremos aquí, son los modelos deformables 3D [14, 44, 1], que trabajan explícitamente en un espacio tridimensional.

4.2.3. Métodos holísticos

Ya vimos en el capítulo 3 que los métodos de detección holísticos, o basados en apariencia, se encuentran entre los más populares en el procesamiento de caras. Estos mismos mecanismos se han utilizado también en la extracción de componentes faciales, dando lugar a localizadores basados en autoespacios [137, 125], redes neuronales [156, 152, 147], SVM [86, 175], AdaBoost [38, 128, 197], y otros [89, 76].

Puesto que los principios básicos de estos métodos ya han sido descritos extensamente en la revisión del estado del arte del capítulo 3, vamos a presentar de forma muy resumida algunas de las aportaciones más interesantes al caso de la localización.

Autoespacios asociados a los componentes faciales

Una de las primeras aplicaciones de los autoespacios en el problema de localización de partes del rostro es la debida a Pentland y otros [137], en 1994. De la misma forma que PCA había sido usado sobre la cara completa [183], se propone que sea aplicado también en la búsqueda de los ojos, la nariz y la boca. En este trabajo, el método de localización consiste simplemente en seleccionar la subregión que minimiza el error de reconstrucción en el autoespacio asociado al componente, denominado la *distancia al espacio del componente* (DFFS, *distance from feature space*, ver la página 103).

Más adelante, Moghaddam y Pentland [125], extienden la técnica básica añadiendo también la *distancia dentro del espacio del componente* (DIFS, *distance in feature space*). La principal observación es que DFFS es insuficiente para modelar por sí solo la distribución asociada a los ejemplos. De esta manera, sobre las imágenes proyectadas se calcula otra métrica de distancia (DIFS), basada en un modelo de distribución gaussiana multivariable. Ahora el proceso de localización busca la posición con mínimo valor de DFFS+DIFS.

En un experimento sobre 7000 imágenes propias –centradas en las caras de las personas y con fondo negro–, el método propuesto en [125] produce un 5 % de fallos de localización, mientras que el que usa únicamente DFFS está sobre el 10 %; en ambos casos, las bases de componentes constan de 5 autovectores. Usando 10 autovectores [137], DFFS es capaz de llegar hasta el 94 % de localización correcta. Los resultados son muy buenos comparados con una simple búsqueda de patrones medios usando suma de diferencias al cuadrado, que no llega al 75 % de localización. No se dan datos usando una medida de correlación.

La idea de los autoespacios asociados a los componentes (*auto-ojos*, *auto-bocas*, *auto-narices*), ha sido utilizada posteriormente en muchos trabajos, introduciendo numerosas variantes. Por ejemplo, en [156] se aplica PCA sobre la imagen de bordes, y redes neuronales para determinar las localizaciones resultantes. Debemos señalar que en este artículo se usan también integrales proyectivas de las imágenes de bordes (proyección horizontal de la derivada horizontal, y vertical de la derivada vertical) para una primera determinación del contorno de la cara y la posición de ojos y boca.

Localización de ojos mediante redes neuronales

Dentro de las aportaciones de Rowley al procesamiento de caras [152], se encuentra un localizador de ojos con redes neuronales. El autor ofrece un conjunto de funciones para buscar la posición más probable de un ojo derecho o izquierdo, dada una posición esperada dentro de una imagen. Desafortunadamente, no se documenta su funcionamiento en [152, 153] ni en otros trabajos posteriores. Debemos suponer que realiza un proceso parecido al detector de caras: búsqueda exhaustiva, normalización de intensidad, perceptrón multicapa, y selección del máximo. El método será incluido en los experimentos de la sección 4.4.

Otro de los trabajos precursores es el de Reinders y otros [147], que abordan también la localización de ojos con redes neuronales. Para conseguir robustez frente a los niveles de gris, se trabaja con la magnitud y la dirección del gradiente. Además, para abordar la elevada variabilidad de los ojos, las redes no se entrenan sobre todo el ojo, sino sobre las llamadas *micro-características*: esquina izquierda, derecha, párpado inferior y superior. La localización final se obtiene combinando los valores asociados a cada una de estas características, considerando el resultado de las 4 redes como medidas de probabilidad. Los autores informan de un error medio de 1,05 píxeles[4]; curiosamente, el error manual estimado es de 0,98 píxeles.

Detección de ojos con SVM

Siguiendo el esquema de detección de caras de Osuna y otros [131], Huang y otros [86] proponen la aplicación de las máquinas de vectores de soporte al problema de localización de ojos. La idea del método es bastante directa: entrenar un clasificador mediante SVM usando ejemplos de ojos y no ojos; dada una cara, aplicar el clasificador sobre las distintas regiones posibles, devolviendo la de máximo valor. En los experimentos que describen se limitan a un simple problema de clasificación ojo/no ojo. Las imágenes de entrada son de 16×16 píxeles, y están normalizadas en intensidad. El conjunto de entrenamiento consta de 186 ejemplos de ojos y otros tantos de no ojos, tomados de la base FERET [52]. Las pruebas son 200 imágenes distintas de las de entrenamiento. Los mejores resultados se alcanzan utilizando un *kernel* polinomial de grado 2, situándose el porcentaje de error en un 4 %. Es llamativo el hecho de que el mecanismo selecciona casi el 20 % de los ejemplos de entrenamiento (68 de 186) como vectores de soporte, un valor relativamente alto.

Las máquinas de vectores de soporte han sido también utilizadas en combinación con el algoritmo de aprendizaje AdaBoost. En [175], se describe un sistema que hace uso de ambas técnicas. En esencia, se aplica en primer lugar un detector de caras y después un detector de ojos sobre las regiones esperadas de los mismos; el clasificador SVM es utilizado como un método para seleccionar el par de ojos candidatos más factible (usando la posición relativa de los ojos en relación a la cara).

[4]No se ofrecen en [147] medidas del error en relación a la distancia interocular, y tampoco se pueden deducir de los datos disponibles.

Localización usando la distancia de Hausdorff

Jesorsky y otros [89], proponen un método para la localización de caras basado en detección de bordes y *distancia de Hausdorff*. Sobre las imágenes de entrada se aplica el operador de Sobel, seguido de una umbralización adaptativa de la magnitud del gradiente. Existe un modelo de bordes de la cara, compuesto por una imagen binaria de 45×47 píxeles. Este modelo se aplica en distintas posiciones y escalas, devolviendo aquella que produzca la menor distancia de Hausdorff. Esta distancia está definida sobre dos imágenes binarias, A y B. Básicamente, su valor se obtiene buscando para cada píxel activo de A (es decir, píxel de borde) el píxel activo más cercano de B; el resultado final es la suma de distancias para todos los píxeles activos de la imagen A. La imagen B es el modelo y A la instancia actual.

En [98] se propone una forma mejorada de crear el modelo de bordes, apoyándose en algoritmos genéticos. Para los experimentos utilizan las bases XM2VTS y BioID. Fijando la distancia máxima en el 25 % de la interocular, los ratios de localización alcanzados son del 94,2 % y 92,8 %, respectivamente. Para un tope del 10 %, ambos quedan próximos al 50 %.

El trabajo de Jesorsky y otros [89], marca una serie de pautas y protocolos de evaluación que serán adoptados posteriormente por otros muchos investigadores[5]: definición del criterio de localización correcta –ecuación 4.2–, representación de los datos –figuras 4.4b,c)–, utilización de las bases XM2VTS y BioID, etc. Por ejemplo, Hamouz y otros [76], proponen un método basado en filtros *wavelet*, modelos de distribución gaussiana (en el espacio de los *wavelet*), agrupación de candidatos y SVM, comparando sus resultados con los de [89]. La mejora que obtienen es muy poco significativa (incluso con peores resultados en BioID), y con un coste de unos 13 segundos por imagen.

Localización de componentes con AdaBoost

El éxito de los sistemas de detección de caras basados en la combinación de clasificadores mediante AdaBoost [188, 110], ha propiciado un creciente interés en la aplicación de esta técnica a otros problemas relacionados, como la localización de componentes faciales. Son ya varios los trabajos que utilizan un esquema similar al introducido por Viola y Jones [188], compuesto por filtros de Haar, AdaBoost y cascada de clasificadores [38, 128, 197].

- **Esquema básico: Haar+AdaBoost.** Cristinacce y Cootes [38], desarrollaron uno de los primeros métodos dentro de esta categoría, destinado a la localización de los ojos y las esquinas de la boca. Los autores llegan a la conclusión de que los elementos faciales contienen un número insuficiente de características para poder ser encontrados por sí solos. Para solucionarlo, se aprovechan las restricciones de forma de una cara normal: en primer lugar se detecta la cara, y después se aplican los detectores de ojos y esquinas de boca (con filtros de Haar y AdaBoost) en las posiciones más probables a priori. Con los candidatos resultantes, se busca una configuración consistente con un modelo de cara,

[5]Como, por ejemplo, el caso ya mencionado de [213], y otros que describiremos seguidamente.

descrito mediante distribuciones de probabilidad gaussianas.

Los clasificadores son entrenados con 995 ojos y esquinas de bocas marcados manualmente, siendo los ejemplos negativos (no-ojos y no-bocas) regiones desplazadas de las posiciones reales. El sistema se prueba sobre la base de caras XM2VTS y sobre una propia, siendo el umbral para declarar un fallo de localización el 30 % de la distancia interocular. En la primera base, el 85 % de las caras tienen una separación media máxima del 10 % de la interocular, mientras que en la segunda no llega al 70 %. En un Pentium II a 500Mhz, el tiempo de ejecución es próximo a 0,5 segundos.

- **Cascadas 2D de clasificadores.** Más recientemente, Niu y otros [128], plantean una extensión del mecanismo de cascada original, dando lugar a lo que denominan como *AdaBoost en cascadas 2D*. Con esta modificación intentan mejorar los problemas de espacio y memoria cuando el número de ejemplos es muy grande y, sobre todo, superar la dificultad de modelar clases no compactas –esto es, en las que la distribución de los ejemplos adopta formas muy variadas–. La idea de las cascadas 2D se muestra en la figura 4.9.

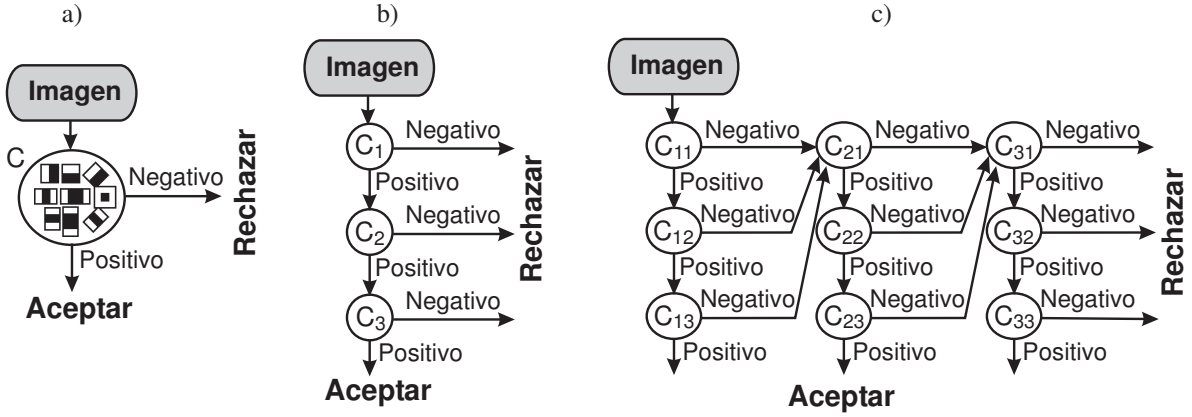


Figura 4.9: *Localización de componentes mediante cascadas 2D de clasificadores y filtros de Haar. a) Un clasificador combinado, compuesto por varios filtros de Haar, entrenado con AdaBoost. b) Cascada de clasificadores combinados [188]. c) Cascada 2D de clasificadores combinados [128].*

Cada muestra a clasificar es pasada a la primera cascada (la de más a la izquierda). Si es admitida por todos los clasificadores combinados, entonces se acepta. En caso contrario, se somete a la siguiente cascada y se repite el proceso. En [128], se define un mecanismo para realizar el entrenamiento, y también varias formas alternativas de aplicarlo al caso de la localización (usando estrategias de agrupación similares a las de [188]).

En los extensos experimentos que se detallan, se ofrecen unos resultados excelentes con varias bases de caras, XM2VTS, BioID, JAFFE y CMU PIE. Por ejemplo, en la primera el porcentaje de localización para un error máximo del 10 % se sitúa en un 97 % (12 puntos mejor que el anterior [38]), y en BioID es del 93 %. También se demuestra una gran efectividad en las condiciones de pose, iluminación y expresión de la base CMU PIE. El tiempo medio en un ordenador a 2,4GHz. está sobre los 25 ms.

- **AdaBoost multimodal, con profundidad y textura.** También en un trabajo reciente, Xue y Ding [197], describen la aplicación de las técnicas AdaBoost sobre imágenes 3D, aprovechando textura y profundidad, para la localización de los ojos y la punta de la nariz. La información 3D es convertida a imágenes de curvatura media y gaussiana. El clasificador es entrenado para usar simultáneamente la información de intensidad y ambas curvaturas. En una base pública de caras con información 3D [138], consiguen un error medio para el ojo izquierdo de 5,7 mm (∼8,1 % de la interocular), para el derecho 5,2 mm (∼7,4 %), y 3,1 mm para la nariz. El número de localizaciones dentro del 10 % de distancia máxima supera ligeramente el 70 %. No profundizaremos más en esta propuesta, ya que el manejo de información 3D cae fuera del ámbito de esta tesis.

4.3. Localización de componentes mediante proyecciones

Como acabamos de discutir en la sección 4.2, muchas técnicas de localización de componentes faciales se pueden entender como particularizaciones de los algoritmos de detección, haciendo uso de dos hechos conocidos: (1) existe una –y sólo una– cara; y (2) se parte de una localización aproximada de la misma. Aplicando la analogía entre ambos problemas, el método que proponemos y desarrollamos en esta sección se basa en un ajuste fino de los modelos de proyección vertical, MV_{cara}, y horizontal, MH_{ojos}; esto es, los mismos que ya fueron usados en el proceso de detección del capítulo 3.

Además, la localización debe resolver el problema adicional de estimar la inclinación de las caras dentro de las imágenes. Esta cuestión es tratada en una etapa inicial del algoritmo, que, como vamos a ver, hace uso de la simetría de forma robusta y se basa también en el cálculo de proyecciones. El alineamiento de señales unidimensionales será, por lo tanto, el fundamento subyacente de todos los pasos del método.

4.3.1. Esquema global del método de localización

Nuestro método de localización mediante integrales proyectivas parte del resultado de un detector de caras –el que propusimos en el capítulo 3 u otro cualquiera–, es decir, una imagen de entrada sobre la cual se ha detectado una región de cara descrita mediante un rectángulo contenedor. También son parte de la entrada los modelos de proyección, MV_{cara} y MH_{ojos}, constituyentes del modelo de caras. El algoritmo consta de tres grandes pasos, como se muestra en la figura 4.10.

- **Paso 1.** En primer lugar, el localizador estima la orientación del rostro, usando las proyecciones verticales de las zonas en las que se espera encontrar ambos ojos. El alineamiento entre las dos señales devuelve un desplazamiento, que se traduce de forma directa en un ángulo de inclinación estimado.

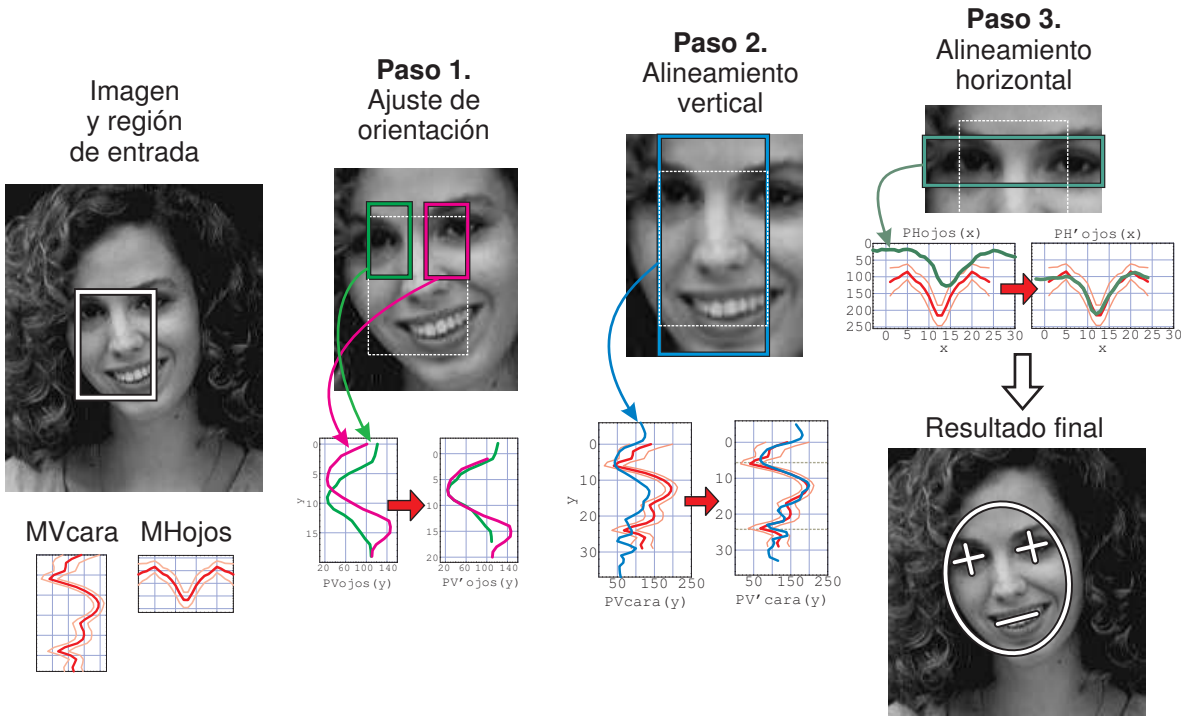


Figura 4.10: *Esquema global del localizador de componentes faciales mediante integrales proyectivas. La entrada (a la izquierda) es una imagen y el rectángulo resultante de un detector de caras. Todos los pasos se basan en obtener y alinear integrales proyectivas. Paso 1. Las proyecciones verticales de ambos ojos se alinean entre sí. Paso 2. La proyección vertical de toda la cara (incluida un área de tolerancia) se alinea con MV_{cara}. Paso 3. La proyección horizontal de los ojos se alinea con MH_{ojos}. Con los resultados de los alineamientos, se relocalizan las posiciones de forma conveniente.*

- **Paso 2.** Seguidamente, se obtiene la proyección vertical de la cara rectificada, incluido cierto margen por arriba y por abajo. Esta proyección se alinea con el modelo de MV_{cara}. Los parámetros resultantes nos indican la posición y escala vertical de los componentes faciales.

- **Paso 3.** Por último, en el tercer paso se calcula la proyección horizontal en torno a la zona de ojos, y se alinea respecto de MH_{ojos}. Como en el paso anterior, la salida del alineamiento permite ajustar la posición horizontal de la cara y sus componentes. En estos dos últimos pasos, además, se aplica el algoritmo rápido de alineamiento de proyecciones, descrito en la página 74.

La técnica que proponemos admite cierta imprecisión en las regiones de entrada, en cuanto a su posición y escala respecto a las caras existentes. De esta manera, la localización se puede ver también como un refinamiento de los resultados del detector. No obstante, esta capacidad está limitada, por un lado, por los márgenes de tolerancia usados en la ejecución del proceso, y por otro lado, por las propias características del método.

A continuación vamos a describir de manera más detallada los tres pasos del localizador, concretando aspectos de implementación y mostrando algunos ejemplos de los resultados tras cada etapa.

4.3.2. Ajuste de la orientación

En el desarrollo de la técnica de detección de caras, no se tuvieron en cuenta –al menos de forma explícita– las distintas posibles orientaciones del rostro. El método suponía caras más o menos verticales, admitiendo pequeños ángulos de inclinación. Sin embargo, ahora sí que es necesario abordar la cuestión de cómo estimar ese ángulo de manera precisa.

La orientación de la cara está relacionada con la simetría de sus dos mitades verticales. Son muchos los trabajos que han usado ya la simetría facial en problemas de detección y localización [148, 157]. No obstante, aunque exista una simetría de estructura, la simetría de apariencia puede no ser tan evidente debido a la iluminación, rotación 3D o expresión facial. Por lo tanto, usaremos esta propiedad de manera robusta frente a esos factores.

Proyecciones verticales de las zonas de ojos

Proponemos aprovechar la simetría de los ojos haciendo uso de las proyecciones verticales de las regiones que los contienen. Tales regiones de ojos vienen dadas en proporción al rectángulo contenedor de la cara, de acuerdo con la definición del modelo de cara. Por ejemplo, una posible definición del rectángulo de ojo izquierdo puede estar basada en los cuatro siguientes parámetros:

- h_{ojomin}, h_{ojomax}: posición mínima y máxima, respectivamente, en el eje Y de la región de los ojos en proporción a la altura del rectángulo de cara.

- w_{ojomin}, w_{ojomax}: posición mínima y máxima, respectivamente, en X de la región del ojo izquierdo en proporción al ancho del rectángulo de cara (simétrico para el derecho).

Recordemos que la localización de partida de la cara es aproximada, de manera que deberíamos hablar más propiamente de "regiones donde se espera encontrar los ojos". Por ello, la subregión especificada incluye ciertos márgenes adicionales. En la figura 4.11a) se interpretan estas proporciones sobre una cara media.

El cálculo de las subregiones de ojos es directo. Supongamos en adelante que el rectángulo contenedor de la cara tiene esquina superior izquierda en el punto (x_{det}, y_{det}), y es de tamaño $w_{det} \times h_{det}$. La región $ojo1$ (ojo izquierdo) es el rectángulo con esquinas:

$$\left(x_{det} + w_{ojomin} w_{det}, y_{det} + h_{ojomin} h_{det}\right) \leftrightarrow \left(x_{det} + w_{ojomax} w_{det}, y_{det} + h_{ojomax} h_{det}\right) \tag{4.13}$$

Y, de forma similar, la región $ojo2$ será el rectángulo:

$$\left(x_{det} + (1 - w_{ojomax}) w_{det}, y_{det} + h_{ojomin} h_{det}\right) \leftrightarrow \left(x_{det} + (1 - w_{ojomin}) w_{det}, y_{det} + h_{ojomax} h_{det}\right) \tag{4.14}$$

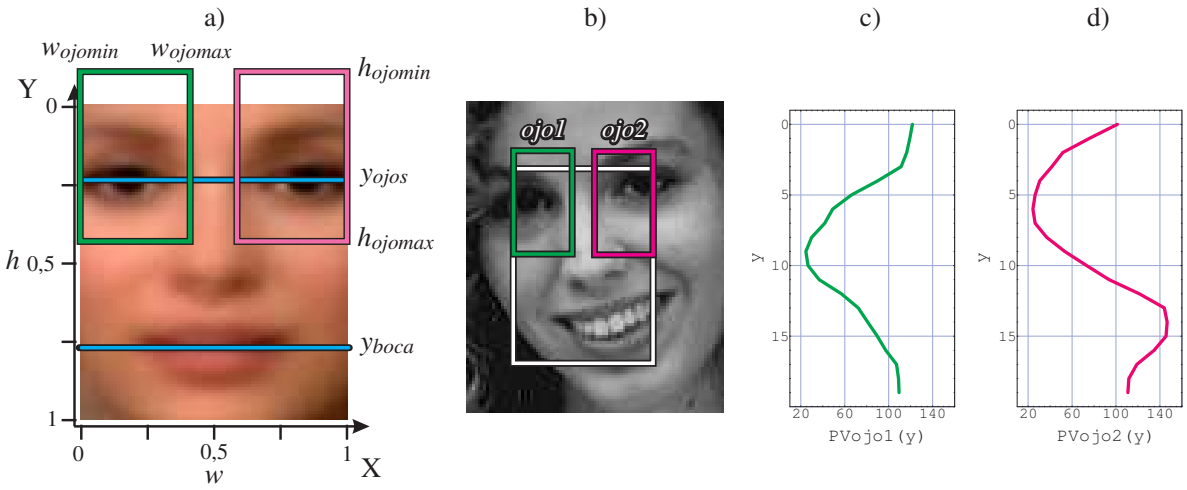


Figura 4.11: *Extracción y cálculo de las proyecciones verticales de los ojos. a) Interpretación de los parámetros que definen las regiones de ojos, sobre el modelo de cara. b) Ejemplo de regiones de ojos, $ojo1$ y $ojo2$, aplicando los parámetros sobre los resultados de la detección. c), d) Proyecciones verticales de las regiones $ojo1$ y $ojo2$, respectivamente.*

Los valores típicos usados en los experimentos[6] son del estilo: $w_{ojomin} = 0$; $w_{ojomax} = 0,4$; $h_{ojomin} = -0,15$; $h_{ojomax} = 0,4$. Estos son los valores que se han aplicado en el caso de la figura 4.11b), y en los restantes ejemplos de esta apartado.

Tomando las dos regiones de las ecuaciones 4.13 y 4.14, se calculan PV_{ojo1} y PV_{ojo2}, respectivamente, como se ilustra en la figura 4.11.

Alineamiento de PV_{ojo1} y PV_{ojo2}

Supongamos que el rostro del individuo es perfectamente simétrico y que el rectángulo contenedor está bien ajustado. En esta situación ideal, si la cara está derecha, ambas proyecciones, PV_{ojo1} y PV_{ojo2}, deben ser exactamente iguales. Es más, pequeñas rotaciones del rostro se convierten en desplazamientos relativos entre ambas señales. Conociendo ese desplazamiento y la distancia entre los ojos, la inclinación se puede obtener con un simple arcotangente. La robustez de las proyecciones proporcionará invarianza frente a caras no completamente simétricas y frente a localizaciones imprecisas.

El cálculo del *desplazamiento entre las proyecciones verticales* de ambos ojos es una simplificación del algoritmo rápido de alineamiento. Por un lado, no se necesita tener en cuenta la escala. Por otro lado, hemos comprobado que la normalización de las señales en el valor no es crítica, incluso aunque los ojos aparezcan con diferente iluminación. En consecuencia, basta con probar diferentes desplazamientos entre las señales y quedarse con el que genere mínima distancia; el proceso de recorrido se resume en el algoritmo 4.1.

Una vez con la distancia, *desp*, obtenida mediante la aplicación del algoritmo 4.1 sobre

[6]Obsérvese que estos parámetros están en función de cuál es la posición estándar del rectángulo contenedor de las caras, es decir, de los otros parámetros del modelo; en concreto, de y_{ojos}, y_{boca} y del ratio de aspecto w/h del modelo de cara utilizado.

ALINEAMIENTO DE SEÑALES MEDIANTE DESPLAZAMIENTOS
ENTRADA:
- Proyecciones a alinear: $P1$, $P2$.
- Desplazamiento máximo permitido: $desp_{max}$.

SALIDA:
- Desplazamiento para el alineamiento óptimo: $desp$.

ALGORITMO:
Inicialización:
$desp := 0$
$distMin := dist(P1, P2)$
Iteración principal:
para $i := -desp_{max}$ ***hasta*** $desp_{max}$ ***hacer***
$distAct := dist\,(P1, escalado_{d=i,e=1}(P2))$ /* *Calcular distancia para cada desplazamiento* */
si $distAct < distMin$ ***entonces*** /* *y quedarse con la mínima* */
$distMin := distAct$
$desp := i$
finsi
finpara

Algoritmo 4.1: *Alineamiento óptimo entre dos señales usando sólo desplazamientos. La función* $escalado_{de} : \mathbb{S} \to \mathbb{S}$ *está definida en la ecuación 2.16 (ver la página 44); y la función* $dist : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$ *es la de la ecuación 2.19 (ver la página 53).*

PV_{ojo1} y PV_{ojo2}, el ángulo de rotación estimado será:

$$\alpha = \arctan \frac{desp}{w_{det} \cdot p_{ojos}} \tag{4.15}$$

Donde p_{ojos} es la relación entre la distancia interocular típica y el ancho de las caras (esto es, del rectángulo contenedor); en nuestro caso, vale aproximadamente $p_{ojos} = 0,6$.

La figura 4.12 presenta un ejemplo del proceso de estimación de la inclinación de la cara. En esta imagen, el ángulo de la cara es de unos 9º, lo que se traduce en un desplazamiento entre las señales PV_{ojo1} y PV_{ojo2} de 3 puntos, para una distancia interocular de unos 20 píxeles.

Obsérvese que el parámetro $desp_{max}$ del algoritmo 4.1 indica el máximo desplazamiento analizado y, por lo tanto, limita el máximo ángulo que se puede estimar. Un valor típico es del 50 % del tamaño de las señales de entrada, lo cual –teniendo en cuenta el resto de valores normales– da un ángulo máximo de unos 25º. Usar desplazamientos mayores puede hacer que el alineamiento sea menos fiable y no garantizará necesariamente mayores ángulos localizados (y de hecho el detector difícilmente encontrará las caras, como ya vimos en los experimentos del capítulo 3).

Extracción de la cara rectificada

Usando el ángulo de inclinación estimado, se puede extraer la cara rectificada a un tamaño y posición estándar mediante una sencilla transformación afín. Supongamos que el tamaño de la cara que queremos obtener es de $w \times h$ píxeles (coincidiendo con el tamaño del modelo).

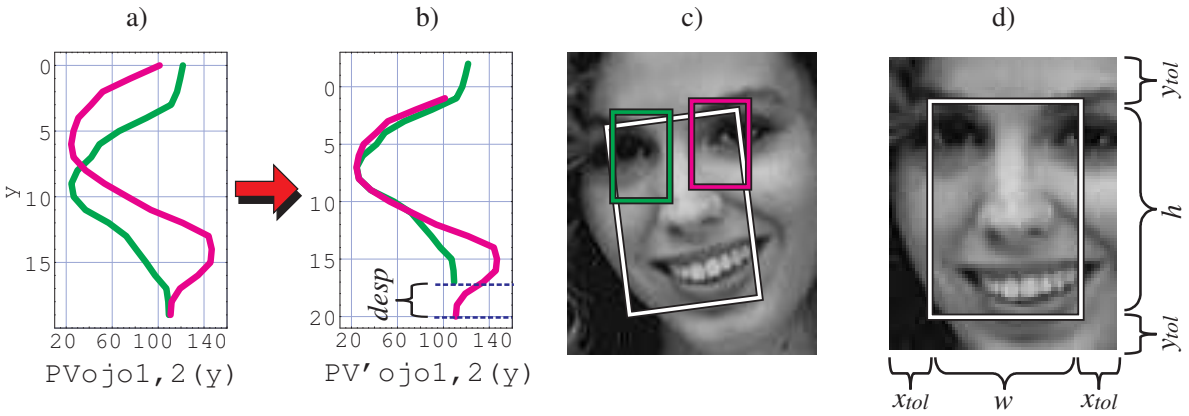


Figura 4.12: *Alineamiento de las proyecciones verticales de los ojos y rectificación. a) Proyecciones verticales de los ojos (ver la figura 4.11c,d). b) Las mismas proyecciones después del alineamiento. El valor $desp$ indica el desplazamiento para el alineamiento resultante. c) Usando $desp$ y la distancia entre las regiones $ojo1$ y $ojo2$, se calcula la inclinación de la cara. d) La cara rectificada y normalizada a un tamaño estándar, $w \times h$, más fragmentos adicionales a los cuatro lados.*

Adicionalmente, añadimos márgenes de tolerancia en los cuatro lados, de tamaños x_{tol} en X, e y_{tol} en Y, como se muestra en la figura 4.12d). Vamos a concretar los valores de la matriz de transformación afín de interés.

Dicha transformación se puede descomponer en los siguientes pasos:

1. Trasladar el centro del rectángulo de entrada, $(x_d^c, y_d^c) = (x_{det} + w_{det}/2, y_{det} + h_{det}/2)$, al píxel $(0, 0)$.
2. Rotar la imagen en ángulo $-\alpha$, según el resultado de la ecuación 4.15.
3. Escalar la imagen en X, $s_x = w/w_{det}$, y en Y, $s_y = h/h_{det}$.
4. Trasladar el píxel $(0, 0)$ al centro de la imagen resultante, $(x_r^c, y_r^c) = (w/2 + x_{tol}, h/2 + y_{tol})$.

Expresando los cuatro pasos de forma matricial, tenemos:

$$\begin{pmatrix} 1 & 0 & x_r^c \\ 0 & 1 & y_r^c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha & \operatorname{sen}\alpha & 0 \\ -\operatorname{sen}\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -x_d^c \\ 0 & 1 & -y_d^c \\ 0 & 0 & 1 \end{pmatrix} \tag{4.16}$$

En definitiva, resolviendo el producto, la matriz de transformación afín asociada a la normalización será:

$$M = \begin{pmatrix} s_x \cos\alpha & s_x \operatorname{sen}\alpha & x_r^c - x_d^c s_x \cos\alpha - y_d^c s_x \sin\alpha \\ -s_y \operatorname{sen}\alpha & s_y \cos\alpha & y_r^c + x_d^c s_y \operatorname{sen}\alpha + y_d^c s_y \cos\alpha \\ 0 & 0 & 1 \end{pmatrix} \tag{4.17}$$

También es interesante la matriz inversa, esto es, M^{-1}, como veremos enseguida.

En la figura 4.13 se pueden ver algunos ejemplos de caras extraídas tras el primer paso del algoritmo. En adelante nos referiremos a ellas con i_{ext}.

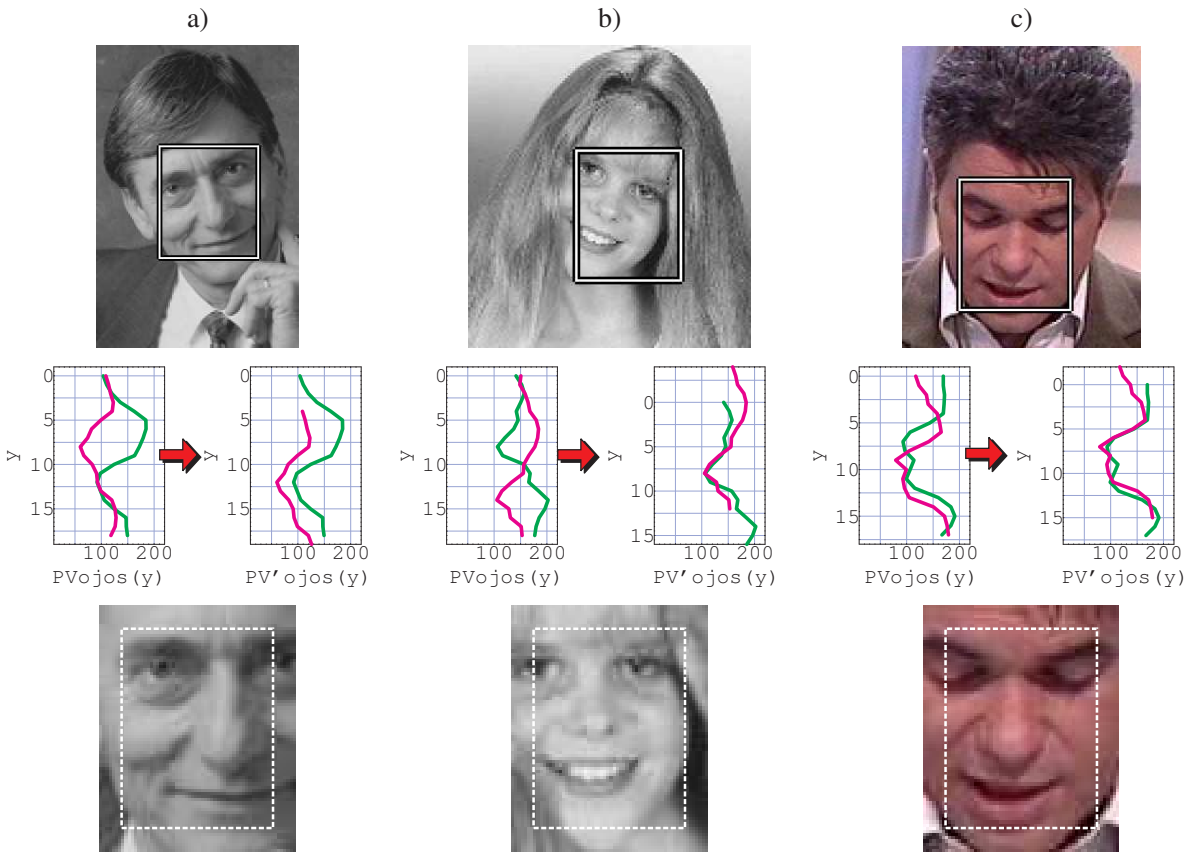


Figura 4.13: *Resultados del primer paso del algoritmo de localización de componentes. Para cada ejemplo, se muestra: arriba, extracto de la imagen y rectángulo resultante del detector; centro, proyecciones verticales de ojo izquierdo (en verde) y derecho (en magenta), antes (izquierda) y después (derecha) del alineamiento; abajo, las caras extraídas, i_{ext}, mediante una transformación afín, incluyendo un margen adicional. Extractos de: a) werb04.gif (CMU/MIT), b) tf5189a.gif (CMU/MIT), c) 617.jpg (UMU).*

4.3.3. Alineamiento vertical de la cara

El segundo paso del algoritmo de localización de componentes faciales se apoya en la proyección vertical de la cara para refinar la posición y escala en Y del rostro. Hasta la fecha, los trabajos previos que han usado integrales proyectivas han resuelto este problema con una simple búsqueda heurística de picos máximos y mínimos [93, 101, 169, 170, 59, 60, 199].

En cambio, nuestra propuesta consiste en utilizar el modelo de proyección, MV_{cara}, para encontrar el mejor ajuste a la proyección obtenida. De esta forma, se reduce la influencia de picos espurios (uno de los grandes inconvenientes de las técnicas mencionadas), se elimina la necesidad de introducir conocimiento heurístico (ya que el modelo de proyección se aprende con ejemplos), y se aprovecha toda la estructura de la cara (puesto que no se buscan los componentes por separado, sino de forma conjunta).

Cálculo de la proyección vertical de la cara

Recordemos que las caras rectificadas son extraídas, en i_{ext}, con el mismo tamaño del modelo, $w \times h$, más un ancho y un alto adicionales, x_{tol} e y_{tol}, respectivamente. El área utilizada para la proyección vertical contiene los márgenes superior e inferior, pero no los laterales.

De esta manera, se toma la región rectangular $cara = (x_{tol}, 0) \leftrightarrow (x_{tol} + w, 2y_{tol} + h)$, y se calcula su proyección vertical, PV_{cara}. En la figura 4.14 se puede ver un ejemplo de la región proyectada, y de la señal resultante en comparación con el modelo correspondiente, MV_{cara}.

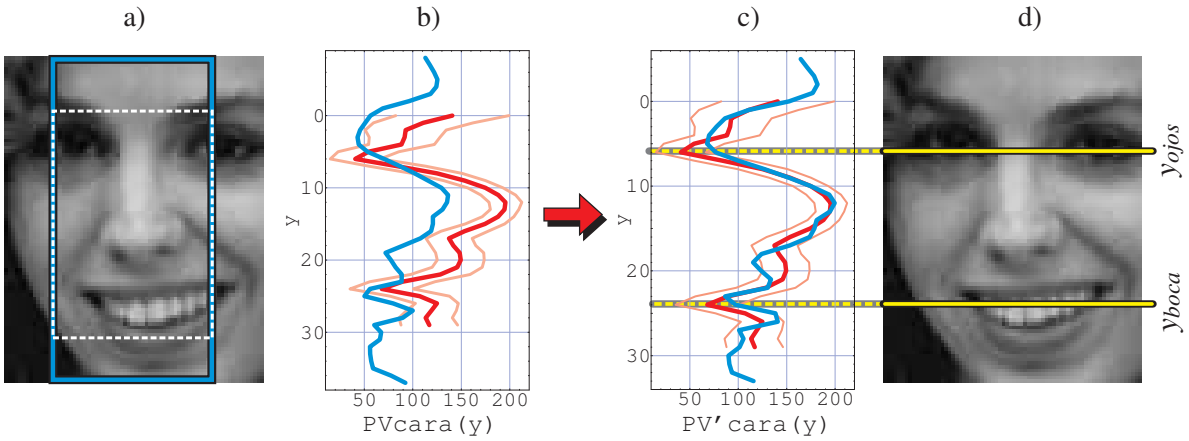


Figura 4.14: *Obtención y alineamiento de la proyección vertical de la cara, en el algoritmo de localización. a) Cara extraída del paso 1, i_{ext}, y la región usada en la proyección (señalada en azul). b) Proyección vertical de la cara, PV_{cara} (en azul), y modelo, MV_{cara} (en rojo). c) Las mismas señales tras el alineamiento. d) Con los resultados del alineamiento, se relocaliza la posición en Y de los ojos y la boca.*

Es interesante notar que, aunque las caras no estén centradas perfectamente, las proyecciones muestran gran robustez frente a variaciones de posición. Esta propiedad es la que garantiza que el ajuste en sentido vertical y en horizontal se puedan resolver de forma independiente; lo cual, a su vez, es una de las claves de la técnica propuesta.

Alineamiento vertical de la cara

Después de proyectar verticalmente la cara, se realiza el proceso de alineamiento de la señal PV_{cara} al modelo MV_{cara}. A diferencia del paso anterior, se debe resolver tanto el desplazamiento como la escala, por lo que resulta necesario aplicar el algoritmo 2.4 para el alineamiento rápido de integrales proyectivas.

El algoritmo de alineamiento incluía dos parámetros adicionales: el intervalo de valores buscados en el desplazamiento, y el intervalo en la escala. El primero se deduce directamente del tamaño del área de tolerancia, y_{tol}, mientras que el segundo está asociado al factor de escala, f, utilizado en el algoritmo de detección.

Como resultado de la aplicación del algoritmo, tenemos los valores de desplazamiento, d_{pv}, y escala, e_{pv}, para el alineamiento óptimo de la señal al modelo. En la figura 4.14 se muestra una señal antes y después de ser alineada al patrón MV_{cara}.

Pero lo realmente interesante del alineamiento es que nos permite relocalizar la posición vertical de los componentes faciales. En concreto, teniendo en cuenta que en el modelo las alturas de los ojos y la boca son y_{ojos} e y_{boca}, respectivamente, las posiciones relocalizadas de ambos en la imagen i_{ext} serán:

- Posición Y de ojos: $y_{tol} + y_{ojos} \cdot e_{pv} + d_{pv}$
- Posición Y de boca: $y_{tol} + y_{boca} \cdot e_{pv} + d_{pv}$

En consecuencia, podemos decir que con este paso hemos resuelto la localización vertical del rostro. Se puede observar en la figura 4.14, y en los ejemplos adicionales de la figura 4.15, cómo este paso es capaz de producir buenos resultados bajo situaciones difíciles, como ojos cerrados, oclusión parcial, o bigote y barba. Esto es posible porque se tiene en cuenta toda la estructura de la cara; tratar de localizar los componentes por separado sería mucho más problemático en esos casos.

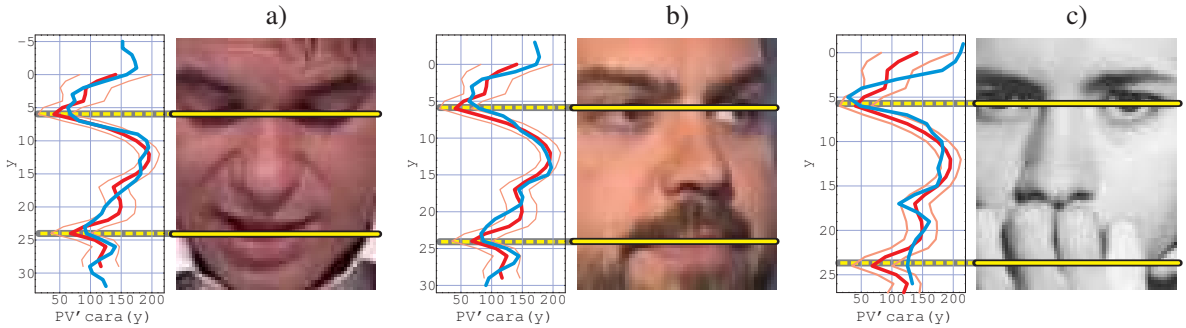


Figura 4.15: *Resultados del segundo paso del algoritmo de localización de componentes. Para cada ejemplo, se muestra la cara extraída, i_{ext} (derecha), y la proyección vertical, PV_{cara} (en azul), alineada respecto del modelo, MV_{cara} (en rojo). Se señala (en amarillo) la posición en Y resultante de ojos y boca, en función de los resultados del alineamiento. Las caras corresponden a: a) 617.jpg (UMU), b) 9H3.jpg (UMU), c) natalie1.gif (CMU/MIT).*

4.3.4. Alineamiento horizontal de la cara

En analogía al segundo paso, en esta última fase del algoritmo de localización se realiza un refinamiento de la posición en X de la cara utilizando proyecciones horizontales. Básicamente, se extrae la proyección horizontal de los ojos, PH_{ojos}, y se alinea respecto de MH_{ojos}. Cabría servirse también de la proyección de la boca, PH_{boca}. Empero, ya pudimos comprobar en el apartado 2.2.5 del capítulo 2 que esta señal es mucho menos informativa, y resultaba prácticamente irrelevante para una clasificación boca/no boca.

Cálculo y alineamiento de la proyección horizontal de ojos

El modelo de cara propuesto en la página 61 define dos topes verticales, $y_{ojosmin}$, $y_{ojosmax}$, de la región proyectada en PH_{ojos}. Esos límites verticales, en la imagen i_{ext}, deben ser reajus-

tados según los resultados del segundo paso (de forma parecida a y_{ojos} e y_{boca}). Además, la región proyectada incluye las áreas de tolerancia a ambos lados.

En consecuencia, se define la región $ojos$ de i_{ext}, como el rectángulo:

$ojos = (0, y_{tol} + y_{ojosmin} \cdot e_{pv} + d_{pv}) \leftrightarrow (2x_{tol} + w, y_{tol} + y_{ojosmax} \cdot e_{pv} + d_{pv})$

Como en el paso anterior, se calcula la proyección PH_{ojos} y se alinea respecto del modelo MH_{ojos} aplicando el algoritmo 2.4. El resultado es el desplazamiento, d_{ph}, y la escala, e_{ph}, en sentido horizontal. En la figura 4.16 se muestra gráficamente este último paso del localizador.

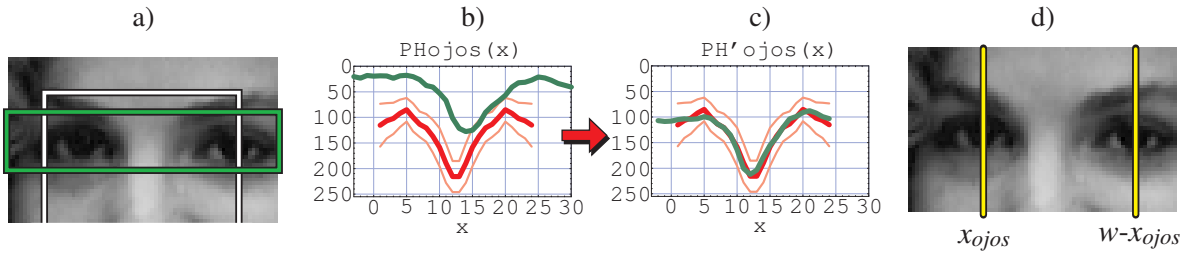


Figura 4.16: *Obtención y alineamiento de la proyección horizontal de ojos, en el algoritmo de localización. a) Rectángulo usado para la región ojos (en verde) según los parámetros del modelo y los resultados del alineamiento vertical. b) Proyección horizontal de los ojos, PH_{ojos} (en verde), y el modelo correspondiente, MH_{ojos} (en rojo). c) Las mismas proyecciones después del alineamiento. d) Con los resultados del alineamiento se localiza la posición en X de los ojos.*

Posición resultante de los elementos faciales

Tras el alineamiento vertical y el horizontal, conocemos las posiciones de ojos y boca en la imagen i_{ext}. Las coordenadas correspondientes en la imagen original se pueden obtener simplemente deshaciendo la transformación afín definida por la matriz M.

Más concretamente, las posiciones son:

- Posición del ojo izquierdo: $(x_{tol} + x_{ojos} \cdot e_{ph} + d_{ph}, y_{tol} + y_{ojos} \cdot e_{pv} + d_{pv})$
- Posición del ojo derecho: $(x_{tol} + (w - x_{ojos}) \cdot e_{ph} + d_{ph}, y_{tol} + y_{ojos} \cdot e_{pv} + d_{pv})$
- Posición de la boca: $(x_{tol} + w/2 \cdot e_{ph} + d_{ph}, y_{tol} + y_{boca} \cdot e_{pv} + d_{pv})$

Adoptamos el criterio de situar siempre la boca entre ambos ojos. Esto no deja de ser una aproximación, aunque veremos que funciona bien en la mayoría de los casos.

Finalmente, las posiciones originales, (x_{orig}, y_{orig}), para un componente situado en (x, y) en la imagen i_{ext} serán:

$$\begin{pmatrix} x_{orig} \\ y_{orig} \\ 1 \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.18}$$

En la figura 4.17 se pueden ver algunas localizaciones resultantes del algoritmo propuesto. Debemos aclarar que la elipse contenedora de la cara se ha obtenido simplemente a partir del centro geométrico de ojos y boca, del ancho y alto de la cara, y del ángulo estimado en el primer paso del proceso.

Figura 4.17: *Posiciones resultantes del algoritmo de localización de componentes. Se ha utilizado en la entrada el método de detección de caras combinado (Haar+IP). De arriba abajo, de izquierda a derecha, se muestran fragmentos de: werbg04.gif (CMU/MIT), 2024.jpg (UMU), 617.jpg (UMU), natalie1.gif (CMU/MIT), brian.gif (CMU/MIT), tf5189a.gif (CMU/MIT), 9H3.jpg (UMU), 4012.jpg (UMU).*

4.4. Resultados experimentales

Una vez desarrollada nuestra propuesta de localización de componentes faciales mediante integrales proyectivas, en esta sección nos vamos a centrar en la evaluación de su precisión, eficiencia y fiabilidad. Ya hemos visto en el repaso del estado del arte que sólo recientemente han aparecido resultados contrastables sobre algunas bases de caras estándar [89, 213, 76, 38, 128, 197]. Por ello, las pruebas comparativas se han llevado a cabo con varios métodos alternativos –implementados por nosotros o disponibles públicamente–, ejecutados sobre la base propia y sobre las imágenes etiquetadas de la base FERET [52].

En particular, disponemos de dos localizadores ya creados, uno basado en redes neuronales [152], y otro en búsqueda de contornos [35]. Además, hemos programado desde cero otras dos técnicas adicionales, que usan búsqueda de patrones y autoespacios [137]. Todos los métodos de localización serán contrastados con el error de precisión manual y con el que ofrece el propio detector de caras; esto es, suponemos un localizador *trivial* –o también llamado "nulo"–, basado en posiciones fijas sobre el rectángulo contenedor de cara.

Para la ejecución de las diferentes pruebas se ha construido un entorno de experimentación, que se muestra en la figura 4.18. La aplicación funciona por lotes, manejando conjuntos de pruebas definidos por el usuario, aunque también permite una visualización directa de los resultados parciales.

Los experimentos sobre la base de caras UMU y sobre FERET están orientados a distintos escenarios de trabajo. En el primero se pueden encontrar situaciones muy variadas y complejas en cuanto a expresión facial, iluminación, pose y resolución, por lo que está más próximo

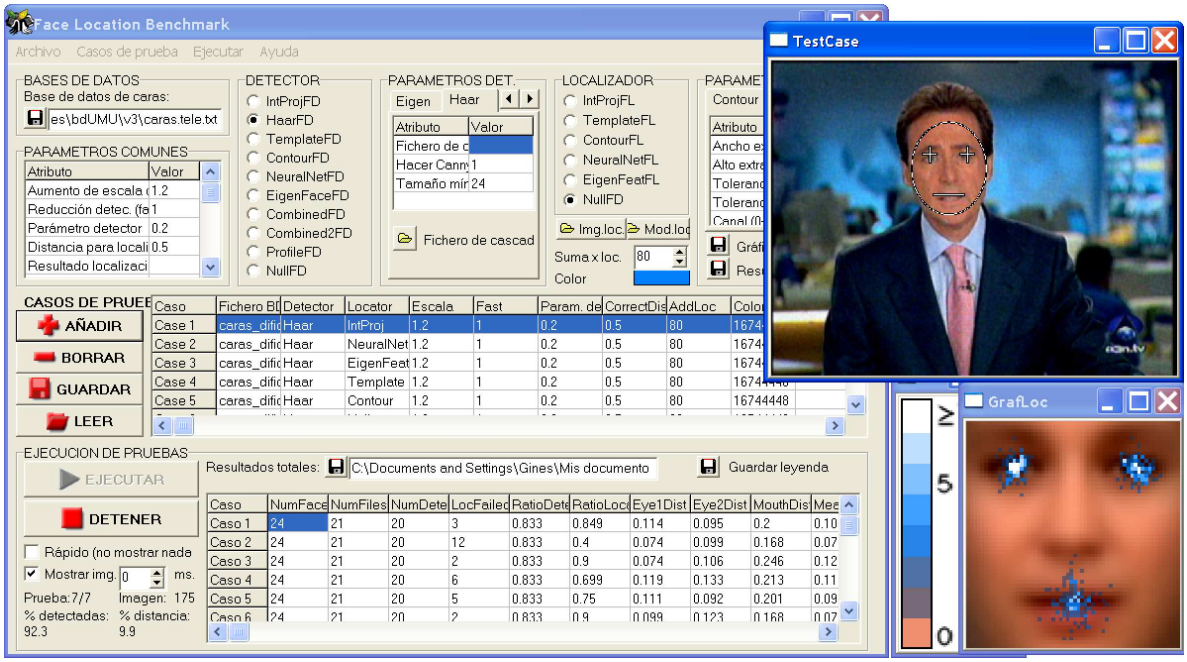


Figura 4.18: *Aplicación creada para la ejecución de los experimentos de localización. En la parte superior de la ventana, los datos que definen cada caso de prueba (base de datos, detector, localizador, etc.). En medio, el conjunto de casos de prueba. Abajo, los resultados de los casos. A la derecha, visualización de los resultados parciales.*

a una aplicación de *análisis de contenido multimedia*. Muchas de esas fuentes de variación no aparecen en la base FERET, en la que los individuos están en primer plano, con buena resolución, mirando de frente a la cámara y con ligeras inclinaciones; por lo tanto, es más similar a un escenario de *reconocimiento facial* de personas. Sin embargo, resulta interesante por la variedad de individuos que presenta, con diferentes edades, etnias, formas de cara, aspectos y elementos faciales.

Seguidamente vamos a detallar las pruebas realizadas y los resultados obtenidos. En primer lugar explicamos los métodos usados en la comparativa, en el apartado 4.4.1, concretando algunos aspectos de implementación. El apartado 4.4.2 describe el desarrollo de las pruebas. A continuación presentamos los resultados sobre los dos conjuntos de imágenes manejados, en los apartados 4.4.3 y 4.4.4. Para cada uno de ellos realizamos una valoración exhaustiva de los datos obtenidos, desgranando los aspectos más relevantes. Finalmente, en la sección 4.5 se pueden encontrar las conclusiones globales de los experimentos.

4.4.1. Métodos alternativos de localización de componentes

El código del método de localización propuesto, así como el de las técnicas alternativas, ha sido incorporado al conjunto de librerías de procesamiento de caras descrito en la sección 3.4. Esto nos permite realizar todas las posibles combinaciones de localizadores con detectores de caras: cualquier localizador puede trabajar con la salida de cualquier detector.

Vamos a detallar los métodos de localización analizados en los experimentos, indicando

los acrónimos usados en adelante para cada uno de ellos. La técnica basada en integrales proyectivas será denotada por **IntProy**. Los modelos de proyección usados son los mismos que en el capítulo 3 (ver la página 131). Los demás ajustes del algoritmo han sido ya indicados en la sección 4.3.

Nulo - Localización según los resultados del detector

No todos los detectores producen los mismos rectángulos contenedores para las caras existentes; algunos devuelven regiones más grandes –que incluyen toda la cabeza–, y otros, fragmentos más ajustados al rostro. No obstante, para cada detector se pueden hallar las posiciones donde se encontrarán, a priori, los elementos de una cara encontrada. Por ejemplo, según nuestros ensayos, en el detector de Haar disponible el ojo izquierdo se sitúa a un 40 % de la altura de la cara (es decir, del rectángulo devuelto por el algoritmo) y a un 33 % del ancho. De esta manera, todos los localizadores partirán de esas posiciones iniciales, con total independencia de los rectángulos de cara devueltos por cada detector.

Lo anterior nos ofrece un método nuevo, y trivial, de localización de componentes faciales, que hemos denominado *localizador nulo*: dado el rectángulo contenedor de la cara, calcular las posiciones de los componentes según unas proporciones fijas asociadas al algoritmo de detección correspondiente. Este método directo se puede tomar como la base para la comparación; en principio, su resultado no tiene por qué ser excesivamente malo si el detector es lo suficientemente preciso.

NeuralNet - Localizador de ojos mediante redes neuronales

Este localizador utiliza un conjunto de funciones ofrecidas públicamente por Henry Rowley [152], dentro de unas librerías de procesamiento de caras humanas con redes neuronales. Recordemos que esta librería fue utilizada ya en las pruebas de detección del capítulo 3.

Existen dos operaciones disponibles para hacer uso de la localización de ojos: (a) una función para detectar un ojo en una imagen (aplicada sobre la región esperada del mismo); y (b) una operación de detección de caras que realiza también la localización de los ojos. En los experimentos descritos adelante utilizaremos la segunda opción.

Es interesante anotar que ambas operaciones indican si se han podido encontrar los ojos o no. En caso negativo, diremos que ha ocurrido un fallo de localización. Lógicamente cuantos menos fallos de este tipo ocurran, mejor será el método.

Los resultados obtenidos con este localizador son en general muy buenos, aunque adolecen de una baja eficiencia computacional. Otro inconveniente de la implementación es que no hay ninguna función disponible para buscar la boca. Esta cuestión se ha resuelto de forma aproximada, calculando la posición de la boca en función de los ojos, según unas proporciones estándar.

TemMatch - Localizador mediante búsqueda de patrones medios

Se trata de un método sencillo de localización de componentes faciales –posiblemente el más intuitivo– basado en búsqueda de patrones (*template matching*), usando modelos medios para los ojos y la boca. Recordemos que en las pruebas del capítulo 3 concluimos que la detección de caras con patrones medios no produce buenos resultados. Sin embargo, en el problema de localización, donde el área de búsqueda está más restringida y sólo se requiere una aparición de cada componente, esta técnica puede resultar más plausible.

En la primera columna de la figura 4.19 se pueden ver los modelos medios aplicados en los experimentos para cada componente facial. Estas imágenes han sido calculadas promediando un subconjunto grande de caras de la base UMU. Aunque algunas de ellas también son usadas después para test, debemos aclarar que el riesgo de sobreajuste es prácticamente inexistente en un modelo medio de este tipo.

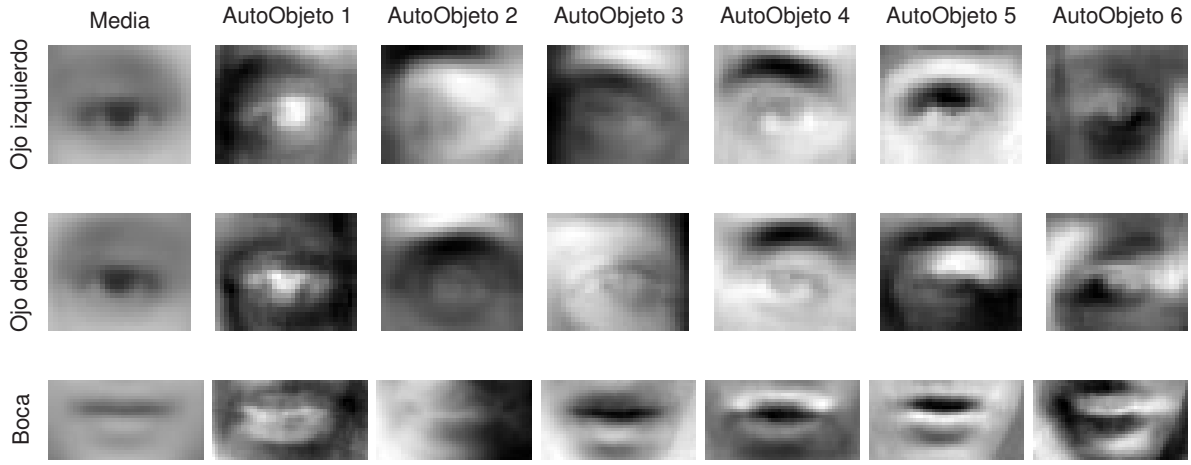


Figura 4.19: *Patrones medios y autovectores asociados a cada componente facial, ojo izquierdo, ojo derecho y boca. Los patrones de los ojos son de* 30×25 *píxeles, y el de la boca de* 49×25*. Los tres han sido obtenidos de un subconjunto de 242 caras de la base UMU.*

La búsqueda de patrones no se realiza directamente sobre las imágenes originales, sino que las caras devueltas por el detector son extraídas mediante una transformación afín a una posición y tamaño estándar de 96×120 píxeles, de forma similar a la explicada en el apartado 4.3.2. Después, se aplica el *matching* para los ojos y la boca en las zonas correspondientes de las imágenes extraídas. Para la búsqueda de los patrones mediante *matching* se aplica una medida de correlación normalizada (ver la ecuación 2.22, en la página 54). En el caso de las imágenes en color se toma el canal R. Finalmente, se deshace la transformación afín para obtener las localizaciones en la imagen original.

EigenFeat - Localizador mediante auto-componentes

Esta técnica implementa el método basado en autoespacios modulares propuesto por Pentland y otros [137]. Para cada posible posición de ojo izquierdo, ojo derecho y boca, se

proyecta el trozo de imagen en el autoespacio asociado a ese elemento. La *distancia al autoespacio* (DFFS) –o, equivalentemente, el error de reproyección– se utiliza para seleccionar la mejor posición del elemento. Se pueden ver algunos de los *autoojos* y *autobocas* utilizados en la figura 4.19.

Por su forma de operar, el proceso de localización es muy similar al basado en búsqueda de patrones. De hecho, las implementaciones de ambos comparten mucho código en común. Igual que en el método anterior, las caras se extraen a un tamaño de 96×120 píxeles, usando el canal R en las imágenes en color. El código utiliza las funciones optimizadas de la librería Intel OpenCV [35], para el cálculo de autoespacios a partir de un conjunto de datos, y para la proyección de patrones en la base creada.

La efectividad de este método de localización puede variar sensiblemente con el número de autovectores utilizados, es decir, con el tamaño de la base. Utilizar un tamaño mayor no necesariamente mejora la precisión de la localización. En nuestras pruebas hemos podido comprobar que el número óptimo de componentes se encuentra siempre entre 4 y 5. Por defecto, usaremos tamaño 4 en los experimentos, a menos que se indique lo contrario.

Cont - Localizador mediante contornos por umbralización de intensidad

La localización de componentes por umbralización de niveles de gris es una de las estrategias básicas que hemos analizado en la sección 4.2.1. Igual que hemos dicho de la técnica basada en búsqueda de patrones, este método –que resultaba inadecuado para la detección de caras– puede ser más prometedor en el problema de localización de componentes faciales.

Para esta técnica partimos de la implementación disponible en OpenCV [35]. Debemos recordar que se trata de una funcionalidad experimental y, por lo tanto, en proceso de depuración y mejora. Aun así, consideramos interesante incluirla en la comparativa, ya que la idea subyacente al método difiere esencialmente del resto de técnicas analizadas.

Las operaciones ofrecidas por la librería representan cada componente de la cara –ojo izquierdo, derecho y boca– mediante un rectángulo contenedor. Estos rectángulos están asociados a regiones contiguas con tono oscuro, teniendo en cuenta restricciones geométricas de posición, forma y tamaño. A partir de estos resultados, localizamos cada elemento facial en el centro del rectángulo correspondiente.

4.4.2. Descripción de los experimentos

Antes de presentar los resultados de los experimentos, debemos hacer las siguientes puntualizaciones sobre su realización:

- En la entrada del proceso se utilizan los resultados del **detector combinado Haar+IP**. Seleccionamos esta opción por ser la que mejores porcentajes de detección consigue. Aunque hemos llevado a cabo numerosas pruebas con otros detectores, los resultados en cuanto al problema de localización son prácticamente idénticos.

- Los **falsos positivos** producidos por el detector son eliminados a priori de la entrada. Algunos localizadores podrían descartarlos por sí mismos –al encontrar que realmente no existe una cara–, mejorando así los resultados de la detección. Sin embargo, también se podrían eliminar caras verdaderas. Creemos conveniente centrarnos aquí en la precisión de localización, por lo que las falsas detecciones son eliminadas usando los datos del etiquetado manual.
- Para cada técnica se mide el número de **fallos de localización**. Estos fallos pueden ocurrir en dos casos: cuando el método informa de la imposibilidad de encontrar los componentes, y cuando las posiciones resultantes de alguno de los ojos están muy alejadas de las esperadas (en concreto, por encima del 20 % de la distancia entre los ojos). En el primer caso, el localizador no modifica las posiciones devueltas por el detector, es decir, se usan las obtenidas mediante el denominado "localizador nulo"[7].

Parámetros y medidas de precisión

En la sección 4.1.3 analizamos detenidamente los objetivos de la localización, y definimos algunas posibles métricas y criterios de bondad. Concretamente, las medidas tomadas en nuestros experimentos para cada una de las seis técnicas serán las siguientes:

- **Ratio loc.**: ratio de localización. Proporción de caras en las que no ha ocurrido un fallo de localización; se indica también el número absoluto de esos fallos. El ratio se puede interpretar como el porcentaje de las caras que han sido localizadas "más o menos correctamente"[8].
- **Dif. ángulo**: diferencia media absoluta entre el ángulo real de inclinación de la cara y el ángulo estimado por el localizador, en grados. Recordemos que el ángulo está dado en función de la línea que pasa por ambos ojos (ecuación 4.5).
- **Dif. tamaño**: diferencia media entre el tamaño real de cara y el obtenido, en proporción al primero. Usamos como medida de tamaño la distancia entre la línea de los ojos y la boca (lo que denominamos "tamaño vertical", ecuación 4.4).
- **Dif. posición**: distancia media entre la posición central de la cara y la calculada. La posición central es el punto medio de ojos y boca. Esta diferencia, así como las restantes medidas de error, están dadas en relación a la distancia real entre los ojos.
- **Error ojo izq., error ojo der., error boca**: errores medios de localización del ojo izquierdo, el ojo derecho y la boca, respectivamente. El error se define como la distancia euclídea entre el punto estimado y el etiquetado manualmente (ecuación 4.2). En algunos casos nos quedaremos únicamente con el promedio del error en ambos ojos.

[7] De esta forma, se podría dar la paradoja de que ocurriera un fallo de localización pero las posiciones devueltas fueran muy precisas. En cualquier caso, no sería mérito del localizador sino del detector.

[8] Recordemos que el fallo de localización se fija en un 20 % de la distancia entre los ojos, lo que permite un error máximo aproximado de unos 14 mm en el peor caso.

- **Tiempo**: tiempo medio de ejecución del proceso de localización. Se mide en milisegundos por cara. Dentro de este tiempo no se incluye el consumido en la lectura del fichero ni en la aplicación del detector, que ya fueron analizados en la sección 3.4.

Además de los resultados numéricos, creemos interesante mostrar gráficamente las localizaciones resultantes en relación a una cara estándar –como se muestra en la figura 4.4a)–. Este formato nos permite observar los tipos de errores cometidos por cada técnica, el grado de uniformidad del método, los casos extremos y las zonas de mayor ambigüedad en la localización. También incluiremos las curvas de distribución de errores y la curva acumulada de distancias, definidas y presentadas en las figuras 4.4b,c).

Estimación del error de etiquetado manual

No olvidemos que al hablar de posiciones "reales" de los componentes nos estamos refiriendo más exactamente a las obtenidas mediante el etiquetado manual, el cual, como es evidente, estará sujeto a cierto error de medición.

Para estimar este error hemos llevado a cabo un pequeño experimento. Se ha pedido a cuatro personas distintas que realizaran el etiquetado del mismo conjunto de 39 caras de la base UMU, con un programa creado a tal efecto. Después se han medido las discrepancias entre las posiciones señaladas por los cuatro individuos. De esta manera obtenemos un valor aproximado para la precisión de la localización manual.

Los resultados de este ensayo se muestran en la tabla 4.1.

Componente facial	**Desviación media** píxeles / % dist. interocular / milímetros
Ojo izquierdo	0,90 / 3,5 % / 2,5 mm
Ojo derecho	0,99 / 4,1 % / 2,8 mm
Nariz	2,06 / 6,7 % / 4,7 mm
Boca	1,24 / 4,9 % / 3,4 mm
MEDIA (ojos+boca)	1,04 / 4,2 % / 2,9 mm

Tabla 4.1: *Errores de precisión del etiquetado manual de los componentes faciales. Se han usado 39 caras y 4 etiquetados manuales. El tamaño medio de las caras (distancia interocular) es de 33 píxeles. Los errores se indican con tres medidas: en píxeles (distancias euclídeas absolutas), en proporción a la distancia interocular (para cada cara, la media de los 4 etiquetados), y en milímetros en el universo de la cara. Para esta última se ha supuesto que la distancia típica entre los ojos es de 70 milímetros. En la media final no se incluye la nariz, ya que no se usa en los experimentos.*

Podemos hacer algunas valoraciones de estos datos. Por un lado, no todas las partes de la cara se localizan con igual precisión. La más imprecisa –y con gran diferencia– es la nariz, aunque su posición no se utiliza en los experimentos posteriores. El error para la boca también es relativamente alto. Esto es debido a la mayor ambigüedad implícita en su posición media, por ejemplo, cuando la boca está abierta. La precisión de los ojos es significativamente mejor, muy similar en ambos, pero algo mejor para el izquierdo.

El valor medio estimado para el error de localización manual, del 4,2 % de la distancia

interocular, marca de alguna forma el error óptimo que teóricamente se puede esperar alcanzar. Dicho de otra forma, cualquier precisión por debajo de ese límite será meramente ficticia. No obstante, este dato depende fuertemente del tamaño de las caras en las imágenes; cuanto mayor sea la resolución de entrada, los errores manuales serán menores. Por ejemplo, seleccionando sólo las 10 caras mayores (cuyo tamaño medio es de 55 píxeles), el error de los ojos en píxeles se mantiene prácticamente igual, pero en porcentaje se reduce al 2 %.

4.4.3. Resultados de las pruebas sobre la base UMU

En este primer experimento se aplican los 6 métodos de localización a las 737 imágenes de la base de caras propia. Sobre esas imágenes se ejecuta el detector combinado Haar+IP, descartando de antemano los falsos positivos, como ya hemos justificado.

Se consideran aquí como *falsos positivos* las caras en las que la distancia de algún ojo (según el localizador nulo) a la posición del etiquetado manual supera el 50 % de la distancia interocular[9]. En total, se encuentran de esta forma 730 de las 854 caras existentes, es decir el 85,4 %, que constituyen la entrada para las pruebas de localización.

Los resultados totales obtenidos para los distintos métodos se resumen en la tabla 4.2.

Método	Ratio loc. (nº fallos)	Dif. ángulo	Dif. tamaño	Dif. posic.	Error ojo izq.	Error ojo der.	Error boca	Tiempo (ms)
Manual	100 % (0)	1,3º	3,1 %	3,0 %	3,5 %	4,1 %	4,9 %	–
Nulo	91,2 % (64)	4,53º	9,6 %	6,2 %	8,2 %	10,1 %	13,6 %	–
IntProj	95,9 % (30)	**1,64º**	**6,7 %**	**4,7 %**	6,0 %	6,7 %	**9,5 %**	1,8
NeuralNet	74,7 % (185)	2,50º	9,9 %	5,9 %	6,1 %	6,8 %	13,7 %	323,6
TemMatch	**96,3 % (27)**	1,75º	7,6 %	6,2 %	6,6 %	**6,6 %**	13,1 %	7,4
EigenFeat	94,8 % (38)	2,00º	9,2 %	6,2 %	**5,4 %**	**6,6 %**	13,7 %	20,5
Cont	78,5 % (157)	4,03º	11,9 %	8,7 %	10,3 %	11,1 %	16,6 %	**0,9**

Tabla 4.2: *Resultados de los distintos localizadores sobre la base de caras UMU. La entrada son 730 caras detectadas con Haar+IP. Para cada método, se muestran las medidas definidas en el apartado 4.4.2. Los errores están en proporción a la distancia interocular. Se señala en negrita el mejor resultado obtenido para cada parámetro estudiado (obviando el método manual). Los datos del ordenador usado aparecen en la tabla 3.2 (ver la página 131).*

Evidentemente, todos los algoritmos disponen de diferentes parámetros y modos de operación ajustables, cuyos valores pueden influir en los resultados finales de la localización. Para evitar posibles discrepancias, se han ajustado todos ellos, mediante ensayo y error, para conseguir un funcionamiento óptimo. Algunos de estos ajustes han sido ya comentados. Otros los mencionaremos a continuación.

En la figura 4.20 se muestran las gráficas de densidad de localizaciones producidas por las técnicas analizadas. Esta representación aporta una información muy interesante sobre los tipos de imprecisiones en que incurre cada método.

[9]Debemos aclarar que este criterio es un poco más exigente que el aplicado en los experimentos de detección de caras del capítulo 3.

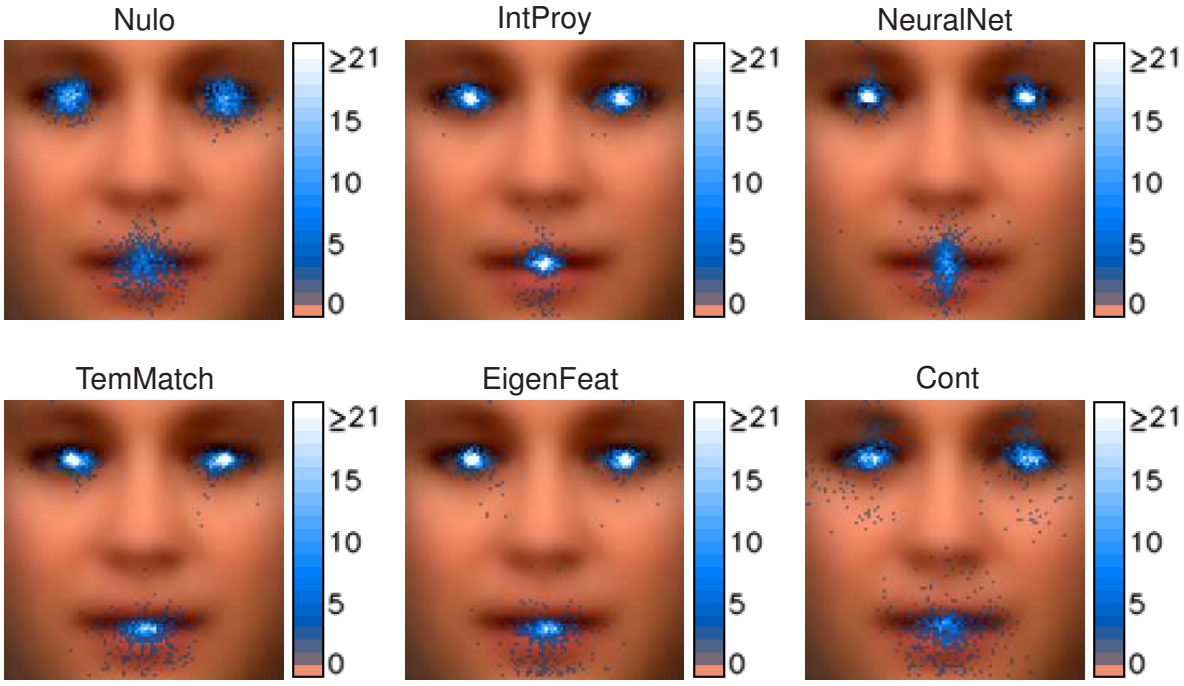


Figura 4.20: *Gráficas de densidad de localizaciones resultantes de los distintos métodos para la base UMU. Sobre una cara media estándar, se representan las densidades de puntos localizados por cada técnica en las distintas partes de la cara (ver la leyenda a la derecha).*

Los porcentajes de localizaciones resultantes en función de la distancia interocular aparecen representados en las curvas de la figura 4.21. Se incluyen gráficas con el número de casos absoluto y con el ratio acumulado, para ojo izquierdo, derecho y boca.

Por último, se pueden ver algunos ejemplos concretos de localizaciones en las figuras 4.22, 4.23 y 4.24. En la primera de ellas se comparan las posiciones producidas por el localizador nulo (en rojo) con la técnica propuesta en este capítulo (en blanco). En las otras se incluyen algunos de los métodos alternativos.

Análisis y valoración de los resultados

Vamos a discutir, punto por punto, las conclusiones que podemos extraer de este primer experimento de localización, empezando por las más generales y acabando por las más específicas. A lo largo de la discusión iremos añadiendo también algunos datos adicionales a los contenidos en la tabla 4.2 y en las figuras mencionadas.

1. **Valoración global y cotas teóricas del error.**

   El primer hecho destacable es la gran *similitud de resultados* en muchos de los parámetros observados, si descartamos los malos datos del localizador basado en contornos. Por ejemplo, en relación al error en los ojos, todos los métodos se mueven entre el 5,4 % y el 6,8 %. En varios casos el método "ganador" (señalado en negrita) lo es por un estrecho margen o se producen empates. Esto puede indicar que los ratios alcanzados se

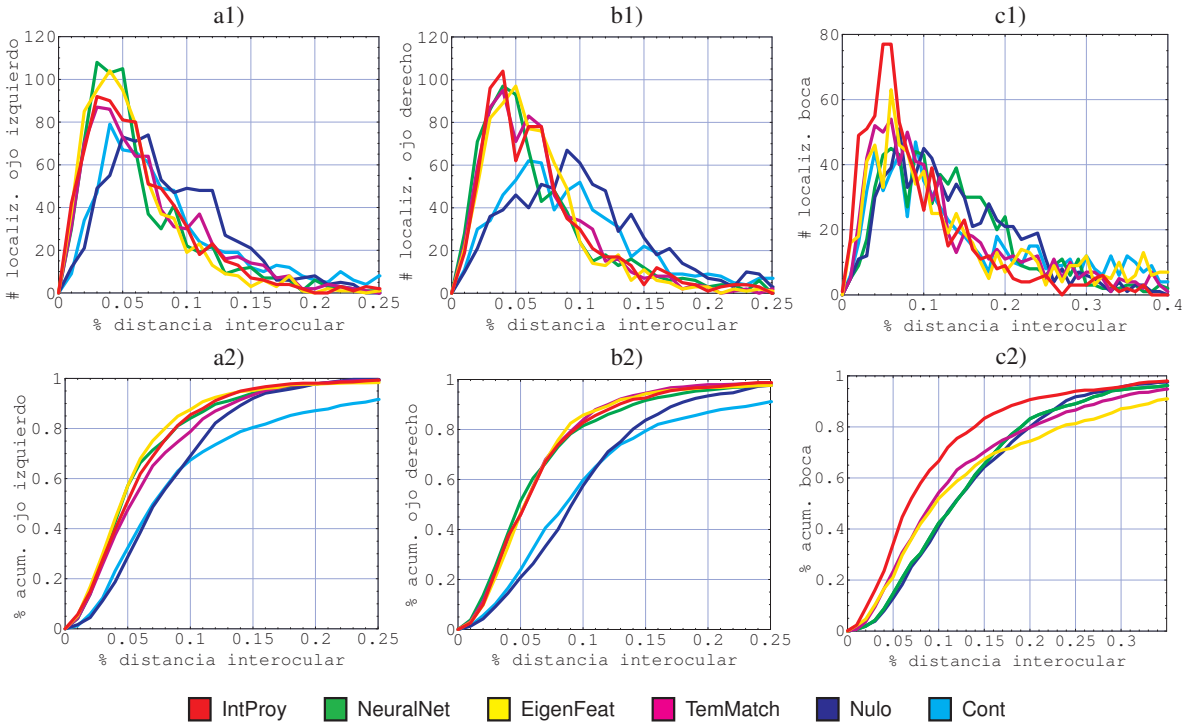


Figura 4.21: *Curvas de distribución de los errores de localización sobre la base UMU, para los distintos métodos. Las gráficas representan el número de casos para cada porcentaje de error (distancia entre la posición obtenida y la etiquetada, en proporción a la separación interocular). a1,b1,c1) Número absoluto de casos. a2,b2,c2) Porcentaje acumulado. a) Ojo izquierdo. b) Ojo derecho. c) Boca.*

encuentran relativamente cerca del óptimo teórico. Para el caso de los errores de precisión, el **menor error alcanzable** parece estar en proporción a la precisión manual. En concreto, ningún método logra errores menores que 1,5 veces la precisión manual.

En relación con lo anterior, si el límite inferior lo marca el error manual, el **límite superior** vendría dado por el error del método "Nulo". Recordemos que esta medida de precisión es la que ofrece el detector, es decir, es el error *antes* de aplicar el localizador. Cualquier valor que lo supere se debe considerar como un mal resultado del localizador correspondiente, ya que supondría empeorar las posiciones de entrada. Sin embargo, no es infrecuente encontrar valores por encima de ese dato. Incluso, en cuanto a la diferencia de posición central, el localizador nulo es el tercer mejor método, sólo superado por IntProy y NeuralNet[10].

En cualquier caso, la proximidad de muchos valores al tope máximo es un indicio de la complejidad del problema que intentamos resolver: lo interesante de un localizador es la mejora que produce sobre las posiciones ofrecidas por el método nulo, pero esa mejora es difícil de conseguir en la práctica.

[10]Este caso concreto tiene cierta lógica, ya que es normal que la posición central de la cara quede desviada cuando algún componente no es localizado correctamente. En este sentido, el detector puede llegar a ser más fiable que aplicar algunos de los métodos de localización más elaborados.

Figura 4.22: *Ejemplos de localizaciones de ojos y bocas resultantes para la base UMU. Se indican los resultados del localizador nulo (en rojo) y del basado en integrales proyectivas (en blanco). De arriba abajo, de izquierda a derecha, se muestran extractos de: 156H3.ext.jpg, 2036.avi.jpg, 032.jpg, voyager2.gif (CMU/MIT), 00H2.ext.jpg, 003.jpg, 630.jpg, 1066.avi.jpg, 201.jpg, lacrosse.gif (CMU/MIT), 21.jpg, a133.jpg, brian.gif (CMU/MIT), 2075.avi.jpg, 1003.web.jpg, 69H2.ext.jpg, 95H4.ext.jpg, 16.jpg, 1061.avi.jpg, 110M3.ext.jpg.*

2. **Comparación y clasificación de los métodos.**

   Posiblemente, de todos los parámetros observados el que mejor resume la bondad global de un método es el *ratio de localización*. Como ya hemos comentado, el porcentaje indica el número de caras que se pueden considerar como bien localizadas, admitiendo un margen de error máximo en los ojos de unos 14mm[11].

   De acuerdo con esa medida, el primer método sería TemMatch (96,3 %), seguido muy de cerca por IntProy (95,9 %) y EigenFeat (94,8 %). El método NeuralNet se ve algo desfa-

[11] Aunque se podría usar otro margen –produciendo porcentajes de localización muy diferentes–, lo interesante aquí es la comparación entre métodos.

vorecido por este criterio, ya que son muchas las situaciones donde no se encuentran los ojos. Aunque no sea lo más adecuado, si consideramos sólo como fallos de localización los debidos a distancias grandes (y no los indicados por el localizador), el porcentaje para NeuralNet ascendería al 91,3 % (con 63 fallos existentes). Esto lo situaría en el cuarto lugar, justo detrás del método sencillo mediante auto-objetos. Finalmente, el algoritmo basado en contornos empeora los resultados del detector nulo, bajando el ratio casi 13 puntos (del 91,2 % al 78,5 %).

Si nos fijamos en los restantes parámetros de la tabla 4.2, el localizador IntProy ocupa también una posición muy destacada, logrando los mejores resultados para 4 de los 7 existentes. Sobresale especialmente en las medidas globales: diferencia de ángulo, del centro de la cara y del tamaño. Existe una buena razón para esto. El método basado en proyecciones es el único que realiza una localización global del rostro, mientras que los demás –a excepción del localizador nulo– se basan en encontrar cada uno de los componentes por separado.

En segundo lugar, se encontrarían TemMatch y EigenFeat, que consiguen la mejor posición en la localización de ambos ojos. Por su parte, los menores errores en la localización de la boca y la inclinación los produce IntProy, con una amplia ventaja sobre el resto.

Figura 4.23: *Ejemplos de fallos de localización sobre la base UMU. Se representan las posiciones resultantes del método basado en integrales proyectivas (en blanco), en redes neuronales (en verde) y en autoespacios (en azul). De arriba abajo, de izquierda a derecha, se muestran extractos de: 402.jpg, 2075.avi.jpg, 618.jpg, am5020a.gif (CMU/MIT), 2052.avi.jpg, 2032.avi.jpg, 628.jpg, y aerosmith-double.gif (CMU/MIT).*

3. **Distribución de los errores de localización.**

Analizando las gráficas de densidades y de distribuciones del error de las figuras 4.20 y 4.21, podemos extraer más conclusiones generales de interés. Por un lado, se puede ver que los errores de localización no sólo son debidos a imprecisiones "de grano fino" en la posición de los componentes, sino que son muchos los casos donde un localizador produce resultados completamente incorrectos: el ojo se sitúa en el tabique nasal, en las cejas o en el entrecejo, la boca se coloca en la nariz, etc. En la figura 4.23 se muestran algunos ejemplos de fallos típicos de localización, para diferentes métodos.

Desafortunadamente, ninguna técnica está libre de este tipo de errores. El ratio de localización es una medida de este problema, donde cada resultado se clasifica como correcto o incorrecto en función de un umbral de distancia máxima. Los porcentajes de error van del 3,7 % de TemMatch al 25,3 % de NeuralNet.

Por otro lado, las gráficas nos permiten también apreciar la *distribución de las nubes de puntos* para cada técnica, de lo cual se deduce la forma del error. En el caso de los ojos, se puede observar, en general, una mayor varianza a lo largo del eje horizontal que del vertical. Es decir, existe más imprecisión en la posición X que en Y. Por ejemplo, en el método IntProy, la varianza en la posición horizontal de los ojos es un 60 % superior a la vertical. Si comparamos las distintas técnicas entre sí, algunas suelen incurrir más en ciertos tipos de errores que otras. Así, por ejemplo, en TemMatch y en IntProy la posición de los ojos está muchas veces desviada hacia el lagrimal, mientras que NeuralNet tiende a situarlos en la esquina opuesta de los ojos. En la figura 4.24 se pueden ver algunos ejemplos de estas situaciones.

Figura 4.24: *Ejemplos de imprecisiones de localización sobre la base UMU. Se representan las posiciones resultantes del método basado en integrales proyectivas (en blanco), en redes neuronales (en verde), en autoespacios (en azul) y en búsqueda de patrones (en rojo). De izquierda a derecha, se muestran extractos de: 17.jpg, c451.jpg, 29.jpg y 1035.avi.jpg.*

En el caso de la boca, los errores se pueden clasificar en dos tipos: los que ocurren en sentido horizontal, y en vertical. Los del segundo tipo producen normalmente la colocación de la boca en la posición real de la nariz o en el labio inferior. El problema es más frecuente cuando la boca no se distingue con claridad del resto de la cara. Este fenómeno se puede comprobar claramente en algunas caras de las figuras 4.23 y 4.24, donde aparecen los resultados de diferentes métodos sobre una misma imagen.

A su vez, dentro de un mismo método, existe también una elevada indeterminación

en la posición horizontal del centro de la boca, como se observa en la figura 4.20. Sólo IntProy consigue reducir este problema. En concreto, la desviación estándar horizontal de las posiciones de boca –en relación a la distancia interocular– está en torno al 1,2 % en NeuralNet, TemMatch y EigenFeat, mientras que en IntProy es del 0,6 %.

Finalmente, resulta curioso el hecho de que, en la mayoría de los casos, la localización del ojo izquierdo sea más precisa que la del derecho. Para IntProy es un 12 % más precisa, y un 22 % para EigenFeat. Casualmente, también existe un 17 % mayor de precisión para ese ojo en la estimación manual[12]. El motivo puede encontrarse en el error de partida, ya que el método nulo ofrece posiciones con un 23 % más de error para el ojo derecho. Aunque los diferentes métodos consiguen reducir los errores de ambos ojos, permanece cierta "preferencia" hacia el izquierdo. En este sentido, destacan los resultados de TemMatch, que logra errores pequeños y muy similares para los dos ojos.

4. **Eficiencia computacional.**

   El coste de ejecutar los diferentes localizadores es un factor que no debe ser despreciado, ya que puede afectar a la viabilidad práctica de las técnicas. En este sentido, la tabla 4.2 muestra una gran disparidad de tiempos. Los resultados se representan gráficamente en la figura 4.25, en la que se han añadido también los tiempos mínimos y máximos de cada ejecución concreta del proceso de localización.

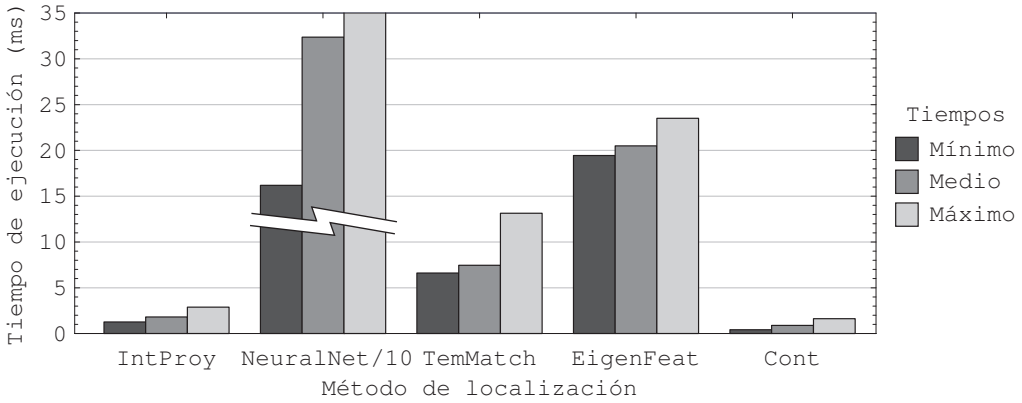


Figura 4.25: *Tiempos de ejecución de los localizadores sobre la base UMU. Para cada técnica de localización se representa el tiempo mínimo, promedio y máximo. Observar que los tiempos de NeuralNet están divididos por 10. El ordenador usado es un Pentium IV a 2,6GHz (ver la tabla 3.2, página 131).*

Los algoritmos más rápidos son Cont e IntProy, que no sobrepasan los 3 milisegundos por cara en el peor caso. Si tomamos el segundo método como referencia, TemMatch es 4 veces más lento, EigenFeat 11 veces más, y NeuralNet unas 180. Estos resultados son coherentes con la complejidad implícita de cada técnica. Mientras que IntProy trabaja

[12]En este caso, puede ser debido a que el propio conjunto de 39 caras de prueba presenta una ambigüedad ligeramente superior para el ojo derecho (por ejemplo, debido a giros o a sombras). Sin embargo, es difícil aplicar ese mismo razonamiento a los métodos no manuales, ya que se usan conjuntos mucho más grandes de ejemplos.

con patrones 1D, TemMatch lo hace en 2D, y EigenFeat requiere aplicar una proyección al autoespacio para cada posición candidata. Por encima de todos ellos, el proceso más costoso es la aplicación de las redes neuronales.

Sin duda alguna, el localizador basado en proyecciones ocupa el primer puesto en eficiencia, gracias a su excelente relación entre precisión y coste computacional. Sólo la técnica basada en contornos es más rápida, pero sus resultados incluso empeoran los del localizador nulo. Por otro lado, los métodos que consiguen menores errores que IntProy resultan varias veces más lentos, pero logran una mejora muy reducida (por ejemplo, de 0,1 puntos en el ojo derecho y 0,6 en el izquierdo).

5. **Precisión en función de la resolución, la inclinación y la fuente.**

   Hasta este punto hemos analizado los resultados de los localizadores sobre todo el conjunto de imágenes. Ahora vamos a diseccionar la base de caras para estudiar la fiabilidad y la robustez de las técnicas frente a diversos factores de interés: la resolución de las caras, su inclinación en las imágenes, y el origen de la captura.

   Igual que hicimos en las pruebas de detección, realizamos diferentes particiones de la base UMU, según los tres criterios definidos. Las pruebas de localización se repiten sobre cada una de estas particiones, obteniendo los resultados de cada fragmento de la base. Para simplificar, nos centramos aquí en las dos medidas más interesantes: el ratio de localización y el error medio de los ojos.

   En relación a la resolución, se han establecido 3 particiones según la distancia interocular de los rostros. Más específicamente, tenemos los siguientes tamaños:

   - **Pequeño**: de 12 a 30 píxeles (186 caras).
   - **Mediano**: de 31 a 60 píxeles (421 caras).
   - **Grande**: de 61 a 184 píxeles (246 caras).

   Los resultados de esta prueba se resumen en la tabla 4.3. Debemos aclarar que el número de caras se refiere a las existentes en la base UMU; lógicamente, no todas ellas son encontradas por el detector.

Tamaño	Nulo		IntProy		NeuralNet		TemMatch		EigenFeat	
caras	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
Pequeño	86,6	9,9	94,9	**6,8**	66,9	7,6	**96,2**	7,1	94,3	**6,8**
Mediano	92,5	8,9	**97,1**	**5,9**	78,1	6,2	96,0	6,3	94,4	6,2
Grande	92,5	8,9	92,5	6,5	74,0	6,1	**97,0**	6,7	96	**5,8**

Tabla 4.3: *Resultados de los localizadores sobre la base UMU en función de las resolución de las caras. Las distancias interoculares (en píxeles) para cada grupo son: pequeño (12-30), mediano (31-60), grande (61-184). Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

La primera conclusión de la tabla 4.3 es que, como cabría esperar, la precisión es baja cuando la resolución de las imágenes es escasa. En estas condiciones IntProy y EigenFeat, son las mejores alternativas. Sin embargo, una elevada resolución no garantiza necesariamente mejor localización. El tamaño óptimo parece estar en el rango de los 31 a 60 píxeles de distancia interocular. Más allá, los errores alcanzados se reducen muy lentamente o, incluso, pueden llegar a aumentar.

En cuanto a la inclinación, realizamos 5 particiones según los ángulos observados, dos a cada lado y una para giros muy reducidos. En particular, los grupos definidos son:

- **Grande izq.**: de -20º a -8,1º (97 caras).
- **Media izq.**: de -8º a -2,1º (262 caras).
- **Baja**: de -2º a 2º (226 caras).
- **Media der.**: de 2,1º a 8º (190 caras).
- **Grande der.**: de 8,1º a 20º (78 caras).

La tabla 4.4 contiene los porcentajes obtenidos para este experimento. Recordemos que el ángulo de giro devuelto por el localizador nulo es siempre 0.

Inclinación	Nulo		IntProy		NeuralNet		TemMatch		EigenFeat	
caras	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
Grande izq.	69,0	13,3	**94,4**	7,4	69,5	8,3	90,1	7,8	**94,4**	**7,1**
Media izq.	93,8	8,4	**96,3**	6,0	76,0	6,4	**96,3**	6,1	95,0	**5,8**
Baja	95,3	7,8	94,8	6,6	74,4	6,7	**96,7**	6,9	92,4	**6,0**
Media der.	94,8	9,0	96,8	**6,1**	75,5	6,4	**97,4**	6,6	95,5	**6,1**
Grande der.	81,8	12,3	98,2	**6,0**	70,9	6,9	**100**	6,1	85,5	8,2

Tabla 4.4: *Resultados de los localizadores sobre la base UMU en función de la inclinación de las caras. Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

Evidentemente, en el localizador nulo los errores aumentan cuanto mayor es el ángulo de inclinación. Sin embargo, el resto de métodos presenta una buena robustez frente a este factor, logrando reducir la imprecisión de entrada. Un caso claro es la inclinación grande a la izquierda, donde se pasa del 69 % de localización correcta en Nulo, al 94,4 % en IntProy y EigenFeat. De esta forma, en los métodos avanzados no existe una relación directa entre el giro y la precisión conseguida. Por ejemplo, en IntProy el ratio de localización siempre está por encima del 94,4 % y el error medio por debajo de 7,5 %. En este sentido, se puede decir que es el más robusto para todos los casos.

Por último, presentamos los resultados según la fuente de entrada. Las imágenes de la base UMU han sido clasificadas en los cinco grupos siguientes, según el origen de la captura:

- **CMU/MIT**: 34 imágenes de la base de caras CMU/MIT (64 caras).

- **TV analóg.**: imágenes capturadas de televisión analógica, fundamentalmente de programas de noticias, reportajes y series (450 caras).
- **TDT**: tomadas de televisión digital terrestre, extraídas de series, noticias y publicidad (120 caras).
- **Webcam**: varias cámaras de videoconferencia en condiciones de interior (57 caras).
- **DVD**: extractos de películas en formato DVD (162 caras).

La distribución entre grupos no es tan uniforme como en los casos anteriores, y algunos de ellos son muy reducidos como para poder extraer conclusiones fundamentadas. No obstante, es interesante analizar la mejora que pueden producir los diferentes métodos sobre cada grupo. El objetivo no es tanto estudiar la influencia del sistema de captura, sino el tipo de contenido habitual en cada categoría. Los resultados de esta prueba se pueden consultar en la tabla 4.5.

Fuente captura	Nulo		IntProy		NeuralNet		TemMatch		EigenFeat	
	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
CMU	84,6	10,5	95,4	7,3	66,2	7,3	**96,9**	7,5	95,4	**7,0**
TV analóg.	92,3	8,9	96,5	**5,8**	74,6	6,4	96,8	6,1	**97,0**	**5,8**
TDT	87,0	9,6	**96,0**	6,7	73,0	**6,5**	93,0	7,7	92,0	6,8
Webcam	**100**	7,5	**100**	5,9	92,0	6,0	98,0	7,1	**100**	**5,6**
DVD	89,5	9,6	88,6	7,6	72,8	**6,4**	**96,5**	6,9	86,8	7,5

Tabla 4.5: *Resultados de los localizadores sobre la base UMU en función de la fuente de captura. Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

Globalmente, las fuentes de captura que alcanzan una mayor precisión son Webcam y TV analógica. En gran parte de estas imágenes los rostros aparecen de frente y con una resolución media/alta. IntProy y EigenFeat alcanzan los mejores resultados en ambos conjuntos, con errores de precisión por debajo del 6 %. Los dos métodos destacan también en el grupo CMU/MIT, en el que las caras tienen en general menor tamaño. En las particiones TDT y DVD es más frecuente encontrar caras con grandes giros 3D, laterales o verticales. En esas circunstancias, el localizador basado en redes neuronales es el que consigue los menores errores. Por otro lado, en las caras de DVD aparecen típicamente sombras muy destacadas. Este hecho presenta problemas para muchas de las técnicas, y en concreto para IntProy. No obstante, siempre consigue una mejora sobre los datos del localizador nulo.

6. **Algunos aspectos relevantes de las implementaciones.**

   Una cuestión de implementación presente en todas las técnicas es la decisión de cómo manejar las *imágenes en color*. Como ya vimos en el problema de detección, utilizar uno u otro canal puede suponer una diferencia significativa en los resultados obtenidos.

Para estudiar la influencia del color en la localización, hemos repetido los experimentos de la tabla 4.2 seleccionando los canales R, G, B y el valor de intensidad de los píxeles. Los valores obtenidos se encuentran sintetizados en la tabla 4.6. Nótese que aquí se señalan en negrita los ganadores para cada columna, es decir, el canal más adecuado en cada técnica.

Canal	IntProy		NeuralNet		TemMatch		EigenFeat		Media	
usado	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
Rojo	**95,9**	**6,4**	74,5	6,8	**96,3**	**6,6**	**94,8**	**6,0**	**90,4**	**6,5**
Verde	95,8	6,6	69,3	6,6	93,3	7,3	91,1	7,0	87,4	6,9
Azul	92,3	7,7	52,6	7,5	88,9	8,6	85,3	8,5	79,8	8,1
Gris	95,6	6,5	**74,7**	**6,4**	94,5	6,9	92,5	6,6	89,3	6,6

Tabla 4.6: *Resultados de los localizadores sobre la base UMU en función del canal usado, en el caso de imágenes a color. Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

Claramente, el canal rojo resulta más adecuado en la mayoría de los casos. En promedio, produce un 7 % más de precisión que el verde, y un 25 % más que el azul. La segunda mejor elección es usar el valor de intensidad de los píxeles. Es más, el localizador basado en redes neuronales consigue mejores ratios con la intensidad que con el rojo. En consecuencia, todas las implementaciones usadas en los experimentos extraen el canal R de las imágenes RGB, excepto NeuralNet que realiza una conversión a escala de grises.

Otro aspecto que afecta a la mayoría de los métodos es el *tamaño de las imágenes procesadas*: las caras encontradas por el detector son extraídas primero a un tamaño fijo, sobre el que se busca después la posición de los componentes faciales. Esta imagen debe tener una resolución suficiente, aunque un valor mayor no siempre garantiza mejor localización. En TemMatch y EigenFeat el tamaño ha sido obtenido empíricamente, mediante diferentes ensayos, encontrando que el valor óptimo está alrededor de los 96×120 píxeles. Las resoluciones utilizadas aquí son mucho mayores que las aplicadas en la detección.

En el localizador basado en integrales proyectivas, la cuestión de la resolución está relacionada estrechamente con el tamaño de los modelos de proyección. Los valores más adecuados han sido estimados también a través de diferentes pruebas. En concreto, la proyección vertical de la cara, PV_{cara}, tiene 90 puntos, y la horizontal de los ojos, PH_{ojos}, 72 puntos. No obstante, el método es bastante robusto frente a este factor. Por ejemplo, si se usan modelos de 2/3 de ese tamaño, el error de precisión aumenta por debajo del 5 %. Y para tamaño 1/3 del óptimo[13] el incremento está sobre el 17 %.

Algunos otros parámetros han sido ajustados en los diferentes métodos, buscando un funcionamiento óptimo de todos ellos. Por ejemplo, en el localizador de componentes

[13] Es decir, PV_{cara} de 30 puntos y PH_{ojos} de 24, igual que los usados en detección.

basado en comparación de patrones se ha comprobado que la medida de correlación funciona mucho mejor que una suma de diferencias al cuadrado. Mientras que la primera consigue un 96,3 % de localización correcta y un 6,6 % de error de precisión en los ojos, la segunda baja hasta un ratio del 64,8 % con un error medio del 13,5 %. Claramente, la correlación es mucho más robusta frente a las sombras y las diferencias de intensidad en las imágenes.

Otro ajuste que ya hemos mencionado es el *tamaño de la base* en el localizador mediante autoespacios asociados a los componentes faciales. La tabla 4.7 presenta los resultados del método EigenFeat sobre la base UMU utilizando distinto número de autocomponentes para los ojos y la boca.

	Tamaño de la autobase							
Medida	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ratio loc.	68,0	92,9	94,8	94,8	93,7	92,5	90,5	88,2
Error ojos	12,4	6,9	6,3	6,0	6,2	6,4	6,7	7,1

Tabla 4.7: *Resultados del localizador basado en autoespacios sobre la base UMU, en función del tamaño de la base de autovectores usada.*

Los resultados indican una mejora rápida para los tamaños pequeños; el funcionamiento óptimo se produce para 4 auto-objetos. A partir de ahí, se evidencia un lento decrecimiento en la efectividad del localizador, que pierde entre 1 y 2 puntos en el ratio de localización por cada nuevo elemento añadido a la base. Estos autovectores con menor autovalor asociado representan modos de variación más específicos –menos generales– que los de mayor autovalor. Su forma está muy influida por los ejemplos de entrenamiento concretos, de lo cual se deduce una peor capacidad de generalización.

Finalmente, los resultados del localizador basado en contornos se encuentran a mucha distancia del resto. El problema no es únicamente una cuestión de implementación, sino que subyace al propio funcionamiento del proceso. Básicamente, podemos señalar dos inconvenientes: (1) en condiciones no triviales, la hipótesis de que las zonas más oscuras corresponden siempre a los componentes faciales puede dejar de ser aplicable; y (2) aun cumpliéndose la primera condición, el contorno es insuficiente para producir una localización precisa de los ojos y la boca.

4.4.4. Resultados de las pruebas sobre la base FERET

En este segundo caso se han utilizado 3816 caras de la base FERET [52], disponible públicamente. En concreto, tomamos todas las imágenes para las cuales existe un etiquetado manual disponible. Normalmente el etiquetado ofrecido es bastante fiable, aunque en algunos casos el error es apreciable a simple vista. En unas pocas imágenes las posiciones dadas son simplemente incorrectas. Estas últimas han sido corregidas antes de ejecutar las pruebas.

Igual que antes, la entrada de los localizadores son los resultados del detector Haar+IP.

El método es capaz de encontrar 3743 caras, es decir, el 98,1 % de las existentes. Todas ellas aparecen en primer plano, con ligera o nula inclinación, y una resolución más o menos buena. La mínima distancia interocular es de 39 píxeles, con un promedio de 67, de forma que la resolución de entrada no es un factor limitador. En unas 253 imágenes existe un importante giro lateral –mirada a izquierda o a derecha– de unos 22º, mientras que en el resto las personas miran de frente.

Para la ejecución de los localizadores se han utilizado los mismos ajustes que los realizados en el apartado previo[14]. De esta forma, si en el caso anterior vimos que las distintas técnicas eran afinadas para producir los resultados óptimos, en esta prueba se analiza el efecto de los mismos parámetros sobre un conjunto de test completamente diferente. Pretendemos así poner a prueba las capacidades de generalización de los diferentes algoritmos, y en particular la del método propuesto basado en integrales proyectivas.

Los resultados numéricos de los localizadores analizados se exponen en la tabla 4.8. La medida de precisión manual ha sido recalculada, teniendo en cuenta que la resolución de entrada es algo más del doble que en la base UMU.

Método	Ratio loc. (nº fallos)	Dif. ángulo	Dif. tamaño	Dif. posic.	Error ojo izq.	Error ojo der.	Error boca	Tiempo (ms)
Manual	100 % (0)	0,70º	1,4 %	1,4 %	1,6 %	1,9 %	2,3 %	–
Nulo	**99,5 % (18)**	2,43º	8,3 %	5,1 %	5,8 %	6,3 %	10,4 %	–
IntProy	98,9 % (41)	**0,95º**	8,7 %	**4,3 %**	**4,6 %**	4,6 %	**9,8 %**	3,1
NeuralNet	90,8 % (343)	1,41º	7,6 %	5,2 %	**4,6 %**	**4,3 %**	10,8 %	346,0
TemMatch	91,1 % (332)	2,03º	**7,3 %**	5,3 %	7,5 %	7,3 %	10,5 %	18,9
EigenFeat	93,9 % (272)	2,35º	9,2 %	5,8 %	6,5 %	6,0 %	11,6 %	45,1
Cont	82,4 % (657)	2,78º	11,2 %	8,5 %	9,6 %	10,5 %	15,4 %	**1,3**

Tabla 4.8: *Resultados de los distintos localizadores sobre la base FERET. La entrada son 3743 caras detectadas con Haar+IP. Para cada método, se muestran las métricas definidas en el apartado 4.4.2. Los errores están en proporción a la distancia interocular. Se señala en negrita el mejor resultado obtenido para cada parámetro estudiado (obviando el método manual). Los datos del ordenador usado aparecen en la tabla 3.2 (ver la página 131).*

En la figura 4.26 se pueden ver las densidades de puntos localizados en proporción a una cara media. Al existir un número mucho mayor de ejemplos, los gráficos permiten apreciar con mayor detalle el tipo de errores específicos de cada técnica.

Las curvas de distribución de los errores son mostradas en la figura 4.27.

Las figuras 4.28 y 4.29 contienen unos cuantos ejemplos representativos de los resultados obtenidos por las diferentes técnicas, desechando el localizador basado en contornos.

Algunos de los datos resultantes de este experimento vienen a confirmar las conclusiones extraídas para el conjunto UMU. Sin embargo, también existen ciertas discrepancias que merece la pena destacar. En cualquier caso, los valores obtenidos con FERET permiten realizar una valoración más fundamentada, gracias a dos hechos: el conjunto de imágenes es mucho

[14]La única excepción es el método EigenFeat, para el cual se utilizan 5 autocomponentes –uno más que antes–, lo que consigue bajar un 10 % su error de precisión. No obstante, estos auto-objetos son los calculados y aplicados en la base UMU; es decir, no hay un entrenamiento específico para las caras de FERET.

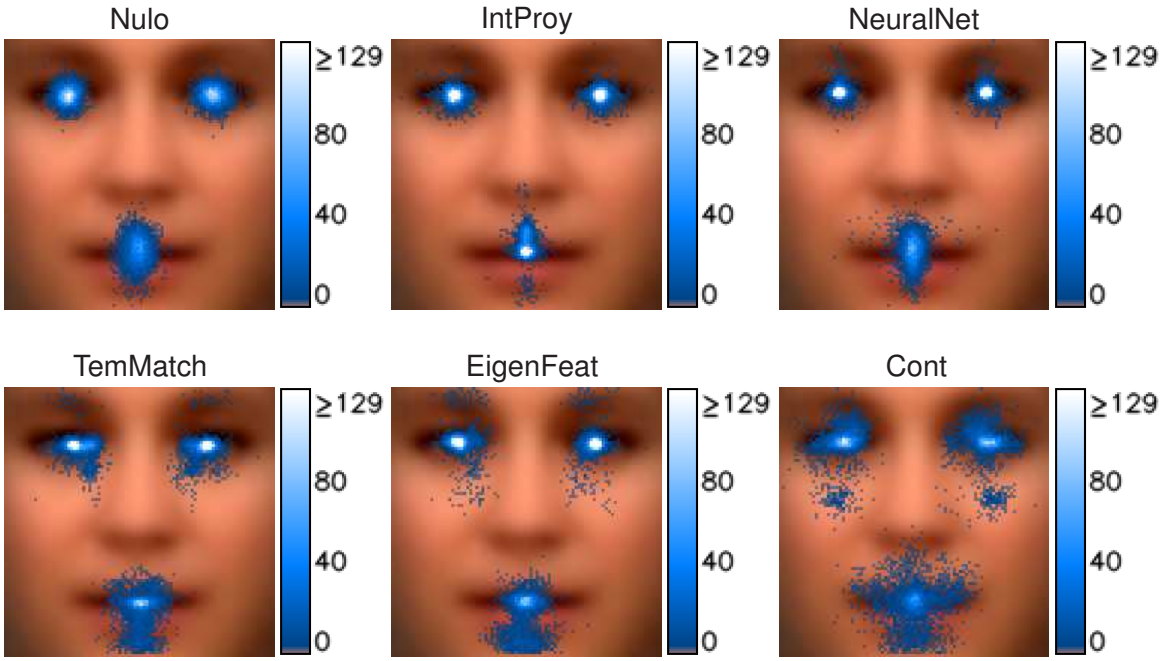


Figura 4.26: *Gráficos de localizaciones resultantes de los distintos métodos sobre la base FERET. Sobre una cara media estándar, se representan las densidades de puntos localizados por cada técnica en las distintas partes de la cara (ver la leyenda a la derecha)*

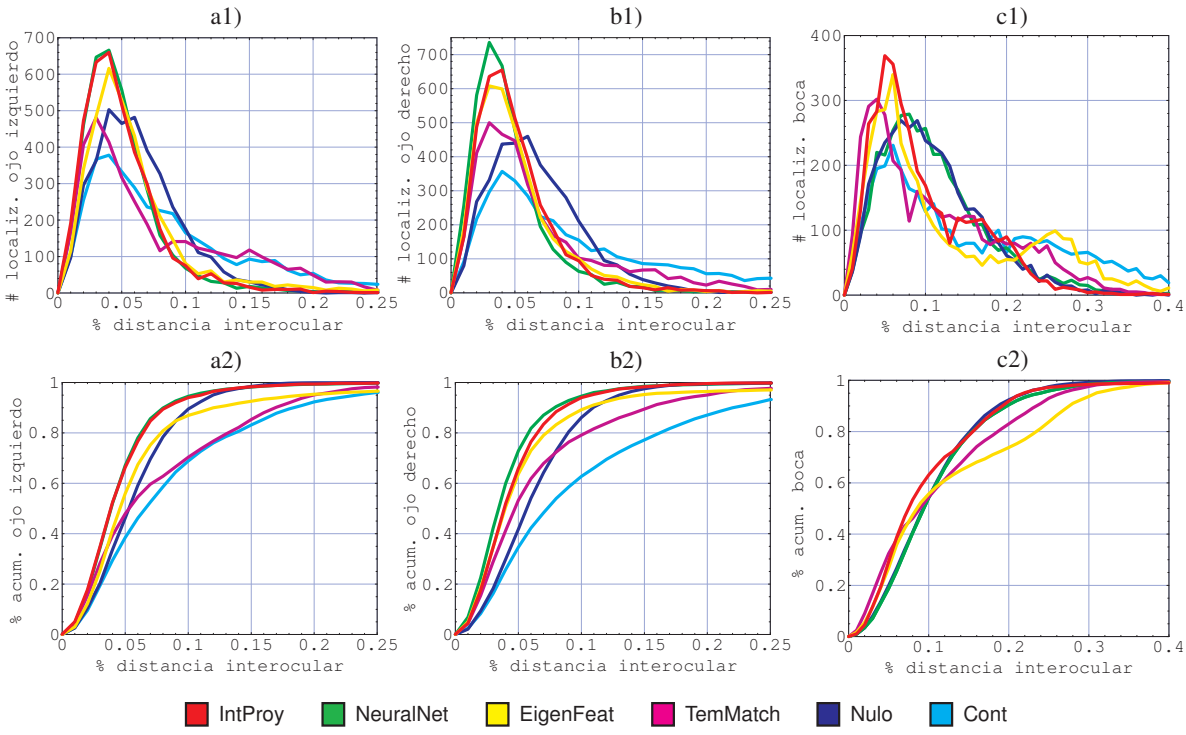


Figura 4.27: *Curvas de distribución de los errores de localización sobre la base FERET, para los distintos métodos. Las gráficas representan el número de casos para cada porcentaje de error (distancia entre la posición obtenida y la etiquetada, en proporción a la separación interocular). a1,b1,c1) Número absoluto de casos. a2,b2,c2) Porcentaje acumulado. a) Ojo izquierdo. b) Ojo derecho. c) Boca.*

mayor; y se evita el riesgo de sobreajuste de las técnicas, al no haber realizado un refinamiento específico de los parámetros. Igual que antes, vamos a discutir y valorar los resultados de la prueba, añadiendo algunos datos adicionales de interés.

1. **Valoración global de los métodos.**

   Al contrario que en el apartado 4.4.3, las diferencias entre los métodos son más acusadas, lo que permite establecer una mejor clasificación de los mismos. En conjunto, el localizador basado en proyecciones es el que consigue los mejores resultados: gana en 2 de los 3 parámetros globales, mantiene bajos los errores de los ojos y la boca, y requiere un tiempo de ejecución varias veces menor que sus competidores. El error en el ángulo estimado, de menos de 1º, se aproxima mucho al óptimo teórico y está a bastante distancia del siguiente mejor. En cuanto a la diferencia de la posición central y de la boca, es el único capaz de mejorar los resultados del localizador nulo. Y en relación al error en los ojos, se logra una precisión reducida y muy similar para ambos, que en términos absolutos está en torno a los 3 píxeles. De las gráficas de la figura 4.27 se deduce que más del 95 % de los ojos son localizados con un error menor al 10 % de la distancia interocular.

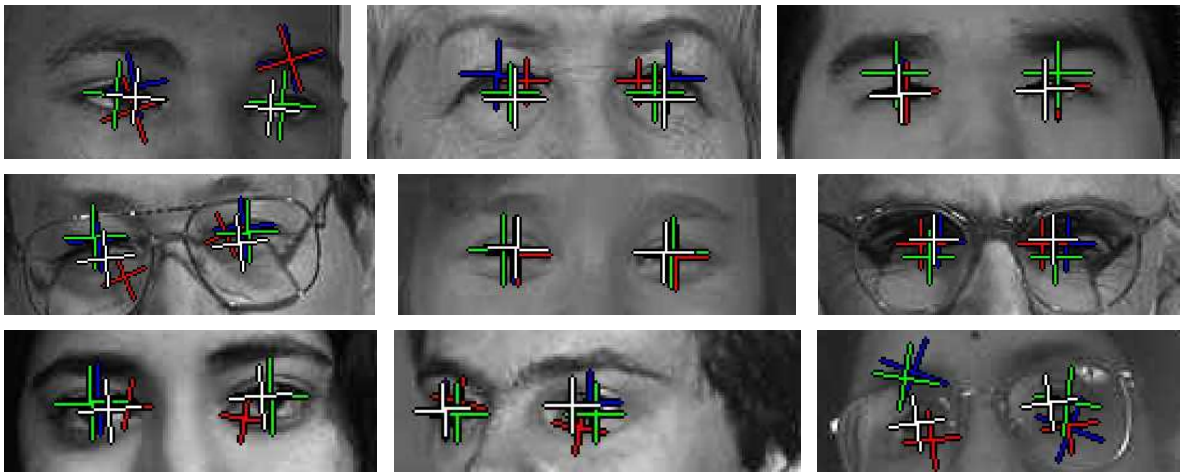


Figura 4.28: *Ejemplos de localizaciones de ojos resultantes para la base FERET. Se representan las posiciones resultantes del método basado en integrales proyectivas (en blanco), en redes neuronales (en verde), en autoespacios (en azul) y en búsqueda de patrones (en rojo).*

   El siguiente mejor método es NeuralNet, que produce los menores errores en la localización de los ojos. Gracias a ello, se consigue una buena precisión en los parámetros globales, aunque sin superar a otros métodos. Pero el gran inconveniente de esta alternativa es su complejidad computacional; el proceso consume 1/3 de segundo por cara, lo cual es 100 veces más tiempo que el localizador basado en proyecciones. Por otro lado, recordemos que el elevado error en la localización de la boca se debe a que la implementación disponible sólo busca los ojos, de manera que la posición de la boca se establece en relación a ellos. De los resultados se deduce que esta forma de situar la

boca exclusivamente en proporción a los ojos es bastante imprecisa.

En la tercera posición podemos situar el algoritmo basado en autoespacios modulares. Sus resultados son bastante parecidos a los conseguidos en la base UMU, mientras que IntProy y EigenFeat los mejoran de forma significativa. Podemos concluir que EigenFeat mantiene un buen comportamiento para ambos conjuntos, pero sin aprovechar la menor complejidad implícita en las imágenes de FERET (más resolución, y menos giros y sombras). Es especialmente problemática la localización de la boca, con casi un 12 % de error de precisión.

A pesar de ello, este experimento hace patente la mayor potencia de los auto-objetos frente a la simple comparación de patrones; el ratio de localización sube casi 3 puntos porcentuales, mientras que las distancias en los ojos bajan 1 punto. Podemos decir que TemMatch presenta una gran dificultad para distinguir los ojos cuando no aparecen claramente más oscuros que el resto de la cara. Por su parte, el localizador mediante contornos falla en muchas situaciones por la existencia de contornos espurios, debido a elementos faciales como gafas, barba, bigote y arrugas. Aunque sea el que tarda menos tiempo, sus resultados son inaplicables en la práctica.

Debemos destacar también el hecho de que el mejor ratio de localización lo consigue el método nulo, que falla sólo en 18 de las 3743 caras, frente a las 41 de IntProy y las más de 270 del resto. El primero de ellos se basa únicamente en los resultados de la detección facial que, aunque imprecisa en la posición exacta, produce localizaciones coherentes y muy ajustadas al rostro. Al intentar refinar la posición de los componentes, muchos métodos se desvían de la posición correcta, empeorando así los ratios de partida. Esto explica que IntProy y NeuralNet reduzcan el error en los ojos pero aumenten el número de fallos de localización. Además, sólo IntProy consigue mantener ese ratio en unos valores comparables a los del localizador nulo.

2. **Distribución de los errores.**

   Las densidades de puntos de la figura 4.26 reafirman muchas de las observaciones realizadas para la base UMU en relación a la distribución de los errores en las diferentes técnicas. En el método nulo, las nubes de puntos tienen formas muy regulares, que se podrían asimilar a gaussianas 2D. Las varianzas son muy parecidas en ambos ejes, excepto para la boca que presenta aproximadamente el doble de imprecisión (de varianza) en sentido vertical.

   En los métodos IntProy y NeuralNet las posiciones de los ojos adoptan también distribuciones similares a gaussianas, pero con mucha menor varianza; es decir, las nubes están mucho más concentradas en torno a la posición central de los ojos. En concreto, la varianza para los segundos es de aproximadamente 0,0014 en ambos ejes –medida sobre la distancia interocular–, mientras que en el localizador nulo está sobre 0,0023 (casi el doble). El inconveniente de ambos métodos es que son varios los casos donde los ojos

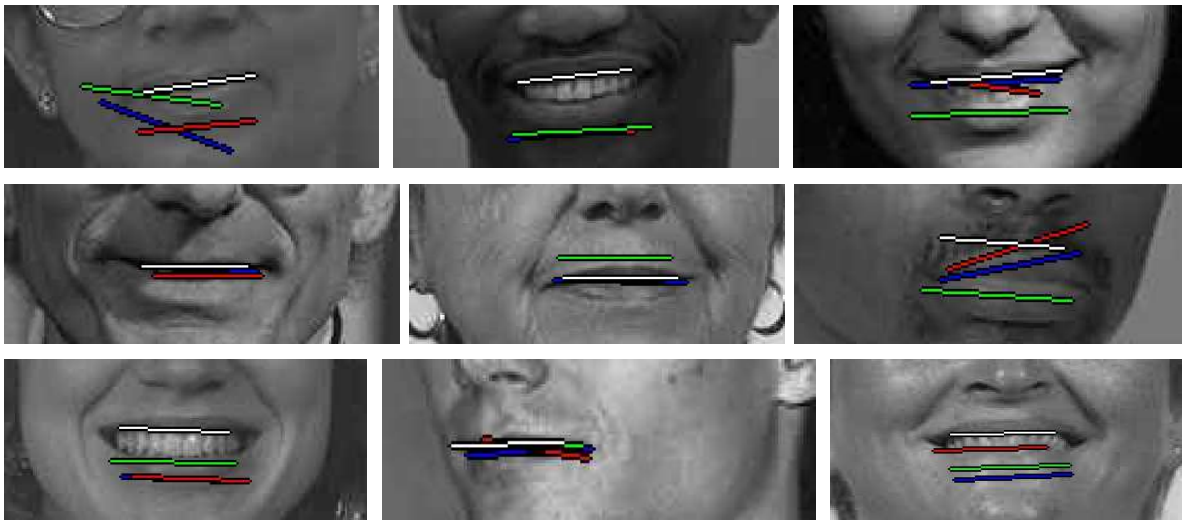


Figura 4.29: *Ejemplos de localizaciones de bocas resultantes para la base FERET. Se representan las posiciones resultantes del método basado en integrales proyectivas (en blanco), en redes neuronales (en verde), en autoespacios (en azul) y en búsqueda de patrones (en rojo).*

se sitúan por encima de las pestañas, como se puede ver en la figura 4.28.

En cuanto a la localización de los ojos en TemMatch y en EigenFeat, existen varios tipos de errores que hacen que se sitúen en el lagrimal, en los bordes de la nariz o, incluso, en las cejas. Los dos primeros problemas surgen normalmente cuando aparecen sombras en los ojos debidas a la propia cara, mientras que el tercero ocurre cuando los ojos no son más oscuros que el color de piel. Por otro lado, como se puede comprobar en el último caso de la figura 4.28, los reflejos en las gafas resultan bastante difíciles de manejar en muchos métodos.

La *localización de la boca* resulta más problemática en la mayoría de los algoritmos. Tanto los errores como las varianzas de las nubes de puntos son mayores que los de los ojos. IntProy es el método que consigue menor error, pero aun así existe una alta indeterminación en sentido vertical, que hace que la boca se sitúe a veces en el labio superior o en la posición de los orificios nasales. Por su parte, en los otros localizadores el error tiende a ocurrir más bien cerca de la barbilla.

La gráfica de densidades para el localizador basado en contornos demuestra una alta confusión en las posiciones de ojos y boca. La mayor resolución de las imágenes no ayuda a paliar los inconvenientes detectados en la base UMU.

3. **Precisión en función de factores demográficos.**

   Una característica interesante de la base FERET es la inclusión de información adicional sobre las imágenes disponibles, las condiciones de captura y los individuos que aparecen. Entre esta información podemos encontrar el sexo, la edad y el grupo étnico de cada persona. Creemos especialmente interesante analizar la precisión alcanzada por

los diferentes métodos de localización en función de este último factor, en cuanto a la influencia que pueden tener los rasgos de una raza en los resultados del proceso.

Para ello realizamos varias particiones del conjunto de imágenes de entrada. Los principales grupos étnicos distinguidos por FERET, y los ejemplos disponibles (y etiquetados) para cada uno, son los siguientes:

- **Asiático**: 326 caras.
- **Hindú**: 104 caras.
- **Hispano**: 110 caras.
- **Negro**: 152 caras.
- **Blanco**: 1227 caras.

No todas las imágenes están etiquetadas. Además, hemos suprimido algunos grupos minoritarios que contienen un número de casos poco significativo. Los resultados de la localización en función del grupo étnico se encuentran en la tabla 4.9.

Grupo	Nulo		IntProy		NeuralNet		TemMatch		EigenFeat	
étnico	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
Asiático	**98,5**	6,4	96,9	5,3	89,3	**4,9**	85,9	8,2	88,7	7,0
Hindú	**100**	6,3	**100**	6,0	93,2	**4,6**	86,4	8,3	90,3	6,7
Hispano	**100**	5,7	**100**	4,5	95,5	**4,2**	91,8	7,1	**100**	4,4
Negro	**99,3**	6,9	98,6	6,1	92,4	**4,4**	73,8	10,9	89	7,8
Blanco	**99,5**	6,0	99,1	4,9	94,7	**4,3**	96,4	6,0	96,2	5,2

Tabla 4.9: *Resultados de los localizadores sobre la base FERET en función del grupo étnico. Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

Las diferencias entre grupos son más significativas en algunos métodos que en otros. Así, en TemMatch y en EigenFeat los márgenes entre el mejor y el peor caso son de más de 11 puntos en el ratio de localización y de 2,5 puntos en los errores de precisión. Sin embargo, en NeuralNet todos los errores están entre 4,2 % y 4,9 %. El localizador basado en proyecciones se encuentra en una situación intermedia; existe una clara diferencia entre grupos, aunque no tan grande como en EigenFeat. En cualquier caso, consigue reducir los errores de precisión para todos los grupos.

En última instancia, la diferente efectividad de los métodos al cambiar de grupo étnico no es más que un reflejo de las condiciones de entrenamiento de las técnicas. Según los resultados obtenidos, la mayoría de los métodos disponibles se aproximan a los casos de los grupos Hispano y Blanco. Una consecuencia práctica es que, si se conoce la distribución demográfica de los sujetos procesados, un entrenamiento específico puede resultar muy adecuado. No obstante, sigue siendo factible utilizar unos mismos parámetros comunes.

Otros factores demográficos, como el sexo o la edad, parecen tener una influencia prácticamente despreciable en la efectividad de los algoritmos de localización. Por ejemplo, sobre sendas particiones de 798 mujeres y 1172 hombres en las imágenes de FERET, la precisión de localización en los ojos para IntProy es de 4,6 % y 4,7 %, respectivamente. También NeuralNet presenta una efectividad ligeramente mayor en el grupo femenino, de 4,4 % frente a 4,5 % para el masculino. El margen es algo mayor en los ratios de localización (por ejemplo, del 99,2 % al 98,4 % en IntProy). Claramente, todas estas diferencias se pueden considerar incluidas dentro de los errores de medición y muestreo.

4. **Algunos resultados adicionales.**

   La valoración de la *eficiencia computacional* es, lógicamente, la misma que para la base UMU. Al existir una extracción previa de las caras a un tamaño estándar, la influencia de la resolución original de las imágenes es prácticamente inexistente. En consecuencia, el localizador basado en proyecciones es el método más rápido, a excepción de Cont. TemMatch es más de 5 veces más lento, EigenFeat por encima de 10 veces más costoso, y NeuralNet unas 100 veces.

   También hemos realizado algunos experimentos, aunque de menor envergadura, variando el *algoritmo de detección de caras*. Estas pruebas tratan de medir la robustez frente a las imprecisiones en la entrada de los localizadores, es decir, la colocación inicial del rectángulo contenedor de cara. En general, los errores de localización crecen lentamente al aumentar el error del localizador nulo; y, comparativamente, las diferencias entre métodos se mantienen.

   En la figura 4.30 se pueden ver los resultados de una prueba –en este caso sobre la base UMU– donde se han modificado aleatoriamente las posiciones del localizador nulo, con el único fin de aumentar la imprecisión de entrada. Aun así, el localizador IntProy es capaz de encontrar correctamente los ojos y las bocas.

   Las variaciones del detector incluyen también la aplicación del método Haar+IP reduciendo previamente la resolución de entrada en varios órdenes. Esto es, la imagen original se reduce por n (para n= 2, 4 y 6), se aplica el detector y se multiplican las posiciones resultantes por n. Sobre los rectángulos de cara obtenidos se ejecutan los diferentes algoritmos de localización en las imágenes originales.

   En la tabla 4.10 se indican los resultados para las 3743 caras de la base FERET. Normalmente la imprecisión de entrada es mayor al aumentar la reducción de las imágenes, aunque de manera esporádica puede suceder lo contrario, como ocurre para tamaño 6.

   La robustez de los métodos frente a la imprecisión de entrada es lo más destacable de estos datos. Es especialmente buena en cuanto a los errores en los ojos para el método NeuralNet, y en relación al ratio de localización para IntProy. Podemos concluir que para una buena localización de los componentes faciales no es necesario aplicar el algoritmo de detección con la resolución original. Es posible –y recomendable si se requiere

Figura 4.30: *Localización de componentes con una elevada imprecisión de entrada. Se representan las posiciones resultantes del método basado en integrales proyectivas (en blanco) y las posiciones de partida (en azul claro). Las segundas se han obtenido modificando aleatoriamente los resultados del detector nulo.*

Reducción	Nulo		IntProy		NeuralNet		TemMatch		EigenFeat	
detector	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos	rloc	eojos
1	**99,5**	6,0	98,9	4,6	91,0	**4,4**	91,4	7,3	93,5	6,0
2	**99,3**	6,2	98,7	4,7	91,2	**4,5**	91,4	7,2	92,9	6,0
4	96,6	7,8	**97,7**	5,5	93,0	**4,4**	91,0	7,6	93,7	6,2
6	98,6	7,5	**99,3**	4,7	95,1	**4,3**	93,6	6,6	95,8	5,5

Tabla 4.10: *Resultados de los localizadores sobre la base FERET en función de la precisión de detección. El detector se aplica reduciendo las imágenes por los factores 1, 2, 4 y 6. Para cada método, se muestra: rloc: ratio de localización correcta; eojos: error promedio en la localización de los ojos. Los errores están en proporción a la distancia interocular.*

alta eficiencia computacional– aplicar una reducción previa de las imágenes. El localizador será el encargado de refinar después las posiciones obtenidas.

4.5. Conclusiones y valoraciones finales

Analizando conjuntamente los resultados de los dos experimentos descritos en los apartados 4.4.3 y 4.4.4, podemos ver que el método de localización mediante proyecciones ofrece una **relación precisión/coste** insuperable: consigue una excelente localización global de la cara, con una estimación del ángulo próxima al etiquetado manual; los ratios de localización correcta están siempre entre los mejores, con el 96 % para los casos complejos de UMU, y el 99 % para FERET; el error medio en los ojos no supera los 5 mm (los 3,2 mm en FERET); y el tiempo de ejecución está entre 2 y 3 milisegundos. Para un tope de distancia del 10 %, se localizan correctamente el 85 % de los ojos en UMU y el 94 % en FERET. Estos resultados se comparan muy bien con muchos métodos del estado del arte analizados en la sección 4.2, aunque hay que tener en cuenta que no son sobre los mismos datos.

Los errores tan reducidos en la **inclinación de la cara** son un reflejo de la robustez del método propuesto para su estimación. Tradicionalmente, la simetría del rostro ha sido poco explotada, por la dificultad de garantizar que la simetría de estructura da lugar a una simetría de apariencia. Las proyecciones nos han permitido crear un mecanismo que aprovecha de forma robusta esta propiedad natural de las caras, capaz de trabajar con sombras desiguales, giros y elementos faciales que dificultan el problema.

Otra de las ventajas del localizador propuesto es el **procesamiento holístico** de la cara, frente a las técnicas que buscan ojos, narices y bocas de forma independiente. El análisis de las proyecciones como un todo ofrece una gran invarianza frente a situaciones complejas, como la oclusión o la deformación extrema de un componente facial. Así, prácticamente nunca llega a ocurrir que ambos ojos se sitúen en las cejas, como suele ser habitual en otros algoritmos basados en proyecciones. En consecuencia, podemos decir que la técnica desarrollada está en la categoría de las *basadas en apariencia*, aunque haga uso de las proyecciones.

La elevada eficiencia computacional del método se deriva del manejo de señales unidimensionales, y de la efectiva **separación de los 5 grados de libertad** del problema en las tres etapas definidas: el primer paso resuelve la inclinación (1 grado); el segundo la posición y escala vertical (2 grados); y el tercero lo hace en sentido horizontal (2 grados).

Siendo más exigentes, pensamos que existen algunos aspectos que podrían mejorarse:

- Por un lado, aunque el **error en la boca** sea menor que en los restantes métodos, sigue siendo relativamente alto. Podría ser interesante estudiar la aplicación de otros mecanismos, como la proyección de bordes o de la varianza, para mejorar su localización.

- Por otro lado, hemos observado que existe cierta tendencia a **situar los ojos en el lagrimal**, sobre todo cuando están cerrados. El cambio de apariencia que ocurre en esas

situaciones hace que el simple modelo de proyección media no se adapte bien a todos los casos. Una posible solución sería utilizar varios modelos para PH_{ojos} obtenidos, por ejemplo, con un algoritmo de k medias sobre las proyecciones de entrenamiento. En la localización se aplicaría el alineamiento de la instancia actual con cada uno de estos modelos de proyección, utilizando el que menor distancia produjera. De esta forma se aprovecharía siempre el modelo más parecido a la apariencia actual. En principio, un número reducido de ellos –en torno a unos 5– podría bastar para describir la mayoría de las situaciones posibles, sin aumentar sensiblemente el coste del proceso.

4.6. Resumen

Las integrales proyectivas ofrecen una valiosa ayuda en la resolución del problema de localización de componentes faciales, como hemos discutido en la revisión del estado del arte. En general, son robustas frente a fuentes de variación como la expresión facial, los factores individuales y la pose, permiten alcanzar una elevada precisión en las posiciones obtenidas, y todo ello con un reducido coste computacional.

El acercamiento que hemos desarrollado en este capítulo presenta diferencias sustanciales en relación a muchas de las propuestas previas basadas también en proyecciones:

- La localización de los componentes se fundamenta en los algoritmos de **alineamiento de patrones unidimensionales**, frente a los métodos heurísticos basados en la búsqueda de picos o zonas de variación máxima de las señales. De esta manera, el resultado final no depende de características puntuales sino de la forma global de las proyecciones. Esto da como resultado una robustez mucho mayor en condiciones no triviales.

- Los **modelos de proyección** usados en el alineamiento se obtienen mediante **entrenamiento** a partir de un conjunto de ejemplos. Se evita así la aplicación del conocimiento "experto" del implementador, que se limita a establecer las regiones que deben ser proyectadas. Además, esta filosofía de trabajo posibilita una mejor adaptación a problemas similares con otros tipos de objetos.

- De forma resumida, el proceso de localización propuesto consta de **tres grandes pasos**. En primer lugar, usando la proyección vertical de ambos ojos, se estima la inclinación del rostro. A continuación, se aplica la proyección vertical de la cara para refinar su posición y escala en el eje Y. Por último, se proyecta horizontalmente la región asociada a los ojos, lo que permite obtener la localización de los componentes en el eje X.

- Los experimentos presentados han demostrado una **excelente fiabilidad** del método frente a otras alternativas comparables. El alineamiento de proyecciones consigue mejores resultados que la simple búsqueda de patrones y que la localización basada en autoespacios. La precisión es similar a la obtenida en la técnica que usa redes neuronales, aunque con un coste computacional dos órdenes de magnitud inferior.

- Lógicamente, existe un **margen de mejora** del método, que deja abiertas algunas cuestiones. Una de ellas es la utilización de otro tipo de proyecciones. Por ejemplo, la proyección asociada a la boca podría ayudar a mejorar su localización, evitando confusiones y localizaciones erróneas en la nariz o en el labio inferior. En ese caso, la utilización de imágenes de bordes parece resultar más adecuada que la simple proyección de la intensidad. Otra cuestión aún pendiente es el modelado de las sombras y la compensación de su influencia en la proyecciones obtenidas.

CAPÍTULO 5

"La Scapigliata", Leonardo da Vinci, c. 1508

Seguimiento de Caras en Vídeo

"Nunca olvido una cara. Pero con usted estaré encantado de hacer una excepción."
GROUCHO MARX

Hasta ahora hemos trabajado exclusivamente con imágenes estáticas, abordando los problemas de detección de caras y localización de componentes faciales. Sin embargo, existe una amplia variedad de aplicaciones que requieren el manejo de fuentes de vídeo en el procesamiento de las caras. El ejemplo más claro es un interface perceptual, en el que un usuario interactúa con la máquina a través de los movimientos, giros y gestos de su rostro. Por tanto, lo interesante aquí no es tanto el estado en un instante dado, sino el análisis de las variaciones de posición y forma a lo largo del tiempo. Los sistemas de actores virtuales, la monitorización, la videovigilancia y la codificación de vídeo son otras posibles aplicaciones que requieren un seguimiento rápido y fiable de las caras que aparecen en una secuencia de vídeo.

Incluso algunas aplicaciones que, en principio, manejan imágenes estáticas, podrían también aprovechar las ventajas de la dimensión temporal que aporta el vídeo. Un ejemplo destacable es el creciente interés por mejorar los sistemas biométricos de reconocimiento facial mediante el uso de vídeo [70], [108, capítulo 8].

La mayoría de las técnicas clásicas de seguimiento están orientadas al análisis de objetos rígidos bajo cambios de posición y orientación en un espacio 3D. La particularidad del objeto *cara* es su estructura flexible y deformable, que se evidencia en la variedad de expresiones faciales posibles. Otra característica propia es la elevada impredecibilidad de su movimiento, sólo sujeto a la voluntad del individuo. Por ello, podemos afirmar que el seguimiento rápido y fiable en situaciones complejas es un problema aún abierto.

En consecuencia, son diversos los objetivos que debe perseguir un buen seguidor de caras: robustez frente a cambios de expresión, iluminación y pose –posición y orientación 3D–, precisión en la localización de la cara, eficiencia computacional, y capacidad de determinar y

recuperarse de las situaciones de pérdida del seguimiento. El ámbito de aplicación será el que determine la mayor o menor importancia de cada criterio.

Las integrales proyectivas permiten construir un proceso de seguimiento rápido, preciso y robusto frente a expresiones faciales, como iremos viendo a lo largo de este capítulo. Empezamos describiendo las características y objetivos del problema dentro de la sección 5.1. Después hacemos un breve repaso de los acercamientos existentes en la sección 5.2. En la sección 5.3 desarrollamos el método de seguimiento mediante proyecciones; veremos que la solución propuesta presenta muchas similitudes con el proceso diseñado para la localización de componentes faciales –similitudes que se derivan, de forma lógica, de la estrecha relación entre ambos problemas–. Uno de los inconvenientes de muchos sistemas existentes es la adaptación frente a movimientos rápidos de la cara. Exponemos una forma de resolverlo usando un mecanismo de predicción basado en color. En la sección 5.4 se documentan algunos experimentos de la técnica desarrollada, comparándola con otros seguidores de caras disponibles bajo diferentes condiciones de uso. La conclusiones de estas pruebas se pueden encontrar en la sección 5.5. Por último, en la sección 5.6 se hace un resumen de las aportaciones más relevantes de este capítulo.

5.1. El problema de seguimiento de caras humanas

Una de las habilidades más sorprendentes de los niños recién nacidos –cuando aún no superan unas semanas de vida– es la capacidad para seguir caras humanas en movimiento [165]. Esta destreza destaca por encima de la pericia para seguir otros tipos de objetos. En este sentido, podemos decir que el seguimiento de caras es una *capacidad instintiva e innata* de los humanos. Este hecho podría tender una justificación biológica en la necesidad de los pequeños de reconocer a sus progenitores de entre un grupo amplio de individuos [165].

En el ámbito de la visión artificial, el seguimiento de caras sólo empezó a cobrar interés con la aparición de los *sistemas de interacción visual* con humanos [136]. Mayoritariamente, las propuestas que se han planteado están centradas de forma particular en el objeto cara, siendo especializaciones de mecanismos generales y, por lo tanto, de difícil aplicación a otros tipos de objetos. Pero esta limitación surge de la propia esencia del problema: las técnicas de seguimiento que se usan, por ejemplo, en procesos industriales o en navegación de robots no se pueden aplicar directamente en el seguimiento de rostros, y viceversa. En el apartado 5.1.3 discutimos algunas de las características que marcan esta diferenciación. Pero antes vamos a describir los pasos genéricos que constituyen un sistema de seguimiento facial.

5.1.1. Componentes de un sistema de seguimiento de caras

En la figura 5.1 se representan gráficamente los grandes bloques de un sistema de seguimiento de caras, sin entrar en técnicas concretas. Realmente, se trata de un esquema genérico similar al que se podría aplicar a cualquier otro tipo de objetos.

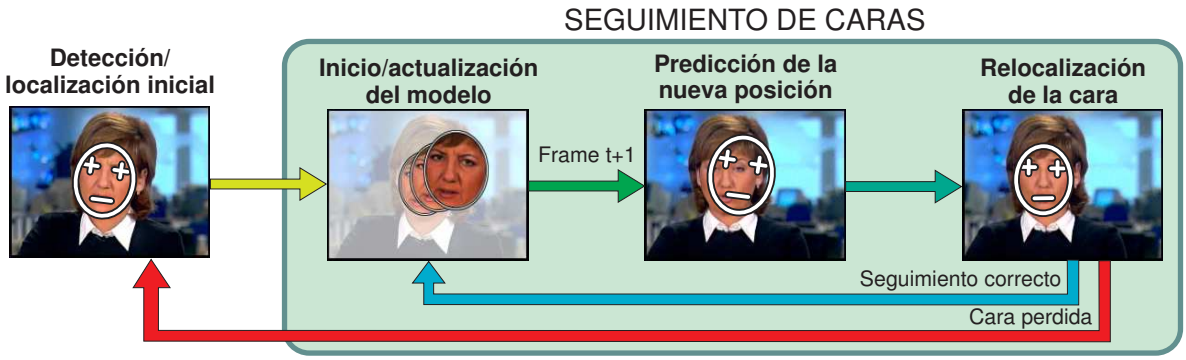


Figura 5.1: *Esquema global de un sistema genérico de seguimiento de caras. En primer lugar se aplica un detector de caras, y opcionalmente un localizador. Existe un modelo de seguimiento, que se inicializa y se actualiza en cada paso. Apoyándose en el modelo, se predice la posición esperada en el siguiente frame del vídeo. Partiendo de esa predicción, se lleva a cabo la relocalización de la cara. Si no se ha podido completar este último paso, se vuelve a la detección inicial.*

Como se muestra en la figura 5.1, el seguimiento es un proceso iterativo realimentado por las sucesivas imágenes (o *frames*) que constituyen la secuencia de vídeo. A la entrada del sistema se sitúa el algoritmo de **detección** de caras, encargado de encontrar las instancias a ser seguidas. En principio, un seguimiento preciso implica también aplicar un **localizador** de componentes faciales, como los que acabamos de estudiar en el capítulo 4. Los resultados de ambos pasos son la entrada para el seguimiento propiamente dicho. Aunque normalmente no aparecerá más de una cara –esto es lo más frecuente con vídeo–, en posible que se detecten varias, y el sistema debe ser capaz de procesarlas por separado.

Pasos del proceso genérico de seguimiento

A grandes rasgos, el seguimiento de caras se puede entender como una *relocalización continua* del rostro y sus componentes a lo largo del vídeo. Pero, en relación al problema de localización en imágenes estáticas, existe una dimensión temporal que se debe tener en cuenta. Es decir, la búsqueda de las caras no es independiente en los distintos *frames* de la secuencia, sino que debe existir información sobre las caras que están siendo seguidas, su posición, orientación, forma, velocidad, etc., en esencia, la "historia" de cada instancia a lo largo del tiempo. Esto es lo que hemos denominado como el *modelo de seguimiento* en la figura 5.1.

Usando esa información, se lleva a cabo una **predicción** de las posiciones esperadas de las caras en cada nueva imagen de la secuencia, esto es, una estimación basada en las posiciones y velocidades de instantes anteriores[1]. El método clásico de predicción son los filtros de Kalman [92]; en el apartado 5.3.2 analizamos un posible uso de los mismos, y proponemos una técnica alternativa basada en color.

Finalmente, partiendo de los valores predichos, el último paso del seguimiento consiste en **relocalizar** el rostro en el nuevo *frame*. Obviamente, cuanto más acertada sea la predicción,

[1]Las predicciones pueden ser los valores esperados para el modelo (posición, forma, orientación, etc.), o bien intervalos de mayor o menor tamaño, en función de la incertidumbre estimada.

más sencillo será el problema. Se puede apreciar claramente una gran similitud con el problema de localización en imágenes estáticas: dada una región "próxima" de cara, refinar la posición de la misma y de sus componentes. Con imágenes individuales, el grado de proximidad depende de los resultados de la detección, mientras que en vídeo está en función del movimiento del sujeto y la exactitud de la predicción. Típicamente, el segundo deberá manejar mayores discrepancias que el primero. Además, la eficiencia se convierte en un factor crítico, ya que el requisito de tiempo real suele ser básico.

Control del proceso y políticas de seguimiento

En cierto sentido, el seguimiento no es un simple *algoritmo* –con una ejecución limitada en el tiempo–, sino que es un proceso que se repite indefinidamente hasta que termina el vídeo o hasta que es requerido por el usuario. Por lo tanto, debe existir un elemento de nivel superior, encargado de controlar los diferentes *componentes del sistema*: detector de caras, localizador de componentes faciales, predictor, relocalizador, etc.

Este **controlador** es el responsable de llevar a cabo la secuencia lógica de operaciones en el funcionamiento normal del proceso, y debe implementar las diferentes políticas de seguimiento, es decir, determinar cuándo se pierde una cara, qué hacer en tal caso, cuántas instancias se pueden seguir como máximo, cómo encontrar caras nuevas, etc. Muchas de estas políticas dependen de las necesidades específicas de cada aplicación, de manera que el controlador debería admitir diferentes modos de trabajo.

5.1.2. Definición del problema y modelos de seguimiento

Una vez vistos los principales componentes y el modo de trabajo típico de los sistemas de seguimiento facial, vamos a proponer una posible definición para el problema.

Definición 5.1 ***Seguimiento de caras.***

Dada una secuencia de vídeo, el objetivo del seguimiento de caras es encontrar los rostros que aparecen en el vídeo y actualizar una descripción de cada uno de ellos a lo largo del tiempo, que puede incluir posición, orientación 3D, velocidad, deformaciones y el instante de aparición y desaparición de cada instancia de cara en la secuencia.

Nótese que hemos dejado cierta ambigüedad en cuanto a la información resultante del seguimiento, o lo que es lo mismo, el *modelo de seguimiento* aplicado. Grosso modo, podemos distinguir entre los métodos que hacen seguimiento 2D –con estimación de escala y posición de la cara en la imagen–, y los seguidores 3D –que resuelven los 6 grados de libertad asociados a la pose de la cabeza–. Entre ellos se sitúan los llamados seguidores 2,5D, similares a los 2D pero añadiendo algún tipo de información sobre la orientación del rostro [180].

Pero, realmente, la variedad de modelos que se han aplicado en el seguimiento facial no acaba en esa simple clasificación. Como veremos en la sección 5.2, existe toda una batería de métodos en cuanto al nivel de detalle con el que se describen los rostros: desde los sistemas

que tratan la cara como una simple elipse, hasta los que ajustan una malla deformable 3D, pasando por los que siguen un conjunto disperso de puntos característicos. En la figura 5.2 se puede ver una pequeña muestra de diferentes trabajos y sistemas comerciales, ilustrando cómo enfoques tan diferentes se incluyen en el seguimiento de caras.

Figura 5.2: *Distintos modelos para el seguimiento de caras, cada uno con una información diferente. a) Seguimiento mediante el algoritmo CamShift [35] (elipse contenedora de la cara). b) Real-Time FaceDetect Demo [53] (posición global de la cara, sin estimación de la inclinación). c) Seguimiento basado en apariencia [20, 19] (rectángulos de ojos y boca). d) Seguimiento en el interface perceptual "Tierra Inhóspita" [61] (posición central de ojos y boca). e) Software de seguimiento de caras del sistema OrbitCam de Creative Labs Ltd. (seguimiento de 22 puntos clave). f) Seguimiento con sistemas de partículas [1, 44] (modelo de malla 3D deformable).*

La elección del modelo de seguimiento no es una simple cuestión auxiliar, sino que marca el diseño, la aplicabilidad y los resultados del método.

- **Técnicas subyacentes.** Según el modelo empleado, tendrá más sentido utilizar unos tipos de técnicas de visión u otros. Por ejemplo, en un sistema que modela la cara como una elipse puede ser interesante usar *color*; pero en uno detallado será menos viable. En un modelo basado en puntos característicos se podría aplicar *búsqueda de patrones*; mientras que en los modelos de mallas puede ser conveniente manejar técnicas basadas en *flujo óptico*. En definitiva, cuanto mayor es el grado de detalle, las características usadas deben ser más locales.

- **Resultados de los seguidores.** El rendimiento de los sistemas de seguimiento también se ve influido por el tipo de modelo de cara en el que se apoyan. En general, los mo-

delos globales ofrecen mayor robustez y mejor adaptación a movimientos rápidos de la cabeza. Sin embargo, la precisión en más pobre y se requieren otros procesos para extraer información de la expresión facial. Por su parte, los métodos que ofrecen más detalle necesitan trabajar con mayores resoluciones; las caras deben cambiar con relativa suavidad y se debe evitar el ruido y el desenfoque. Además, suelen requerir más entrenamiento para un funcionamiento óptimo.

- **Aplicación del sistema.** Por último, el modelo debe ser elegido en función de la aplicación final. Para un interface perceptual puede ser suficiente con conocer la posición global de la cara. Podríamos decir lo mismo de muchas aplicaciones de análisis multimedia, monitorización, vídeo-vigilancia y entretenimiento, donde la resolución no siempre está garantizada. Los modelos intermedios pueden ser interesantes para el reconocimiento biométrico basado en vídeo, y para videoconferencia (por ejemplo, en generación de avatares). En aplicaciones de análisis de expresiones faciales para animación y sistemas de actores virtuales, los modelos de malla ofrecen mejores resultados, ya que aportan la información necesaria para esos usos.

En nuestro caso, el método de seguimiento que desarrollamos en la sección 5.3 entraría dentro de los seguidores 2,5D. En concreto, el objetivo es **localizar la posición de los dos ojos y la boca** a lo largo del vídeo. En el capítulo 7 veremos algunas aplicaciones de la técnica propuesta en estimación de pose, interfaces perceptuales y análisis de expresiones faciales.

5.1.3. Desafíos y dificultades en el problema de seguimiento

El seguimiento de caras humanas debe enfrentarse a problemas muy diferentes de los que surgen típicamente en control visual de procesos industriales, navegación de robots o monitorización de tráfico, por poner algunos ejemplos relacionados. La característica más definitoria es que existirá un objeto de interés –o unos pocos–, claramente destacado del fondo, con estructura 3D compleja, deformable y que puede moverse de manera impredecible. A diferencia de los otros ejemplos de seguimiento, no se puede prever a priori una dirección del movimiento, ya que la posición de la cara está sujeta a la voluntad del individuo.

Muchos de los grandes desafíos son comunes a la detección y localización, y han sido ya identificados y descritos en los capítulos 3 y 4. Vamos a destacar ahora las cuestiones específicas y más relevantes que se deben tratar en el seguimiento de caras humanas.

- **Movimiento de las caras.** Como acabamos de señalar, una de las principales propiedades del movimiento del rostro es su carácter altamente impredecible. En condiciones típicas, las variaciones de posición serán muy lentas, y normalmente tendrán una acotación espacial (por ejemplo, si el sujeto está sentado en una silla). Pero el cambio puede ocurrir de forma repentina. Un gesto de mirar a uno u otro lado puede tardar unas pocas décimas de segundo, de manera que aunque la velocidad no sea muy alta, la aceleración puede ser grande.

- **Desenfoque por movimiento.** En condiciones de laboratorio, usando una iluminación adecuada y cámaras de alta resolución, es improbable que aparezca este problema. Sin embargo, en aplicaciones orientadas a uso público –como sistemas de videoconferencia e interfaces perceptuales–, se utilizarán cámaras de bajo coste y las condiciones de iluminación pueden no ser las óptimas. En esas circunstancias, el efecto del desenfoque por movimiento puede afectar seriamente a la calidad de las imágenes con las que se trabaja. En la figura 5.3 se pueden ver algunos ejemplos de caras extraídas de una secuencia con movimiento; el emborronamiento de las imágenes es bastante visible. Otro problema que puede ocurrir es el entrelazado de vídeo que, idealmente, debería ser eliminado a priori en una etapa de preprocesamiento.

Figura 5.3: *Desenfoque por movimiento en una secuencia de prueba. Se muestran extractos de imágenes de la secuencia "ggm3.avi". La resolución original es de* 320×240 *píxeles a 15fps.*

- **Oclusión y desaparición.** Son varias las situaciones en las que una cara puede dejar de verse, de forma total o parcial, en una secuencia de vídeo. La más común es la salida por uno de los extremos de la imagen. También puede ocurrir oclusión de elementos externos, como la mano del propio sujeto. En la figura 5.4 se muestran varios ejemplos de estas situaciones en una secuencia de la base pública de vídeos faciales NRC-ITT[2] [70]. Una característica del vídeo es que las caras que dejan de verse pueden volver a aparecer; el sistema podría identificar esta reaparición o considerar los distintos fragmentos de forma independiente.

Figura 5.4: *Oclusión parcial de la cara en una secuencia de prueba. Se muestran algunas imágenes completas del vídeo "00-2.avi" de la base de vídeos faciales NRC-ITT [70, 69]. La resolución original es de* 160×120 *píxeles a 20fps.*

[2]Esta base de vídeos es utilizada originalmente para reconocimiento de personas en vídeo, y se ofrece de forma gratuita. Obsérvese, sin embargo, que la resolución y la calidad de las imágenes son muy limitadas.

- **Posición y orientación 3D.** Igual que en los problemas de detección y localización, la orientación de las caras –como objetos tridimensionales que son– variará su apariencia de forma sustancial. La mayoría de las técnicas de seguimiento existentes suponen que el individuo mira de frente a la cámara, y admiten ciertos márgenes de inclinación y giro lateral o vertical. En condiciones extremas –a menos que se haya tratado de forma explícita esta cuestión–, algunas técnicas producen resultados poco fiables, y otras simplemente pierden el seguimiento.

- **Cambios de iluminación y expresión facial.** Nuevamente, estos factores son comunes al resto de problemas con caras. En el caso del vídeo, lo importante es lograr una adaptación continua del proceso de seguimiento. En relación a la iluminación, puede ocurrir un cambio de matices de color, por ejemplo, debido a una modificación en el balance de blancos. Por otro lado, volverá a estar presente el problema de las expresiones faciales, como se observa en la figura 5.5.

Figura 5.5: *Expresiones faciales en una secuencia de prueba. Extractos de imágenes del vídeo "a3-08.avi". El vídeo es un fragmento de un programa de noticias capturado de televisión analógica. La resolución original es de* 640×480 *píxeles a 25fps.*

La secuencia mostrada en la figura 5.5 ha sido extraída de un programa de noticias, al igual que otras usadas en los experimentos. La ventaja de usar captura de televisión es que las expresiones, gestos y movimientos son más naturales, a diferencia de las condiciones de laboratorio, donde se observa típicamente más artificiosidad.

- **Tiempo real.** La eficiencia computacional adquiere un papel primordial cuando se trata de manejar fuentes de vídeo. Aunque en la experimentación se trabaje con secuencias grabadas, en condiciones normales de uso se llevará a cabo un procesamiento en tiempo real. Evidentemente, esto impone fuertes restricciones sobre la complejidad computacional de los algoritmos de seguimiento.

En relación al procesamiento de caras humanas en imágenes estáticas, el seguimiento parte de un estado inicial que debe ser aprovechado para la relocalización posterior. La cuestión fundamental es cómo modelar –o conseguir invariancia frente a– las variaciones sobre esa apariencia inicial ocasionadas por los factores señalados.

5.1.4. Criterios y medidas para la evaluación del seguimiento

La bondad de una técnica concreta de seguimiento está en función de su *eficiencia*, *precisión* y *robustez* frente a los factores descritos en el apartado anterior. Desafortunadamente, no existe un consenso claro sobre la mejor forma de evaluar y comparar los resultados de diferentes seguidores, como ocurre con los problemas de detección y localización.

En parte, una buena razón para ello es que las condiciones de experimentación varían mucho según los tipos de modelos utilizados. En los métodos globales el objetivo primordial es evitar la pérdida del seguimiento. En los detallados se busca una adaptación precisa de la malla o de los puntos característicos de la cara; y muchas veces la evaluación es meramente subjetiva. La situación se complica porque algunos trabajos suponen una inicialización manual del seguimiento [108, capítulo 4].

Centrándonos en la evaluación de un seguidor facial basado en la localización de ojos y boca, proponemos las siguientes métricas de bondad haciendo una analogía con las medidas de rendimiento introducidas en detección y localización:

- **Ratio de caras seguidas.** Esta medida se puede entender como una aplicación al vídeo del ratio de detección. Es decir, es equivalente a considerar por separado todas las imágenes de la secuencia y aplicar el criterio del detector. En concreto, sea n_{frames} el número de imágenes del vídeo; n_{caras} el número total de caras que aparecen –contadas imagen por imagen–; y $\{s_1, s_2, \ldots, s_k\}$ las localizaciones resultantes del seguidor para todos los *frames*. Cada s_i se clasifica como correcta o incorrecta, según el criterio de distancia máxima a los ojos del etiquetado manual[3]; sean r_{caras} el número de casos correctos, y $r_{nocaras}$ el de incorrectos. El *ratio de seguimiento*, o *ratio de caras seguidas*, será:

$$RatioSeguim = \frac{r_{caras}}{n_{caras}} \tag{5.1}$$

  El número de *falsos negativos* viene dado por: $n_{caras} - r_{caras}$.

- **Ratio de falsos positivos.** También en analogía al detector, sería el número de regiones devueltas que realmente no corresponden a caras, contadas imagen por imagen. Más precisamente, se calcula con:

$$RatioFPos = \frac{r_{nocaras}}{n_{frames}} \tag{5.2}$$

  En parte, un mal dato de este factor podría achacarse al funcionamiento incorrecto del detector de caras. No obstante, el seguimiento de una no-cara también puede deberse a una pérdida del rostro, que debería ser identificada por el proceso. Cuanto mejor y más fiable sea la detección de la situación de pérdida, menor será este ratio.

[3] Véase la ecuación 4.2 y la definición de los fallos de localización, a partir de la página 167.

- **Precisión de localización.** Todas las medidas de precisión definidas en el capítulo 4 (ver la página 167 y sucesivas) pueden aplicarse ahora sobre las caras seguidas: diferencias de ángulo, de posición central, de tamaño de las caras, y errores en las posiciones de ojos y boca. Las tres últimas suelen ser las más interesantes, porque reflejan la precisión del resultado. Pero recordemos que la obtención de estas medidas requiere un etiquetado manual, que también estará sujeto a cierto error.

- **Tiempo de ejecución.** Ya hemos subrayado la importancia del factor computacional en el problema de seguimiento. El tiempo se mide como el número de milisegundos necesario por cada rostro seguido. Obviamente, el valor concreto depende del ordenador utilizado en los experimentos. Alternativamente, el orden de complejidad puede aportar una medida más abstracta del coste del algoritmo.

Existen más criterios relevantes que no se reflejan en ninguna de estas medidas. Por ejemplo, supongamos una secuencia en la que se producen 10 falsos negativos; no es lo mismo que todos ellos tengan lugar al principio del vídeo, o que el seguimiento se corte en 10 momentos diferentes. Seguramente, la segunda situación será más grave que la primera. Por ello, en algunos experimentos indicaremos el *número de cortes del seguimiento*.

Otro criterio relevante es la estabilidad de los resultados. El concepto de *estabilidad* está relacionado con la evolución de los errores a lo largo del tiempo. El seguimiento de una cara puede ser impreciso pero estable, si el error en la localización de los ojos es el mismo en todos los *frames*. Algunos autores evalúan la estabilidad y la fiabilidad con medidas asociadas al propio método usado, como los *residuos* o el *error de reconstrucción*; esta elección dificulta la comparación con técnicas que no se basen en ese tipo de medidas.

Otros parámetros interesantes de las técnicas de seguimiento pueden hacer referencia a sus condiciones normales de trabajo: máximo desplazamiento entre *frames* admitido para las caras; mínima resolución (en píxeles) para un funcionamiento aceptable; márgenes tolerados en los distintos giros: inclinación, giro lateral y vertical. Más difíciles de cuantificar son la robustez frente a cambios de expresión facial, iluminación, oclusión parcial, etc., si bien lo interesante de estos factores es la comparación de resultados aplicando diversas técnicas sobre unas mismas secuencias de prueba.

5.2. El estado del arte en seguimiento de caras

La variedad de problemas diferentes que han sido denominados con el término genérico de *seguimiento de caras* es enorme: desde los sistemas que siguen la cabeza como una nube gaussiana imprecisa [16, 130, 87, 53], hasta los que ajustan una malla 3D deformable y adaptable a la expresión facial [32, 44, 1, 171]; desde los que se plantean como simples aplicaciones repetidas del proceso de localización [147, 132, 66, 24, 169], hasta los que requieren un entrenamiento específico para cada usuario [32, 20, 19, 121, 171]; desde los que se basan en un modelo

rígido y predefinido de la cara [169, 170, 24], hasta los que se podrían aplicar a cualquier clase de objetos [115, 179, 15, 74].

La existencia de esta "plétora de seguidores" [108] es, posiblemente, una de las causas por las que hasta la fecha no ha aparecido ninguna revisión completa y autoritativa del estado del arte[4]. Lógicamente, tampoco entra en los objetivos de esta tesis abordar esa labor de recopilación. En su lugar, intentaremos dar una visión global de la disciplina, concretando unos pocos trabajos interesantes y relacionados con el tipo de técnicas que proponemos nosotros. La relación es siempre débil, ya que el seguimiento basado sólo en proyecciones no ha sido aplicado previamente a las caras (aunque sí en otros contextos, como vimos en el capítulo 1).

La estructura de esta sección es la siguiente. En primer lugar, el apartado 5.2.1 clasifica los métodos existentes usando varios criterios alternativos. En el apartado 5.2.2 se describen algunos mecanismos habituales de predicción y otras formas de hacer uso de la información temporal. Después, el apartado 5.2.3 profundiza en algunos seguidores que usan color y otras características de bajo nivel. Finalmente se comentan los fundamentos de los algoritmos basados en apariencia dentro del apartado 5.2.4.

5.2.1. Clasificaciones de los modelos y técnicas de seguimiento

Existen diferentes formas de clasificar los seguidores faciales. Según el modelo de seguimiento podemos distinguir: métodos 2D, 2,5D, 3D, modelos de puntos y de mallas. Según la estrategia de búsqueda encontramos: seguidores basados en movimiento y basados en modelos. Finalmente, según el modo de trabajar con las imágenes tenemos: métodos basados en apariencia y basados en características.

Estas categorías no son disjuntas o excluyentes, sino más bien ortogonales. Por ejemplo, una técnica basada en apariencia puede adoptar una estrategia que use movimiento o modelos; y un algoritmo basado en características puede hacer seguimiento 2D, 3D, etc. Lógicamente, algunas combinaciones son poco viables; así, los seguidores con mallas deformables son casi siempre basados en modelos. Veamos las principales propiedades de cada grupo.

Clasificación según el modelo de seguimiento

La forma de describir los resultados del seguimiento es uno de los aspectos más relevantes en el diseño de un seguidor de caras. A grandes rasgos, podemos distinguir las siguientes categorías de modelos usados en el problema:

- **Seguimiento 2D.** Estos métodos realizan un seguimiento plano de la cara en las imágenes [53], que incluye las estimaciones de posición y tamaño del rostro, es decir, 3 grados

[4] A este respecto, podemos mencionar algunas revisiones aunque parcialmente relacionadas con el tema o incompletas. Por ejemplo, en [180] se hace una exhaustiva recopilación de seguidores de caras, pero esta revisión data de 1998. Más recientes son [49] y [85], aunque el primero se centra en el análisis de expresiones faciales y el segundo en monitorización y vigilancia. También se puede consultar el capítulo 4 de [108], que repasa de manera muy superficial los seguidores de caras 3D, describiendo unos pocos casos concretos.

de libertad. Por ejemplo, el resultado podría venir dado con un rectángulo, como en la figura 5.6a); se puede ver que esta descripción resulta bastante imprecisa.

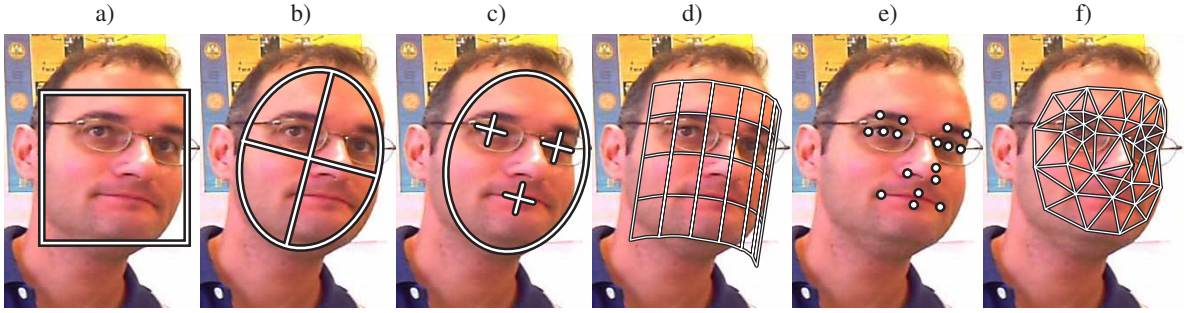


Figura 5.6: *Posibles modelos para el seguimiento facial. a) Seguimiento 2D mediante rectángulo contenedor de cara. b) Seguimiento con elipse contenedora. c) Seguimiento 2,5D con localización de componentes faciales. d) Seguimiento 3D con modelo cilíndrico de la cara. e) Seguimiento de puntos característicos. f) Seguimiento con malla deformable.*

- **Seguimiento 2,5D.** Además de la posición y tamaño de la cara, se añade alguna información sobre la orientación de la cabeza[5], típicamente la inclinación (giro respecto del plano de imagen) –lo que da lugar a 4 grados de libertad–. A su vez, podemos distinguir entre los que siguen globalmente la cabeza [16, 130, 87], y los que siguen elementos faciales [57, 20, 214]. En la figura 5.6b,c) se ilustra la diferencia entre ambos. Aunque los dos contengan los mismos grados de libertad, en los primeros no se conoce exactamente la posición de ojos y boca, que pueden variar con diferentes giros. En los segundos, la posición de la cabeza se deduce de las localizaciones de los componentes.

- **Seguimiento 3D.** En este grupo se resuelven los 6 grados de libertad correspondientes a posición y orientación de la cabeza [23, 88, 180, 8, 172, 196, 74]. Si se utiliza un modelo de cara –no necesariamente lo hacen todos los seguidores de esta clase–, es posible deshacer la transformación para obtener una imagen normalizada del rostro. La figura 5.6d) muestra uno de los formatos más típicos, mediante un modelo cilíndrico [196, 23].

- **Seguimiento basado en puntos característicos.** El resultado del seguimiento es un conjunto disperso de puntos correspondientes a localizaciones predefinidas del rostro [169, 66, 199, 88]: ojos, cejas, esquinas de la boca, etc., como se puede ver en la figura 5.6e). El número de grados de libertad depende del número de puntos. Algunos métodos son capaces de deducir la pose a partir de los puntos [199], aunque la mayoría trabajan de forma plana. Otros incorporan también el contorno de la cabeza [169].

- **Seguimiento con mallas y modelos deformables.** Esta es la alternativa que presenta un mayor número de grados de libertad. Se basa en la definición de un modelo genérico de cara mediante una malla densa de puntos, como en la figura 5.6f). La malla puede ser

[5]El concepto de 2,5D aparece en otros muchos contextos, como en los *sketch* 2,5D con los que David Marr describe el paso de 2D a 3D [118]. Aquí lo usamos simplemente para referirnos a algo intermedio entre 2D y 3D.

2D [105, 33, 171, 32] o 3D [44, 1]. Los puntos de la malla no varían arbitrariamente, sino que existe una serie limitada de modos de variación, incluyendo movimientos, giros 3D, expresiones faciales y el aspecto del individuo. Los modelos pueden ser de forma sólo [33, 105], o de forma y textura [32, 171, 44, 1].

Clasificación según la estrategia de búsqueda

Esta clasificación hace referencia a las técnicas de análisis de imágenes empleadas en el seguimiento. De acuerdo con [108], se pueden distinguir estrategias basadas en movimiento y basadas en modelos, también denominadas *feed-forward* y *feed-back*, respectivamente.

- **Seguidores basados en movimiento.** En estos métodos, la posición de la cara se deduce a través de técnicas de flujo óptico y análisis de movimiento; por ejemplo, con técnicas de *matching* por bloques [35], o con el algoritmo de Lucas y Kanade [115], aplicados sobre partes del rostro. En la figura 5.7a) se puede ver un ejemplo del primer tipo de técnicas. Con los resultados del análisis se calcula la nueva posición 2D o 3D de la cara, usando métodos de mínimos cuadrados o filtros de Kalman extendidos [122]. Algunos ejemplos de estos seguidores son [7, 12, 88]. La característica común de estos sistemas es que se apoyan únicamente en las diferencias de píxeles entre un *frame* del vídeo y el anterior. Esto conduce a la aparición del problema de *deriva* (en inglés, *drifting*): los pequeños errores en las localizaciones se acumulan a lo largo de la secuencia, de manera que el resultado puede acabar completamente descolocado de su posición correcta. Para evitarlo, se han propuesto seguidores basados en movimiento pero aplicando restricciones derivadas de los modelos [43, 44, 67].

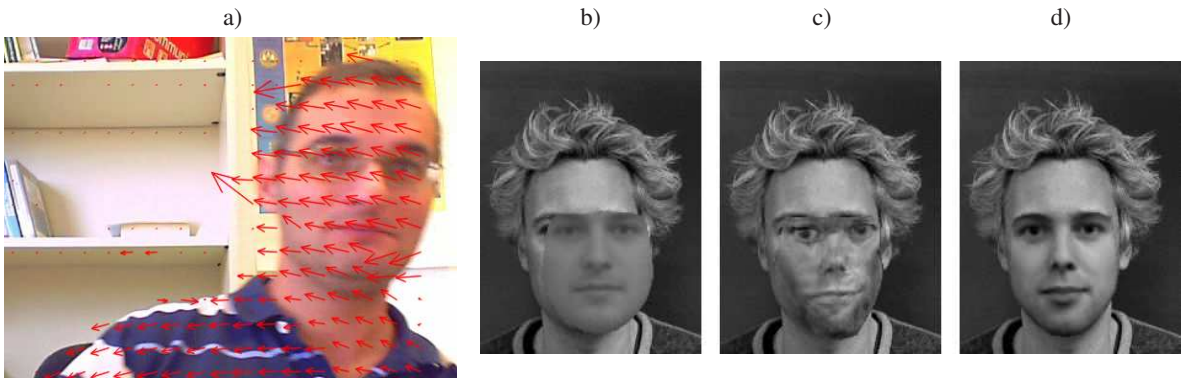


Figura 5.7: *Estrategias de seguimiento basadas en movimiento o en modelos. a) Seguimiento mediante movimiento utilizando flujo óptico (matching por bloques) sobre la secuencia "ggm5.avi". b-d) Seguimiento mediante ajuste iterativo de modelos deformables [171]. Se muestran varias iteraciones del proceso de búsqueda. Información extraída de:* `http://www2.imm.dtu.dk/~aam/`

- **Seguidores basados en modelos.** Los modelos de cara almacenan conocimiento a priori sobre propiedades de los rostros (como el color, las proyecciones, etc.), sobre su apariencia 2D, su forma 3D o su dinámica. La estrategia de búsqueda en los algoritmos de

este grupo se apoya en la aplicación del modelo. El proceso de seguimiento se puede interpretar como un ajuste del modelo a la nueva imagen, esto es, buscar la posición, orientación y los posibles parámetros del modelo que mejor encajen con la cara actual, como se muestra en la figura 5.7b-d). La mayoría de los seguidores que hemos mencionado previamente se pueden clasificar en este grupo [16, 130, 57, 214, 105, 33, etc.]. El modo de operar de estos métodos evita el problema de *drifting*, pero su debilidad se encuentra en que el modelo usado puede ser poco flexible para adaptarse a todas las situaciones posibles.

Ahlberg y Dornaika [108], distinguen otras dos subcategorías dentro de esta clase: los que construyen el modelo a partir del *primer frame* de la secuencia; y los que utilizan *modelos estadísticos*. Los primeros tienen la ventaja de la flexibilidad. Potencialmente pueden funcionar de forma fiable y precisa con cualquier individuo [57, 67, 172]. El inconveniente es que la inicialización del modelo es crítica, por ejemplo, si la imagen inicial no es muy representativa. Los segundos requieren un entrenamiento, que puede ser genérico o específico para cada usuario [19, 105, 1, 23]. Como ejemplo de esta distinción, en métodos basados en color de piel tenemos los siguientes casos: en [16] el modelo de color se calcula seleccionando la cara en la primera imagen; en [130] existe un modelo genérico y predefinido de color de piel que se aplica y se adapta al contenido de la secuencia; en [169] el modelo de color es fijo y definido a priori.

Métodos basados en apariencia o en características

Esta última clasificación está estrechamente relacionada con las que propusimos para los problemas de detección y localización. Los seguidores pueden tratar las imágenes de las caras de forma global (*holística*) o bien usando características extraídas de las mismas.

- **Seguimiento basado en apariencia.** Estos métodos [74, 20, 19, 196, 44, etc.], intentan aprovechar toda la información de las imágenes y del modelo, tratando las caras holísticamente. El problema de seguimiento, en general, se puede plantear de la siguiente forma [108]. Tenemos una imagen actual, I, y un modelo de cara parametrizado $I_m(\mathbf{p})$, donde $\mathbf{p}$ son los parámetros del modelo e I_m son las imágenes generadas por el modelo. El seguimiento consiste en encontrar el conjunto de parámetros $\mathbf{p}$ que minimice:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \delta(I, I_m(\mathbf{p})) \tag{5.3}$$

  Siendo δ una medida de distancia. Este es el fundamento de lo que se conoce como *análisis a través de la síntesis*, que ya mencionamos en el capítulo 4. La dificultad estriba en que $\mathbf{p}$ es una variable de muy alta dimensionalidad. Para resolver el problema eficientemente se aplican estrategias heurísticas, técnicas de gradiente descendente, o métodos iterativos como el de Lucas y Kanade [115]. En la mayoría de los casos, el valor inicial de $\mathbf{p}$ en una nueva imagen es el resultante del *frame* anterior.

- **Seguimiento basado en características.** Como en los detectores y localizadores basados en características, se trata de aprovechar propiedades invariantes y fáciles de seguir. El color de la piel es una de las propiedades más usadas [16, 130, 20, 169, 199, 162, etc.], a veces por sí solo y en otras ocasiones como un método complementario. Además, se han usado bordes [53, 77], y regiones predefinidas del rostro, por ejemplo, en sistemas orientados al seguimiento de puntos clave para el análisis de expresiones faciales o la lectura de los labios [66]. También se ha aprovechado la localización de los puntos para obtener la posición 3D de las caras [180, 88].

La aplicación de estos acercamientos no es necesariamente excluyente. Un seguidor puede usar características de muchos tipos diferentes junto con técnicas holísticas, orientadas a una reducción progresiva del espacio de búsqueda del problema. En este sentido, Toyama [180], propone una estrategia llamada *foco de atención incremental* (IFA), de la que se deriva un seguidor que incorpora: agrupación por color de piel, análisis del movimiento, búsqueda holística del patrón de cara, y localización de los componentes faciales. Cada etapa reduce el espacio de búsqueda (centra el foco de atención) de cara a la siguiente.

5.2.2. Mecanismos de predicción y uso de la información temporal

En la discusión del proceso genérico de seguimiento, plasmado en la figura 5.1, distinguimos entre la predicción del nuevo estado y la relocalización de la cara usando los valores predichos. Muchos algoritmos de seguimiento facial –o en general de cualquier tipo de objetos– incorporan filtros avanzados de predicción. Pero también existen bastantes propuestas en la literatura que obvian el predictor, basándose en la hipótesis trivial de que la posición esperada en el nuevo *frame* corresponde a la observada en el instante anterior [147, 132, 57, 169, 66, 24, 214]. Por el contrario, en algunos algoritmos que dan más importancia a los filtros de predicción, suele ocurrir que se aplican técnicas muy simples de análisis de imágenes, como color o detección de bordes (ver algunos ejemplos en [35, 87]). Por otro lado, los filtros predictores también han sido usados como mecanismos para estabilizar un resultado ruidoso u oscilatorio [64, 130].

Métodos de seguimiento basados en aplicación repetida de la localización

Es posible encontrar una gran cantidad de trabajos que abordan conjuntamente los problemas de localización y seguimiento de caras, resolviendo el segundo con una simple aplicación repetida del primero para cada nueva imagen de la secuencia [147, 132, 66, 24, 162]. Incluso hay casos –como el sistema de reconocimiento facial en vídeo de [70, 69]– donde se aplica el detector de forma independiente para cada *frame*. Generalmente estos métodos incluyen mecanismos elementales para aprovechar la información del estado anterior, como buscar los componentes en las posiciones previas, y restringir el tamaño de las zonas de búsqueda.

Algunos otros métodos siguen esta filosofía, pero realizando ligeras modificaciones del

proceso usado en la localización. En este grupo podemos clasificar algunas propuestas descritas ya en el capítulo 4. Como muestra, podemos señalar los siguientes casos:

- El localizador de Sobottka y Pitas [169, 170], basado en color de piel, componentes conexos y proyecciones. En el seguimiento, el análisis de componentes conexos es sustituido por la descripción con *snakes* del contorno de la cara. El *snake* es inicializado en el primer *frame*, y se actualiza con una función de energía que depende del color de piel. Sobre el resultado se aplica el mismo método de integrales proyectivas usado en la localización.

- En el método de Yang y otros [199], el seguimiento se hace reduciendo las ventanas de búsqueda de los ojos y la boca, según la posición en el instante anterior. Los ojos se siguen con una simple búsqueda de mínimos locales en la ventana correspondiente, igual que en la localización. Para la boca se sustituye el método basado en proyecciones por un proceso heurístico basado en niveles de gris.

- Los modelos de forma activa (ASM) [33, 105, 171], son otra de las técnicas en las que seguimiento y localización están estrechamente relacionados. Como vimos en el apartado 4.2.2 del anterior capítulo, se basan en estimar el conjunto de parámetros que mejor adaptan el modelo de cara a la instancia actual. En el seguimiento, los parámetros iniciales para el *frame* t son los resultantes del $t-1$.

- Más recientemente, Zhu y Ji [214], describen un sistema de detección y seguimiento de ojos en imágenes de infrarrojos. El detector busca candidatos por diferencia de intensidades –aprovechando el efecto de "pupila brillante"–, y luego aplica SVM para verificar los candidatos. El seguidor usa sólo el primero de los métodos, en una región estimada con los resultados de un filtro de Kalman.

En principio, con un buen diseño, estos métodos pueden conseguir resultados aceptables. El inconveniente es que, en la mayoría de los casos, el uso que se hace de la información de instantes anteriores es muy reducido.

Predicción mediante filtros de Kalman

Una de las técnicas más populares de predicción son los *filtros de Kalman* [92, 122]. En este contexto, el sistema es modelado con cuatro componentes: el vector de estado interno del proceso, x_t (en nuestro caso, la posición de la cara, su orientación, velocidad, aceleración, forma, etc.); la observación o medición obtenida, z_t (el resultado del análisis de la imagen); una variable de ruido de la medición, m_t; y el ruido en el proceso, w_t. La evolución del estado de un instante t al siguiente es controlada por una matriz de transición, Φ, siendo:

$$x_{t+1} = \Phi \cdot x_t + w_t \tag{5.4}$$

La forma del vector de estado x_t y de la matriz Φ asociada, deben ser seleccionadas de

manera adecuada según el modelo de seguimiento. Por otro lado, la observación z_t está relacionada con el proceso a través de otra matriz H:

$$z_t = H \cdot x_t + m_t \tag{5.5}$$

El algoritmo de Kalman define un modo de obtener la estimación óptima del estado, $\hat{x}_t$, en cada instante, t, partiendo del estado previo, $\hat{x}_{t-1}$, de las observaciones, z_t, las matrices H y Φ, y las matrices de covarianza de w_t y de m_t. Lo interesante es que esta estimación se puede obtener en forma cerrada, aplicando los cálculos indicados en el algoritmo 5.1 (ver la página 256). En el apartado 5.3.2 describimos de forma más completa y detallada una posible aplicación de este tipo de filtros a la predicción en el seguimiento de caras.

Existen muchas variantes del mecanismo original propuesto por R.E. Kalman en 1960. Una de las más interesantes son los llamados *filtros de Kalman extendidos* (EKF) que se aplican cuando las funciones son no lineales. Por ejemplo, Jebara y Pentland [88], utilizan EKF para deducir la estructura y pose 3D de la cara a partir del seguimiento de fragmentos característicos del rostro (ojos, esquinas de la boca y de la nariz). Más recientemente, en [77] se utilizan filtros bayesianos para realizar un seguimiento genérico, donde la propiedad medida son los histogramas de la orientación del gradiente. Para el cálculo rápido de estos histogramas se usan imágenes integrales [188].

Predicción con el algoritmo Condensation

Isard y Blake [87], señalaron el hecho de que los filtros de Kalman suponen una distribución gaussiana para la función de densidad de probabilidad del estado, $p(x_t)$, lo cual puede resultar inadecuado para modelar la incertidumbre de ciertos procesos. Para solventar esta carencia, proponen el *algoritmo Condensation* (*Conditional Density Propagation*). Este algoritmo permite mantener varias hipótesis simultáneas, dando lugar a una distribución multimodal. Por lo tanto, el resultado del cálculo no es una estimación concreta para $\hat{x}_t$, sino una función de densidad de probabilidad $p(x_t)$. En particular, la función se describe con un conjunto de N muestras –vectores de estado–, $\{s_t^1, s_t^2, \ldots, s_t^N\}$, cada una con un peso relativo, $\{r_t^1, r_t^2, \ldots, r_t^N\}$, según su verosimilitud estimada.

De forma simplificada, el algoritmo procede de la siguiente manera. Supongamos que en el instante $t-1$ tenemos ya la estimación de $p(x_{t-1})$, es decir, las muestras y los pesos correspondientes; el método lleva a cabo los siguientes pasos:

- hacer un muestreo aleatorio de x_{t-1} según la distribución de probabilidad $p(x_{t-1})$, obteniendo las muestras: ${s'}_t^1, {s'}_t^2, \cdots, {s'}_t^N$;

- predecir de forma estocástica el nuevo valor, s_t^j, de cada muestra, ${s'}_t^j$, según el modelo de evolución del estado (en terminología de Kalman, sería equivalente a aplicar la ecuación 5.4, sustituyendo x_{t-1} por ${s'}_t^j$, y x_t por s_t^j, para todo j);

- sobre la imagen, medir la propiedad observada, z_t, en las posiciones asociadas a cada s_t^j, obteniendo un valor de densidad de probabilidad $\beta_t^j = p(z_t|x_t = s_t^j)$;
- el nuevo peso para las muestras s_t^j es proporcional a la probabilidad, β_t^j, normalizando todas ellas para que la suma sea 1, esto es: $r_t^j = \beta_t^j / \sum \beta_t$. De esta manera, tenemos las nuevas muestras y con sus pesos, que definen la distribución $p(x_t)$.

En [87] se evalúa la capacidad de seguimiento de esta técnica, usando como observaciones los bordes extraídos de la imagen. Las pruebas incluyen el seguimiento de una cabeza con movimientos rápidos y esporádicos sobre un fondo complejo. Los resultados obtenidos son excelentes, y especialmente cuando se comparan con los filtros de Kalman.

La principal aportación de esta técnica al dominio de las caras humanas es la capacidad de adaptación a situaciones donde el movimiento resulta impredecible, normalmente lento pero rápido y espontáneo en ciertos tramos. Otras cuestiones, como la robustez frente a fondos complejos, no suelen ser tan problemáticas en seguimiento facial, siempre que se usen características más adecuadas que los bordes. Otros tipos de filtros que se han manejado en el seguimiento son los modelos HMM [27], los sistemas de partículas [154] (de los cuales el algoritmo Condensation es una instancia) y los sistemas de múltiples hipótesis.

5.2.3. Seguimiento de caras basado en color y otras características

La utilización de color [16, 199, 130, 180, 20, 162], bordes [53, 77, 87], movimiento [180], niveles de gris [199, 214], y proyecciones [169, 170, 66, 57], es habitual entre muchos métodos que realizan exclusivamente un seguimiento 2D o 2,5D. Estas características suelen ofrecer una adecuada invarianza y suficiente información para un seguimiento no demasiado detallado. El tipo de técnicas que se aplican incluye típicamente: análisis componentes conexos, *snakes*, *blobs*, algoritmo CamShift, Condensation, etc.

Algoritmo CamShift

Una aportación interesante en el seguimiento de caras con color es el algoritmo CamShift (*Continuously Adaptive Mean Shift*) propuesto por Gary Bradsky en 1998 [16], como una extensión del algoritmo *Mean Shift* [55]. El autor describe un interface perceptual completo, desde el proceso de seguimiento hasta la transformación de los resultados en variables de control del interface. Para modelar el color de piel se utiliza el histograma del canal Hue del espacio HSV, descartando los píxeles con bajo valor de saturación; también se eliminan los que sean muy claros o muy oscuros. El histograma se obtiene de la primera imagen de la secuencia y se aplica sobre las restantes, produciendo imágenes de probabilidad de color de piel, $p(x, y)$.

Para seguir la región de la cara se aplica el algoritmo CamShift, que se puede entender como una alternativa robusta al análisis de componentes conexos. El algoritmo parte de una ventana de búsqueda inicial, v, y lleva a cabo los siguientes pasos:

1. Calcular la media de la imagen de probabilidad, p, dentro de la ventana v.

2. Mover el centro de la ventana v según el resultado del paso 1, y cambiar el tamaño según la suma de p dentro de v (es decir, la ventana aumenta o disminuye según las probabilidades que contiene).

3. Repetir los pasos 1 y 2 hasta alcanzar la convergencia.

El método es conceptualmente sencillo, fácil de implementar, robusto, adaptable a movimientos rápidos y muy eficiente. No obstante, sus resultados son muy imprecisos, ya que la cabeza es descrita como una nube gaussiana en la que no se conoce la posición de los elementos faciales, como puede comprobarse en los ejemplos la figura 5.8. Además, la forma de la nube puede verse afectada por la aparición de *distractores* externos (manos, cuello, otras personas, fondo con color similar a la piel, etc.) –véase, por ejemplo, el quinto extracto de la figura 5.8–, o por la escasa definición del color (imágenes muy claras, oscuras o grises).

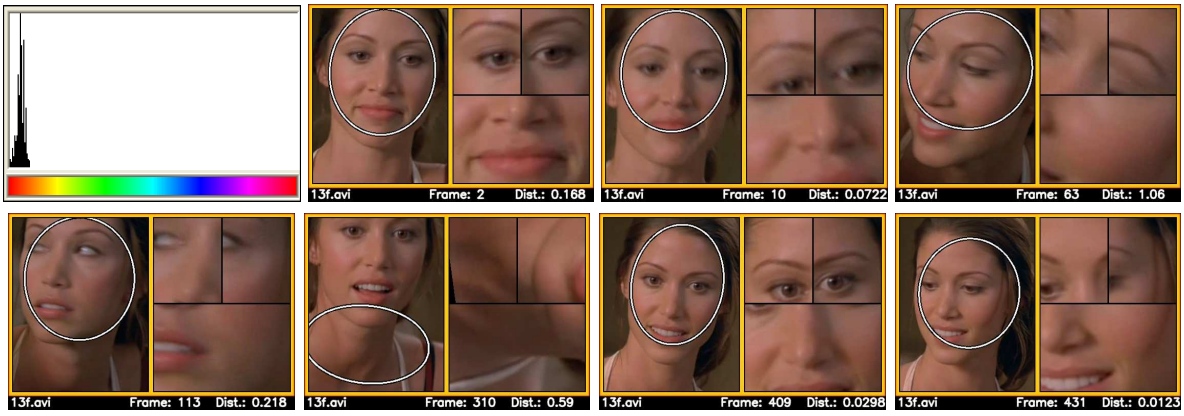


Figura 5.8: *Seguimiento de caras mediante color usando Camshift, sobre la secuencia de prueba "13f.avi". Arriba a la izquierda, el histograma del canal Hue. Para cada frame, se muestra la elipse contenedora de la cara (izquierda) y un extracto de las posiciones esperadas de ojos y boca (derecha).*

En el apartado 5.3.2 (página 256 y sucesivas) proponemos una variación del algoritmo CamShift para paliar estos inconvenientes. El método original es también contrastado en los experimentos de seguimiento de la sección 5.4.

En [16] se pone un énfasis especial en la eficiencia del método. El algoritmo CamShift en sí es muy rápido, ya que alcanza rápidamente la convergencia. Pero cuando trabajamos con imágenes grandes, la transformación RGB a HSV puede ser demasiado costosa[6]. Para paliar este inconveniente, en [187] sugerimos una forma ordenada de reducir el espacio de búsqueda, donde la conversión sólo se debe aplicar en un número muy reducido de píxeles, sin que ello afecte a la precisión del resultado. En este caso, se usaba también el espacio HSV y el resultado es descrito mediante un contorno poligonal.

[6]Por ejemplo, usando las librerías optimizadas Intel OpenCV [35], sobre una imagen de 640×480 píxeles, la conversión tarda unos 12 ms en un Pentium IV a 2,6GHz. A 30 fps, se consumiría más de 1/3 de segundo sólo en la conversión RGB a HSV.

Seguimiento mediante *blobs*

Otra de las propuestas pioneras en el seguimiento facial mediante color es el sistema LAFTER (*lips and face tracker*), desarrollado por Oliver y otros [130]. En relación a la técnica de Bradsky [16], las principales diferencias son:

- En lugar de usar HSV, se toman los canales R y G del espacio RGB normalizado: $r' = r/(r+g+b)$, $g' = g/(r+g+b)$.
- Los modelos de color se describen usando mezclas de gaussianas, cuyos parámetros son estimados con el algoritmo EM. Existe un modelo para el color de piel, otro para el fondo y otro para el conjunto boca/labios.
- La agrupación en regiones de color de piel se realiza con un algoritmo de *clustering* usando una representación de *blobs* para los conjuntos de píxeles; esto es, para cada píxel se forma un vector que combina posición y color: (x, y, r', g').
- De forma parecida a [16], del *blob* asociado a la cara (que es, en definitiva, una forma gaussiana) se extraen los parámetros de posición, tamaño e inclinación.
- Los modelos de color se actualizan de forma continua a lo largo del vídeo. Para evitar una oscilación imprevista, se utilizan filtros de Kalman.

Además de la cara, se extrae también la región de la boca usando el color. Con el análisis de sus variaciones, usando modelos HMM, se describe una aplicación sencilla de generación de animaciones, que distingue un conjunto discreto de estados de la boca: neutra, abierta, triste, sonrisa y riendo con la boca abierta. También se presenta una aplicación de control automático de cámara, destinado a mantener el rostro del usuario centrado en la imagen.

Compensación de los cambios de iluminación

Un posible inconveniente relacionado con el seguimiento mediante color es el cambio en las condiciones de iluminación de la escena. En la práctica, el problema de la *constancia de color* es menos grave en seguimiento que en detección y localización de caras, aunque puede existir. Para tratar de resolverlo se han aplicado diversas estrategias, aunque ninguna de ellas absolutamente fiable:

- Yang y otros [199], proponen una adaptación continua del modelo de color, que es descrito como una distribución gaussiana en el espacio RG normalizado. La media y la matriz de covarianzas del modelo se actualizan según las caras observadas en la secuencia. Una actualización similar es realizada en [130], aunque en este caso con un modelo de mezcla de gaussianas, actualizadas con el algoritmo EM.
- En [116], Lucena y otros tratan el seguimiento facial en condiciones cambiantes de iluminación exterior, donde los individuos pueden pasar de sol a sombra con cierta frecuencia. Para conseguir robustez se utilizan varios modelos de color (en los ejemplos

descritos, 3 modelos) correspondientes a distintas iluminaciones. Cada modelo consiste en un histograma de color en el espacio RGB o en el HSV. El algoritmo localiza la mejor posición según cada modelo, y selecciona el más adecuado de los tres en función de un criterio de similitud de histogramas y posición respecto del *frame* anterior.

- Buenaposada [20], estudia el problema de la constancia de color partiendo del algoritmo conocido como *Grey World* (GW), consistente en dividir cada canal de RGB por la media del mismo, $(r/\overline{r}, g/\overline{g}, b/\overline{b})$. De esta forma se consigue un efecto de "balance de blancos automático", logrando que el seguidor sea invariante frente a cambios de iluminación. El inconveniente surge con los cambios de geometría (por ejemplo, la aparición en la escena de objetos con diferentes colores). Para paliarlo se propone el llamado *algoritmo GW dinámico*, basado en la aplicación de GW básico, junto con una monitorización del color medio de la cara. Cuando se produce un cambio significativo, se vuelva a reiniciar el modelo de color de piel en el espacio GW.

Pese a toda la investigación que se ha dedicado a buscar la constancia de color, no se ha conseguido hasta la fecha ningún método capaz de garantizarla en todas las situaciones [20]. Pero, afortunadamente, los cambios de iluminación no suelen ser muy habituales en las aplicaciones típicas del seguimiento facial. Por ejemplo, un usuario manejando un sistema de videoconferencia no suele encender y apagar luces con frecuencia; y si esto ocurre, se podría simplemente detectar la pérdida del seguimiento y reiniciar el proceso.

5.2.4. Seguimiento de caras basado en apariencia

Como ya hemos visto, la idea de los métodos basados en apariencia consiste en encontrar un desplazamiento –o, en general, una transformación cualquiera– de la cara actual o de puntos característicos de la misma, que minimice la diferencia entre la imagen transformada y el modelo. La diferencia, o *residuo*, suele ser una suma de diferencias al cuadrado. El modelo de cara puede ser 2D o 3D, genérico, entrenado para cada usuario, o bien obtenido de la propia secuencia; además, puede ser un modelo fijo o admitir diversos modos de variación (iluminación, expresión, individuo, etc.).

El aspecto crítico de estos métodos es cómo resolver de forma eficiente el alineamiento en un espacio de elevada dimensionalidad, constituido por los parámetros y posiciones del modelo. La base de muchos sistemas se encuentra en el clásico algoritmo iterativo de Lucas y Kanade [115], desarrollado originalmente para el problema genérico de calcular el flujo óptico. Vamos primero a describir de forma muy breve y simplificada esta técnica, para luego referirnos a algunas extensiones, variaciones y aplicaciones al seguimiento de caras humanas.

Algoritmo de alineamiento iterativo de Lucas y Kanade

Dadas dos imágenes en instantes sucesivos, I_t e I_{t+1}, el objetivo del método es encontrar el desplazamiento que ha tenido lugar en un píxel dado, (x, y), que denotamos por (v_x, v_y)

–velocidad en X y en Y–. La forma más sencilla de solucionar el problema (el algoritmo de *fuerza bruta*) consistiría en aplicar un *matching* de una *ventana* de I_t centrada en (x, y) sobre una vecindad local en I_{t+1}, devolviendo el desplazamiento que genere la menor diferencia.

El algoritmo de Lucas y Kanade [115], define un modo más eficiente y robusto de resolver el mismo problema, evitando la búsqueda exhaustiva y suponiendo que el desplazamiento ocurrido es pequeño. El objetivo perseguido se puede formular como encontrar los valores de (v_x, v_y) que hagan cumplir la ecuación:

$$I_t(x, y) = I_{t+1}(x + v_x, y + v_y) \tag{5.6}$$

Puesto que el desplazamiento del píxel se supone de tamaño reducido, el segundo término se puede desarrollar por series de Taylor:

$$I_{t+1}(x + v_x, y + v_y) \approx I_t(x, y) + \frac{\partial I_t}{\partial x} v_x + \frac{\partial I_t}{\partial y} v_y + \frac{\partial I_t}{\partial t} \tag{5.7}$$

Donde $\partial I_t/\partial x$ es la componente X del gradiente de I_t; $\partial I_t/\partial y$ la componente Y; y $\partial I_t/\partial t$ se puede asociar con la diferencia $I_{t+1} - I_t$. Igualando las ecuaciones 5.6 y 5.7 tenemos:

$$\frac{\partial I_t}{\partial x} v_x + \frac{\partial I_t}{\partial y} v_y + \frac{\partial I_t}{\partial t} = 0 \;\Rightarrow\; \begin{bmatrix} \partial I_t/\partial x & \partial I_t/\partial y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = I_t - I_{t+1} \tag{5.8}$$

Obsérvese que tenemos dos incógnitas y una sola ecuación. Para poder resolverlas, consideramos una pequeña *ventana* de píxeles en torno a (x, y), suponiendo que todos ellos se mueven con la misma velocidad (v_x, v_y). Sean $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, los puntos de la ventana. La unión de todos ellos genera el sistema de ecuaciones sobredeterminado:

$$\begin{bmatrix} \partial I_t(x_1, y_1)/\partial x & \partial I_t(x_1, y_1)/\partial y \\ \partial I_t(x_2, y_2)/\partial x & \partial I_t(x_2, y_2)/\partial y \\ \cdots & \\ \partial I_t(x_n, y_n)/\partial x & \partial I_t(x_n, y_n)/\partial y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} I_t(x_1, y_1) - I_{t+1}(x_1, y_1) \\ I_t(x_2, y_2) - I_{t+1}(x_2, y_2) \\ \cdots \\ I_t(x_n, y_n) - I_{t+1}(x_n, y_n) \end{bmatrix} \tag{5.9}$$

Que se puede resolver en forma cerrada por mínimos cuadrados. Idealmente, con una sola ejecución se obtendría el valor óptimo de (v_x, v_y), el que genera el menor residuo. En la práctica se requieren varias iteraciones del proceso hasta alcanzar la convergencia, aunque por fortuna se consigue muy rápidamente. Esta es la base del algoritmo.

Una limitación de este método es que el máximo movimiento permitido debe ser pequeño, y nunca mayor que el tamaño de la ventana, que tampoco puede ser muy grande para cumplir la suposición de movimiento uniforme (por ejemplo, en [179] es de 15×15 píxeles). Para evitar este problema se sugiere una búsqueda con resolución creciente (*coarse to fine*), donde el algoritmo se repite con diferentes escalados de la imagen. Por ejemplo, en [15] se describe una implementación piramidal del proceso, donde la resolución se va multiplicando por 2

en cada nuevo nivel. En los experimentos de la sección 5.4 contrastaremos los resultados de la versión piramidal del algoritmo de Lucas y Kanade, aplicándola al seguimiento del ojo derecho, el izquierdo y la boca.

Extensiones y variaciones del método de Lucas y Kanade

La gran aportación de Lucas y Kanade es la idea de dar una solución –teóricamente en forma cerrada– para el seguimiento usando los gradientes de la imagen. De forma resumida, los dos ingredientes del método son: (1) el valor de $I_t - I_{t+1}$ mide la variación en la imagen en niveles de gris; (2) los gradientes explican esa variación en función del movimiento en X y en Y. Usando terminología de la robótica, la matriz de gradientes se ha denominado también el *jacobiano*, o *plantillas de movimiento* [20].

En el punto anterior hemos tratado de simplificar al máximo la explicación del algoritmo de Lucas y Kanade. En la aplicación práctica al seguimiento de caras humanas, podemos señalar las siguientes modificaciones sobre ese esquema básico:

- **Tamaño de las ventanas.** El tamaño utilizado para las ventanas permite definir una escala de posibilidades [180], que va desde el seguimiento de pequeñas regiones características [12, 88, 172] (esquinas de la boca, fosas nasales, etc.), hasta la aplicación sobre la imagen completa de la cara [74, 23, 44, 196, 32]. Un término intermedio es el propuesto en [20, 19], donde se distinguen tres grandes fragmentos independientes: ceja y ojo izquierdos, ceja y ojo derechos, y boca.

- **Imagen de referencia.** En el punto anterior hemos supuesto que la nueva imagen, I_{t+1}, se compara siempre con la anterior, I_t, lo cual puede acarrear los problemas de *drifting* ya mencionados. En la práctica, el alineamiento se suele hacer respecto de un modelo, que puede ser una imagen de la misma secuencia [74, 12, 23, 172, 44], o un modelo estadístico entrenado [121, 20, 19, 32].

- **Funciones de modelado del movimiento.** Cuando el tamaño de las ventanas es muy pequeño, el simple modelo de traslación es suficiente. Pero a medida que el tamaño aumenta, se requieren funciones más complejas para describir el movimiento facial. Se han usado: modelos de movimiento plano (rotación, traslación y escala, RTE) [74, 19, 88], afín [74], proyectivo [20, 23, 44, 32], etc. Como consecuencia, el jacobiano de movimiento debe incluir plantillas asociadas a cada posible parámetro del modelo. Por ejemplo, en la figura 5.9 se muestran plantillas aplicables a un movimiento RTE.

- **Movimientos no rígidos e iluminación.** Se han propuesto varias formas de abordar estas cuestiones. Las más frecuentes son: mediante la separación de forma y textura con modelos de malla deformables [172, 44]; y con la utilización de autoespacios [74, 121, 19, 23]. En los primeros se define un mapeo entre la imagen actual y un modelo rectificado, trabajando sobre las imágenes rectificadas. En los segundos se separan las

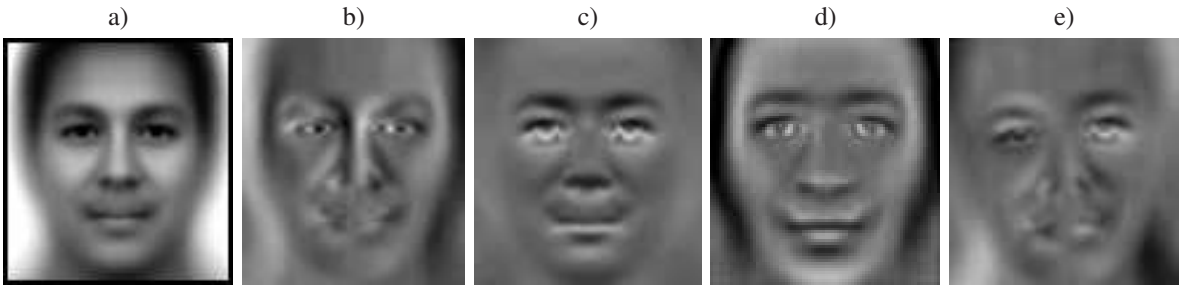


Figura 5.9: *Una cara media e imágenes de gradiente para el algoritmo de seguimiento. a) Cara media, promediada de varias caras de la base FERET. b) Gradiente horizontal. c) Gradiente vertical. d) Gradiente respecto del cambio de escala. e) Gradiente respecto de la inclinación.*

distintas fuentes de variación con PCA o técnicas similares. Por ejemplo, en [19] se representa la cara como la suma en dos subespacios lineales independientes: el espacio de la iluminación y el de las expresiones faciales. Los jacobianos se calculan sobre las bases de estos autoespacios, dando lugar a lo que se denomina como *eigentracking factorizado*.

- **Factorizaciones en series de Taylor.** La factorización en series de Taylor de la ecuación 5.7 no es única. Se han sugerido otros modos de llevarla a cabo, orientados a mejorar la eficiencia de los cálculos [74], o a utilizar un modelo *composicional* en la actualización de los parámetros [121] (en lugar del modelo aditivo subyacente al algoritmo original).

- **Medidas de distancia.** El algoritmo básico de Lucas y Kanade se basa en una minimización de la suma de diferencias al cuadrado. Algunos autores han planteado la utilización de otras métricas [74, 12], buscando mayor robustez frente a valores atípicos.

Globalmente, los métodos basados en apariencia son actualmente los que ofrecen mayor precisión, detalle y estabilidad en el seguimiento 3D de las caras. Algunos de sus principales inconvenientes, en relación a los seguidores basados en características –y donde se centra buena parte de la investigación más reciente–, son su elevado coste computacional, la dificultad de adaptación frente a variaciones rápidas, la reinicialización tras la pérdida del seguimiento, y la mayor complejidad del entrenamiento de los modelos.

5.3. Seguimiento de caras mediante integrales proyectivas

En esta sección vamos a ver cómo es posible construir un seguidor de caras preciso, robusto y eficiente utilizando integrales proyectivas. Como en los problemas tratados en los capítulos anteriores, la base del método es la comparación y el alineamiento de proyecciones. Hacemos uso, nuevamente, de los conceptos presentados en el capítulo 2, donde desarrollamos el contexto teórico para el manejo de proyecciones.

5.3.1. Esquema global del método de seguimiento

Ya hemos discutido la estrecha relación que existe entre seguimiento y localización facial: el seguimiento se puede entender, en esencia, como una relocalización continua a lo largo de una secuencia de imágenes. Este hecho se refleja claramente en el método que proponemos –mostrado a grandes rasgos en la figura 5.10–, que presenta muchas similitudes para ambos problemas. El objetivo del seguidor es actualizar, para cada nueva imagen del vídeo, la elipse contenedora de la cara y las posiciones de ojos y boca; como ambas están asociadas, el problema consta realmente de 5 grados de libertad (los mismos que en la localización).

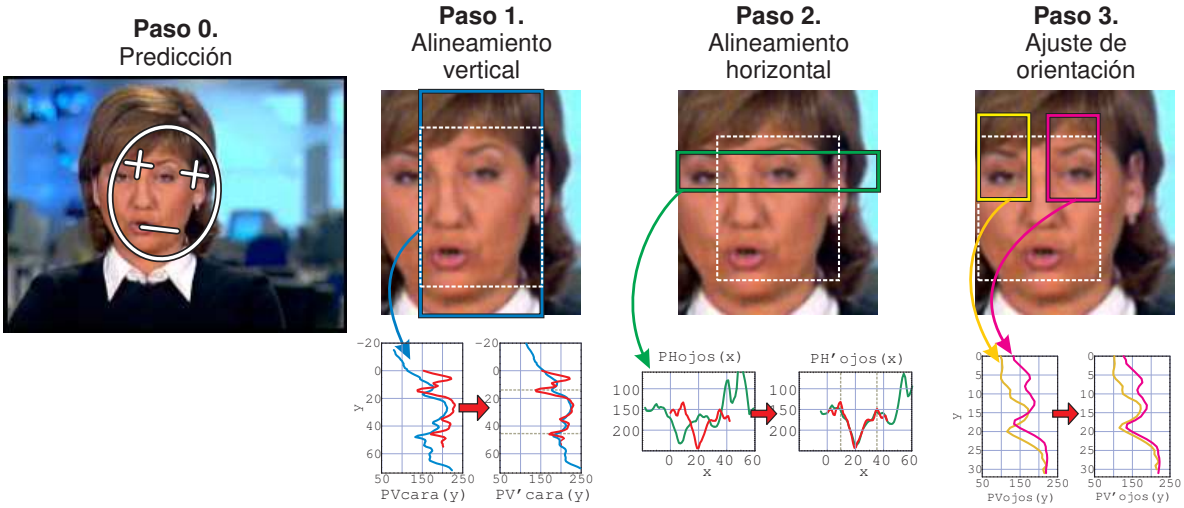


Figura 5.10: *Esquema global del seguidor de caras mediante proyecciones. El proceso parte de una posición predicha, según un modelo de seguimiento dado. Haciendo uso de esa posición, se extrae la cara para los siguientes pasos. Paso 1: se reajusta la posición vertical a través del alineamiento de la proyección vertical de cara, PV_{cara} (en azul), respecto del modelo, MV_{cara} (en rojo), obtenido en la inicialización. Paso 2: se ajusta la posición horizontal con la proyección horizontal de los ojos, PH_{ojos} (en verde). Paso 3: se estima la inclinación usando las proyecciones verticales de ambos ojos.*

Recordemos que la técnica de localización, que desarrollamos en el capítulo 4, constaba de tres grandes pasos: ajuste de la inclinación, alineamiento vertical, y horizontal. El mecanismo de seguimiento introduce las siguientes modificaciones:

- Se añade una nueva etapa de predicción que no se requería con imágenes estáticas. Esto implica definir un modelo adecuado de posición y movimiento de los rostros.
- Los modelos de proyección, MV_{cara} y MH_{ojos}, usados en el alineamiento no son los genéricos, sino que se construyen a partir de las propias instancias de la cara. En particular, se calculan con la primera imagen de la secuencia.
- El paso de estimación de la orientación cambia de orden dentro del proceso, ejecutándose al final del algoritmo, mientras que en la localización se realizaba al principio.

En relación al último punto, se plantea la cuestión: ¿es mejor estimar la inclinación antes o después de los otros pasos? En el problema de localización, la inclinación de la cara puede

ser relativamente grande; pero se espera que la variación en la posición sea más limitada. Sin embargo, en el seguimiento, el desplazamiento –que depende del movimiento de la cabeza en el vídeo– puede ser mucho mayor; la inclinación también puede ser alta, pero lo importante es la variación de ángulo de un *frame* al siguiente, que normalmente no será muy grande. En consecuencia, en el primer problema lo prioritario es estimar el giro, mientras que en el segundo es la posición. Por eso en un caso se ejecuta al principio y en el otro al final.

A continuación describimos más detenidamente todos los pasos del proceso de seguimiento, haciendo especial hincapié en lo relativo a la predicción. La etapa de relocalización de la cara –compuesta por los pasos 1, 2 y 3– será descrita de forma más breve, puesto que es muy parecida al método de localización de componentes estudiado en el capítulo 4.

5.3.2. Predicción de la posición nueva

Predecir significa "anunciar algo que ha de suceder". Anticipar el futuro siempre es arriesgado. Y más cuando lo que se predice no es un fenómeno físico aislado, sino los movimientos de una persona frente a la cámara. Es de esperar, no obstante, que en un corto plazo de tiempo –digamos, dentro del orden de la décima de segundo– el rostro mantenga su trayectoria o la modifique con suavidad. De esta manera, la predicción podría ser útil para mejorar las capacidades de seguimiento de las caras.

La predicción es una parte importante de muchos sistemas de visión. De forma simplificada –suponiendo que las observaciones miden directamente el estado del proceso[7]– el problema de predecir una variable, x, que varía en función del tiempo, se puede formular como: dada una serie de observaciones previas de la variable, $x_0, x_1, \ldots, x_{k-1}$, estimar el valor que tomará x_k. El dato predicho se suele denotar por $\hat{x}_k$. La técnica clásica de predicción son los filtros de Kalman [122], en los que se modela el ruido de la medición y el estado interno del proceso no tiene por qué coincidir con el valor medido.

En este apartado vamos a plantear distintas alternativas de predicción para el caso concreto del seguimiento de caras humanas. Empezamos en primer lugar con dos métodos sencillos, pero que pueden ser viables en determinadas situaciones. Después vemos una posible forma de utilizar filtros de Kalman con un modelo lineal de velocidad. Por último, describimos otro modo de hacer predicciones, no basado en las observaciones previas sino en ciertas pistas que pueden orientar en la relocalización del rostro. En particular, analizaremos cómo el color ayuda a obtener una predicción rápida y robusta. Todos estos métodos serán contrastados en los experimentos de la sección 5.4.

Predicción nula

Aunque hemos justificado el interés de hacer predicciones, en la práctica en muchas aplicaciones típicas el movimiento relativo de la cabeza será escaso o nulo. Por ejemplo, un

[7] Evidentemente, esto no tiene por qué ocurrir siempre así, como veremos enseguida.

usuario situado frente a una cámara web, manejando un interface perceptual, o un presentador de televisión, analizado en un sistema de indexación multimedia, presentarán muy poco movimiento. En tales situaciones es factible usar lo que podemos denominar como *predictor nulo* o *trivial*: las posiciones predichas para los ojos y la boca en el *frame k* serán las posiciones obtenidas en el *frame* $k-1$.

Con esta simple predicción, la *máxima velocidad admitida* en el seguimiento está limitada por la capacidad de ajuste del mecanismo de relocalización. Por ejemplo, suponiendo que trabajamos con vídeo a 30 fps y que se puede relocalizar una cara desplazada $1/4$ de su ancho, la velocidad máxima sería aproximadamente de unos 0,8 m/s. Esto puede ser más que suficiente en un gran número de escenarios.

La principal ventaja de esta alternativa es su extrema sencillez. Además, evita las dificultades de técnicas más avanzadas para manejar cambios bruscos de velocidad. Debido a la *inercia* de esos otros métodos, puede ocurrir con facilidad que la distancia de la cara real a la predicha sea mayor que el desplazamiento de un *frame* al siguiente (por ejemplo, con un balanceo de la cabeza); de esta forma, se puede dar la paradoja de que el seguimiento funcione mejor con la predicción nula que con una más avanzada. Este es el motivo de que muchos de los seguidores repasados en la sección 5.2 no utilicen ningún predictor.

Predicción lineal básica

El mayor inconveniente de la predicción nula es que fija un límite para la máxima velocidad permitida. Cualquier movimiento que la supere hará que se pierda el seguimiento. Una forma sencilla de superar esta limitación es modelar la velocidad de la cabeza suponiendo un movimiento lineal uniforme. Sea x una variable genérica, la predicción sería del tipo:

$$\hat{x}_k := \hat{x}_{k-1} + \hat{v}_{k-1} \cdot \triangle t \tag{5.10}$$

Donde $\hat{x}_k$ es el nuevo valor previsto de x; $\hat{x}_{k-1}$ es el predicho en el instante anterior; $\hat{v}_{k-1}$ es la velocidad estimada; y $\triangle t$ es la unidad de tiempo (podemos asumir que vale 1). Si fijamos un factor de inercia, α, la velocidad prevista en el instante k sería:

$$\hat{v}_k := \alpha \cdot \hat{v}_{k-1} + (1-\alpha)\frac{x_k - \hat{x}_{k-1}}{\triangle t} \tag{5.11}$$

Siendo x_k el valor observado de la variable en k, y tomando los valores iniciales:

$$\hat{x}_0 := x_0 \tag{5.12}$$

$$\hat{v}_0 := 0$$

Estos modelos deberían repetirse para cada una de las variables a predecir, en nuestro caso, los dos ojos y la boca. No obstante, tratar de modelarlos de forma separada puede acabar produciendo un movimiento incontrolado, que no mantenga la estructura del rostro. En su

lugar, usamos variables que describen la posición global: el centro de la cara, (c_x, c_y), el ángulo de inclinación, a, y el tamaño, s. En la figura 5.11 se muestra gráficamente el significado y la obtención de estos parámetros a partir de las posiciones de ojos y boca.

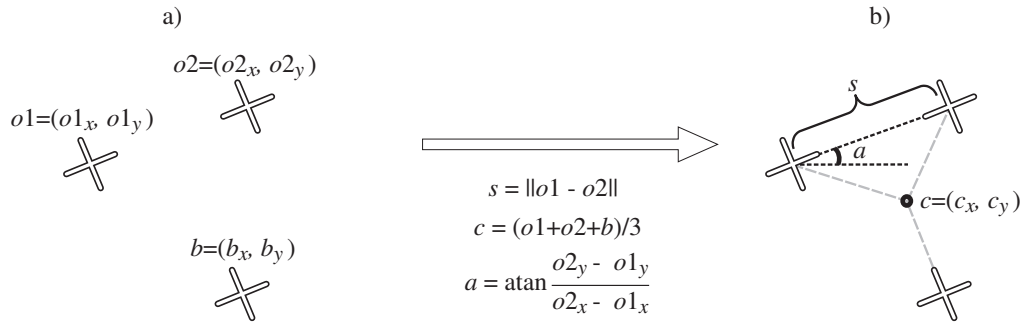


Figura 5.11: *Parámetros de descripción global de la cara usados en la predicción. a) Posiciones de ojos $(o1, o2)$ y boca (b) resultantes de la localización. b) Parámetros de posición central (c), inclinación (a) y tamaño (s) usados para la predicción. Se indica la forma de calcularlos. Nótese que el segundo método tiene menos grados de libertad que el primero.*

Este mecanismo de predicción es relativamente sencillo, y puede ser útil en ciertos casos. Pero un inconveniente importante es la falta de un factor de *amortiguamiento*. Por ejemplo, suponiendo que la cara empieza a moverse y después se detiene completamente, la predicción nunca llegará a estabilizarse, sino que realizará un movimiento oscilatorio en torno a la posición observada. En última instancia, esto es debido a la simplicidad excesiva del modelo, que no distingue entre el estado del proceso y la observación realizada.

Predicción mediante filtros de Kalman

La técnica desarrollada por R.E. Kalman [92], ofrece una solución óptima, en sentido estadístico, para el problema de predicción [181]. La literatura relacionada con los filtros de Kalman es muy extensa, empezando por el libro clásico de P.S. Maybeck [122]. Aquí vamos a centrarnos en la predicción lineal de las cuatro variables de interés mostradas en la figura 5.11: (c_x, c_y, a, s), correspondientes al centro de la cara, el ángulo y la escala, respectivamente. En principio, suponemos que todas son independientes entre sí, por lo que planteamos la solución para una de ellas, la c_x; el mecanismo para las demás sería idéntico.

Básicamente, el filtrado de Kalman consiste en un proceso iterativo en el que existe un estado interno que se va actualizando según las observaciones realizadas. Supongamos un movimiento lineal como el de la fórmula 5.10. En este caso, el estado interno sería un vector: $x_k = [c_{x,k}; v_{x,k}]^{\mathrm{T}}$, con la posición (c_x) y la velocidad (v_x) de la cara en el eje X para cada instante k. La ecuación 5.10 puede expresarse de forma matricial como:

$$x_k = \Phi \cdot x_{k-1} + w_{k-1} \tag{5.13}$$

Donde w_{k-1} es una variable aleatoria que modela el error del proceso –que se supone gaussiano y con media cero–, y Φ es la matriz constante:

$$\Phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{5.14}$$

Por su parte, las observaciones realizadas en el instante k, que denominaremos z_k (en el caso que estamos tratando, un simple número real con la posición central de la cara en X), se pueden modelar como aproximaciones a x_k sujetas a cierto error.

$$z_k = H \cdot x_k + m_k \tag{5.15}$$

Siendo ahora m_k el error de observación –también supuesto gaussiano de media cero– y H la matriz:

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{5.16}$$

Para aplicar el algoritmo de Kalman necesitamos estimar las matrices de covarianzas de las variables de error w_k y m_k. No existe un método trivial para conseguirlo. Pero, afortunadamente, lo importante es el peso relativo entre ambas, que indicará si las medidas observadas son más o menos fiables; o, lo que es lo mismo, si hay menor o mayor inercia, respectivamente. Sea Q la matriz de covarianzas de w_k, y R la de m_k. Podemos tomar un factor de inercia, α, y asignar los valores[8]:

$$Q = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix};\ R = [10^{-1}] \tag{5.17}$$

Para un valor pequeño de α, por ejemplo, del orden de 10^{-5}, el proceso es poco ruidoso frente al error de las observaciones, lo que hace que aumente la inercia de las predicciones. Por el contrario, cuando α es grande, próximo a 10^{-1}, la observación es más fiable y contribuye más a la actualización del estado.

Una vez con todos estos datos, sólo nos queda aplicar ordenadamente las ecuaciones del filtro de Kalman, que se detallan en el algoritmo 5.1.

Además de los parámetros ya explicados, necesitamos asociar valores iniciales a la estimación del vector de estado $\hat{x}_0$, y a la matriz que modela la covarianza del error de estado, P_0, también conocida como *matriz de covarianzas del error estimado a posteriori*. Para $\hat{x}_0 = [c_{x,0}; v_{x,0}]^{\mathrm{T}}$, tomamos la posición inicial de la cara en X, y velocidad 0. Por su parte, para la inicialización de P_0 se recomienda usar valores grandes y arbitrarios [181]. En nuestro caso, tomamos simplemente una matriz identidad[9].

Una vez ejecutado el algoritmo para cada iteración, la predicción de la posición y velocidad de c_x en el siguiente *frame* de la secuencia será:

[8] Obsérvese que en la ecuación 5.17 existe otra simplificación, al suponer independencia estadística entre los errores del proceso para las variables de posición, c_x, y velocidad, v_x.

[9] Hay que destacar que este valor no es crítico, ya que se va actualizando en el proceso del algoritmo.

FILTRO DE KALMAN
ENTRADA:
- Última predicción del vector de estado: $\hat{x}_{k-1}$.
- Medición realizada en el instante k: z_k.
- Matriz de transición del proceso: Φ.
- Matriz asociada a las observaciones: H.
- Matrices de covarianzas asociadas al error del proceso y de las observaciones: Q, R.
- Matriz de covarianzas del error de estado en $k-1$: P_{k-1}.

SALIDA:
- Nueva predicción del vector de estado: $\hat{x}_k$.
- Matriz de ganancia en el instante k: K_k.
- Matrices de covarianza, a priori y a posteriori, del error de estado en k: P'_k, P_k.

ALGORITMO:

$P'_k := \Phi \cdot P_{k-1} \cdot \Phi^{\mathrm{T}} + Q$
$K_k := P'_k \cdot H^{\mathrm{T}} (H \cdot P'_k \cdot H^{\mathrm{T}} + R)^{-1}$
$\hat{x}_k := \Phi \cdot \hat{x}_{k-1} + K_k (z_k - H \cdot \Phi \cdot \hat{x}_{k-1})$
$P_k := (I - K_k) P'_k (I - K_k)^{\mathrm{T}} + K_k \cdot R \cdot K_t^{\mathrm{T}}$

Algoritmo 5.1: *Filtrado de Kalman para la predicción de una variable. Se ha simplificado ligeramente la notación respecto del algoritmo original, suponiendo que las matrices Φ, H, Q y R no se modifican con el tiempo. Ver el texto para una explicación más detallada del significado de los parámetros y los valores que suelen tomar.*

$$\hat{x}_{k+1} = \Phi \cdot \hat{x}_k \tag{5.18}$$

No es nuestra intención profundizar más aquí en los filtros de Kalman ni justificar las ecuaciones que lo caracterizan, mostradas en el algoritmo 5.1. En general, el filtrado funciona bien como un mecanismo de suavizado de las observaciones tomando un valor de inercia adecuado. Sin embargo, ya hemos comentado que el movimiento de la cabeza tiene un componente de arbitrariedad difícil de modelar con los filtros de Kalman. Con una inercia grande aparecen los problemas ya discutidos en situaciones de movimientos rápidos y esporádicos; mientras que si la inercia es baja, el resultado se acercaría mucho al del predictor nulo. Aunque se puede añadir la aceleración al estado interno del proceso, creemos que no mejorará sustancialmente los inconvenientes, sino que más bien puede empeorarlos.

Predicción mediante color

El obstáculo con el que tropiezan los mecanismos anteriores es que se requiere una adaptación mucho más rápida a los movimientos del individuo. Y, en estas circunstancias, es difícil conseguir un buen compromiso entre *inercia* y *rapidez*. Vamos a plantear un método alternativo que evita estas dificultades, considerando el problema de predicción desde una perspectiva diferente; reformulamos la cuestión de interés: estimar la nueva posición de la cara antes del paso de relocalización.

En cierto sentido, podemos entender la predicción como una localización *grosso modo* de la cara, que será refinada después en los sucesivos pasos del seguimiento. De esta forma,

es admisible que la predicción para el instante k use el *frame* k de la secuencia, manejando propiedades globales y rápidas de calcular. En concreto, proponemos hacer uso del color como esa característica global de las caras. Ya justificamos que el uso de color es más factible en vídeo que en imágenes estáticas: primero, porque se puede garantizar la constancia del color de piel siempre que se mantengan las condiciones de captura; y segundo, porque es más frecuente encontrar vídeo en color que en blanco y negro.

Básicamente, el método de predicción que proponemos consiste en una detección de *nubes* de color piel, usando el espacio HSV y el algoritmo CamShift [16], para obtener un **vector de desplazamiento** entre el *frame* anterior y el actual. De esta forma, el modelo de color de piel se obtiene de la cara extraída en la imagen i_{k-1} de la secuencia, y se aplica sobre las imágenes i_{k-1} e i_k, calculando después las variaciones de una a otra. El proceso es representado gráficamente en las figuras 5.12 y 5.13.

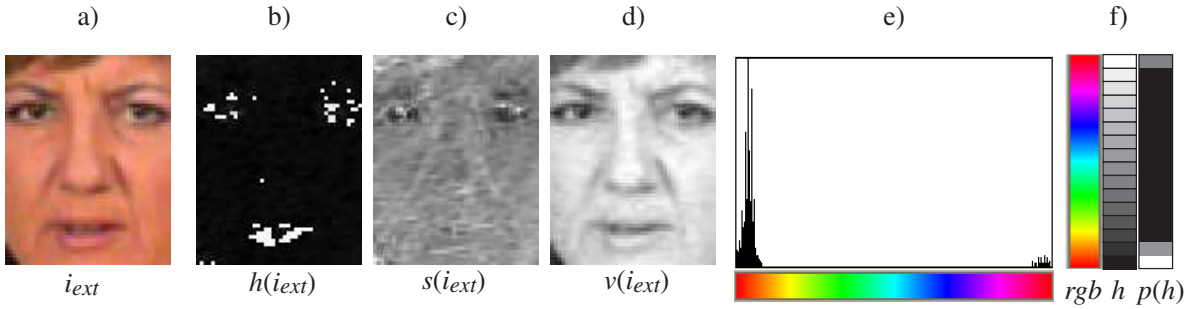


Figura 5.12: *Obtención del modelo de color de piel para el seguimiento. a) Cara extraída en el instante $k-1$ de la secuencia, i_{ext}. b,c,d) Canales H, S y V, respectivamente, de la imagen i_{ext}. e) Histograma del canal H de i_{ext}, descartando los píxeles grises o muy oscuros. f) Transformación resultante (reproyección del histograma); para cada valor de Hue, h, se muestra el resultado, $p(h)$.*

Los tres grandes pasos del algoritmo de predicción por color son los siguientes:

1. **Obtener un modelo de color de piel, p, usando la cara extraída en el *frame* i_{k-1}.**

   1.1. Extraer la región de cara de la imagen i_{k-1}, a la imagen i_{ext}, según la posición del seguimiento en el instante $k-1$.

   1.2. Transformar la imagen i_{ext} al espacio HSV (matiz, saturación, valor).

   1.3. Señalar los píxeles de i_{ext} cuyo valor de S esté por debajo de cierto umbral s_{min} (píxeles próximos al gris) o el valor de V por debajo de v_{min} (píxeles muy oscuros).

   1.4. Construir el histograma del canal H, descartando los píxeles eliminados en el paso 1.3. Es conveniente que este histograma tenga un número reducido de celdas (por ejemplo, del orden de 16 ó 32).

   1.5. Normalizar el histograma del paso 1.4 a valores entre 0 y 1. El histograma resultante será la función $p(h)$ que modela la probabilidad de color de piel para cada valor del canal H.

2. **Aplicar el modelo de color a los *frames* i_{k-1} e i_k.**

   2.1. Convertir la imagen i_{k-1} al espacio de color HSV.

   2.2. Reproyectar el histograma obtenido en el paso 1.5 sobre el canal H del paso 2.1. Es decir, obtener una nueva imagen $p(h(i_{k-1}))$, en la que cada píxel (x, y) toma el valor del histograma asociado al canal H del píxel $i_{k-1}(x, y)$.

   2.3. Poner a 0 los píxeles de $p(h(i_{k-1}))$ que cumplan el criterio expuesto en el paso 1.3.

   2.4. Repetir los pasos 2.1, 2.2 y 2.3 sobre la imagen i_k, obteniendo $p(h(i_k))$.

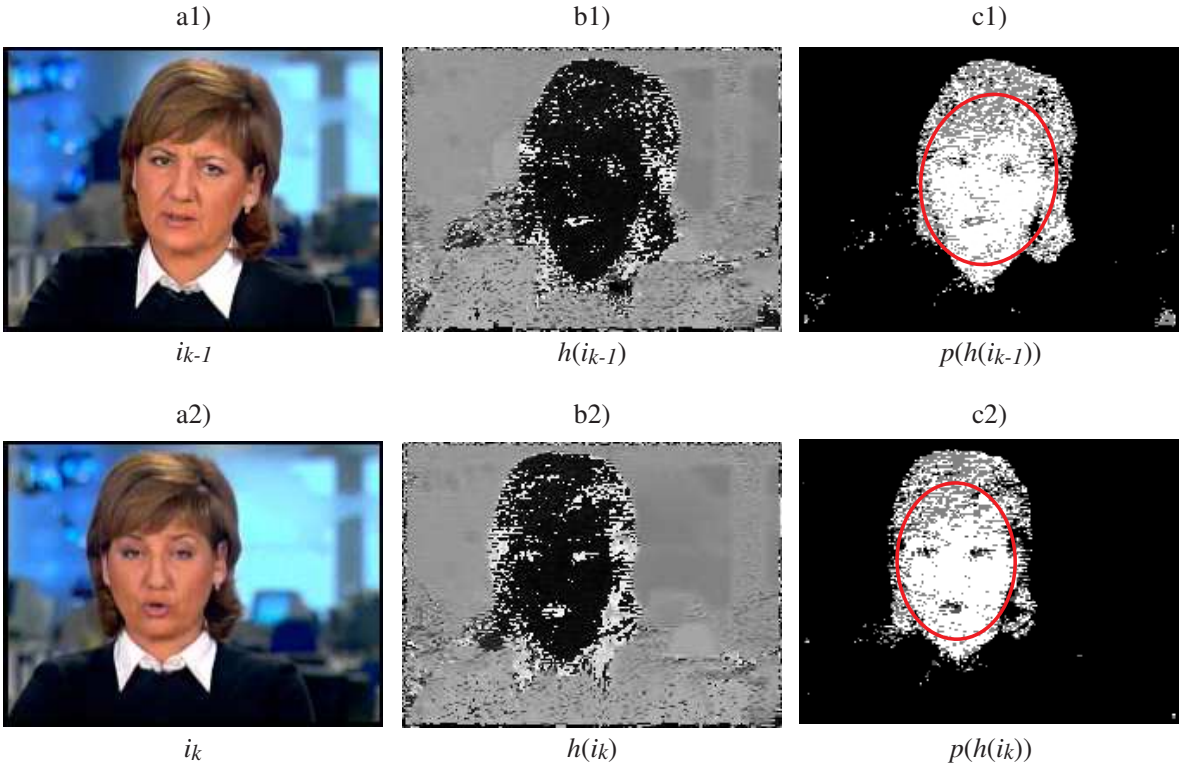


Figura 5.13: *Aplicación del modelo de color de piel y detección del centroide. a1,a2) Imágenes de la secuencia en los instantes $k-1$ y k, respectivamente. b1,b2) Canal H de i_{k-1} e i_k. c1,c2) Aplicación de la función $p(h)$ (ver la figura 5.12) sobre $h(i_{k-1})$ y $h(i_k)$. Se muestra en rojo la nube de probabilidad resultante del algoritmo CamShift.*

3. **Aplicar el algoritmo CamShift sobre $p(h(i_{k-1}))$ y $p(h(i_k))$.**

   3.1. Inicializar la ventana de búsqueda en $p(h(i_{k-1}))$ según la posición y el tamaño de la cara en el instante $k-1$ (esto es, según la iteración anterior del seguimiento).

   3.2. Calcular el centro de masas de $p(h(i_{k-1}))$ dentro de la ventana de búsqueda.

   3.3. Situar el nuevo centro de la ventana de búsqueda en el centro de masas obtenido.

   3.4. Volver al paso 3.2 mientras el proceso no converja. Llamamos al centro de masas resultante cm_{k-1}.

3.5. Aplicar los pasos 3.1-3.4 (el algoritmo CamShift) sobre $p(h(i_k))$. Obtenemos cm_k, el centro de masas de la nube de píxeles de color de piel en k.

3.5. Suponiendo que c_{k-1} es el centro de la cara en el instante $k-1$, el nuevo centro predicho para el instante k será $\hat{c}_k := c_{k-1} + cm_k - cm_{k-1}$.

El algoritmo CamShift ha sido ya usado como un método de seguimiento de caras. Sin embargo, nuestra propuesta difiere en algunos aspectos importantes del modo de aplicación descrito originalmente por Bradsky [16]. En primer lugar, en nuestro caso se trata únicamente de un paso inicial del proceso de seguimiento[10]. En cierto sentido, podemos decir que hay una combinación de métodos. La aplicación del algoritmo CamShift seguida de la relocalización mediante proyecciones, permite obtener las ventajas de ambas técnicas: adaptación frente a movimientos muy rápidos del primero; y precisión en la localización del segundo.

Otra diferencia sustancial es la forma de aplicar el resultado. En [16] se utiliza directamente la elipse resultante como una descripción de la posición, tamaño e inclinación de la cara. Pero esa elección está sujeta a numerosas imprecisiones, puesto que en muchos casos se detectarán como color de piel partes del pelo (como en la figura 5.13), del cuello (ver la figura 5.8) o de objetos del fondo. Además, el ángulo de orientación obtenido es muy impreciso cuando la forma de la nube tiende a ser circular. Por eso: (1) usamos únicamente la posición central de la nube de puntos, sin tener en cuenta la inclinación ni el tamaño; y (2) la predicción se basa en el desplazamiento relativo de la imagen antigua a la nueva, de manera que es irrelevante que la nube no esté centrada perfectamente sobre el rostro (sólo que mantenga su posición relativa respecto del mismo).

5.3.3. Relocalización de la cara

El objetivo de la fase de relocalización es refinar las posiciones obtenidas tras la predicción –sea cual sea el mecanismo aplicado–, analizando la nueva imagen de la secuencia. Los pasos de relocalización del seguidor son análogos al método de localización de componentes faciales, que desarrollamos en el capítulo 4. Pero existen algunas modificaciones menores. Ya hemos adelantado dos de las principales: la inclinación se estima al final del proceso, en lugar de hacerlo al principio; y los modelos de proyección no son los genéricos, sino que se obtienen del propio vídeo.

Existe otro requisito básico que no aparecía en el problema de localización: la necesidad de **determinar el final del seguimiento**, es decir, decidir cuándo se ha perdido una cara, por oclusión, por salirse del campo de visión o, simplemente, por fallo del seguidor. En este sentido, el proceso de relocalización ofrecerá una medida de fiabilidad del seguimiento realizado. Será responsabilidad de un mecanismo posterior, encargado de llevar a cabo las políticas de seguimiento, fijar umbrales para esta medida y decidir qué hacer en caso de pérdida de la cara.

[10] Y, además, opcional, ya que se puede sustituir por otros métodos de predicción alternativos.

Obtención de los modelos de proyección MV_{cara} y MH_{ojos}

El método de localización –y, por lo tanto, también el de relocalización–, se basa en el alineamiento de modelos de integrales proyectivas. Hasta ahora hemos usado siempre modelos genéricos entrenados a partir de ejemplos. En el caso del seguimiento, existe una razón bastante evidente para cambiar esta filosofía: es de esperar que la cara mantenga más o menos su apariencia de un *frame* al siguiente. Observemos los ejemplos de la figura 5.14, obtenidos de una secuencia típica de un presentador de televisión hablando frente a la cámara.

Figura 5.14: *Invarianza de las proyecciones verticales frente a expresión facial. a) Distintos extractos de una secuencia de un presentador de televisión. b) Proyecciones verticales de las caras de la parte a).*

La variación en expresiones faciales es grande; además, aparecen pequeños giros laterales de la cabeza. A pesar de ello, las proyecciones verticales mantienen una estructura bastante homogénea a lo largo de toda la secuencia. Aunque la señal PV_{cara} en un instante dado difiera mucho del modelo genérico, MV_{cara}, las proyecciones del mismo sujeto en distintos momentos son muy similares entre sí.

En consecuencia, proponemos que los modelos de proyección, MV_{cara} y MH_{ojos}, se obtengan a partir de la propia secuencia. Concretamente, planteamos una solución sencilla: calcular estos modelos usando la primera imagen, es decir, la posición resultante del paso de detección.

Cabe hacer algunas consideraciones y matizaciones sobre esta decisión:

- Una primera opción sería asociar directamente MV_{cara} a la proyección calculada para el primer *frame*, PV_{cara}, con varianza uniforme; e igual para MH_{ojos}. Tendríamos lo que en el capítulo 2 denominamos "modelos de proyección media", frente a los "modelos de media/varianza". En la figura 5.15b) se puede ver un caso típico, que usaremos como modelo en los siguientes ejemplos de este apartado.

- El inconveniente del modelo de media es que no tiene en cuenta la distinta variabilidad de diferentes zonas de la cara: la boca es muy variable, los ojos un poco menos, etc. Una alternativa viable sería utilizar modelos media/varianza *actualizados de forma continua* para cada nueva imagen de la secuencia.

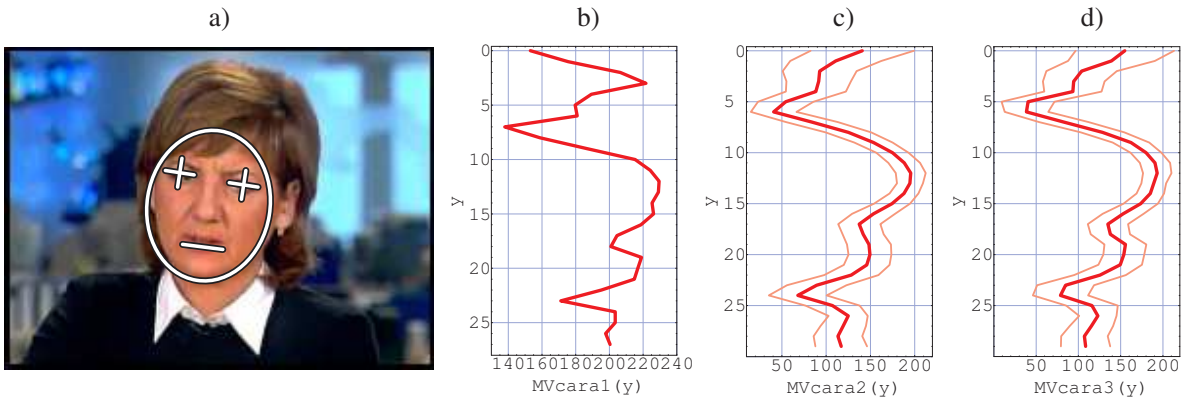


Figura 5.15: *Distintas formas posibles de obtener los modelos de proyección para el seguimiento de caras. a) Primera imagen de la secuencia, a partir de la cual se calcula el modelo. b) El modelo de MV_{cara} es directamente la proyección vertical de la cara extraída, PV_{cara}. Nótese que en este caso no tenemos varianza. c) Modelo media/varianza genérico, MV_{cara}. d) Modelo promedio entre los obtenidos en b) y c), después del alineamiento del primero.*

En nuestras pruebas no ha quedado claro que esta extensión suponga una mejora significativa. Por un lado, el proceso se hace más complejo y, por lo tanto, menos eficiente. Pero el verdadero inconveniente es que el modelo creado no es necesariamente mejor que el modelo medio, si los datos con los que se ha construido (las caras previas en la secuencia) no son representativos; por ejemplo, puede que no haya cambios en la boca, y la varianza en esa zona tenderá artificiosamente a cero.

- Aunque el modelo medio funciona bien en muchos casos, depende excesivamente de la representatividad de la cara inicial. Si ésta presenta una expresión muy exagerada, por ejemplo, pueden ocurrir problemas en los alineamientos posteriores. Una forma de reducir este inconveniente es tomar una *proporción del modelo genérico*. Es decir, si la proyección calculada en el primer *frame* es P y la proyección genérica es G, el modelo usado en el seguimiento será: $\alpha \cdot P + (1 - \alpha)G$. Es más, este modelo puede incluir la varianza del modelo genérico. En la figura 5.15d) se muestra un ejemplo de este promedio entre la cara seguida y el modelo genérico, añadiendo la varianza de este último.

Normalmente, la elección del tipo de modelo usado y la forma de calcularlo no son el factor más influyente en la robustez del seguimiento. Pero sí que pueden tener una gran importancia en la precisión y estabilidad del resultado. En los experimentos veremos algunos ejemplos concretos. El modelo promedio será el usado en la mayoría de los casos.

Paso 1. Alineamiento vertical de la cara

Después de predecir la nueva posición de la cara en la secuencia, el siguiente paso del algoritmo de seguimiento consiste en refinar la localización y escala vertical del rostro. La forma de proceder se detalla en el apartado 4.3.3 del capítulo anterior (ver la página 193). Se puede comprobar en la figura 5.16 la aplicación sobre un ejemplo.

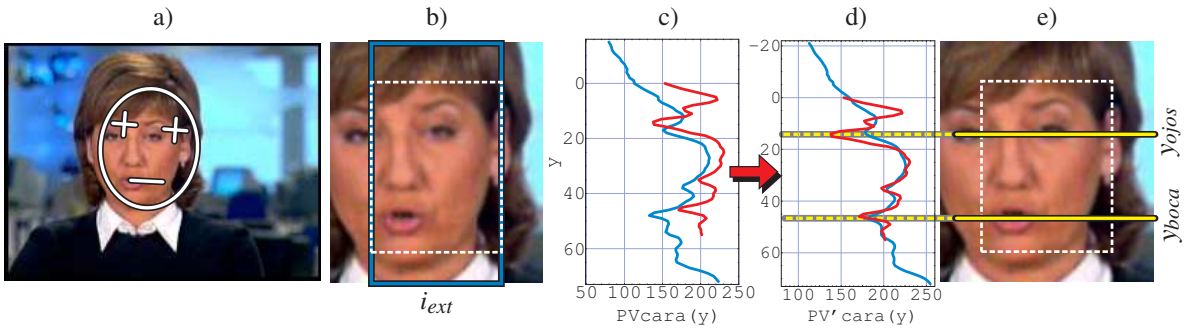


Figura 5.16: *Paso de alineamiento vertical en el proceso de seguimiento. a) Posición predicha de la cara en la nueva imagen de la secuencia. b) Extracción de la cara, mediante una transformación afín. c) Proyección vertical de la cara extraída, PV_{cara}, incluyendo márgenes superior e inferior (en azul); y el modelo, MV_{cara} (en rojo). d) Las mismas proyecciones después del alineamiento. e) Con los resultados del alineamiento, se puede reajustar la posición Y de la cara.*

Vamos a recordar las operaciones que constituyen este proceso de manera muy resumida:

1. Se calcula una transformación afín, basada en las posiciones predichas de ojos y boca, para extraer la cara de entrada a una imagen, i_{ext}, con tamaño y posiciones estándar, definidos en el modelo de cara. Esta imagen incluye márgenes adicionales de tolerancia a los cuatro lados.

2. Se calcula la proyección vertical de i_{ext}, incluyendo los márgenes superior e inferior, pero no los laterales. Esta proyección se denomina PV_{cara}.

3. Se aplica el algoritmo 2.4 para el alineamiento rápido de proyecciones, sobre PV_{cara} y el modelo MV_{cara} creado para el seguimiento de esta cara.

4. Los parámetros resultantes del alineamiento (desplazamiento d y escala e) sirven para indicar la nueva posición y escala de la cara en sentido vertical, como se ilustra en la figura 5.16e).

La distancia de alineamiento obtenida en el paso 3, se puede utilizar para determinar la fiabilidad del seguimiento. Cuando desaparezca una cara que está siendo seguida, la distancia aumentará de forma súbita, lo cual servirá para detectar la situación de pérdida.

Por otro lado, el tamaño del margen adicional extraído (dentro del paso 1) está relacionado con la máxima velocidad permitida para el rostro (suponiendo una predicción nula). Por ejemplo, si el margen es del 20 % de la cara, podrá haber un desplazamiento del 20 % sin que se salga del fragmento extraído. Lógicamente, si se reduce este margen se disminuye el movimiento permitido; aunque si se aumenta indefinidamente, no se garantiza necesariamente una mayor velocidad de seguimiento.

Paso 2. Alineamiento horizontal de la cara

Los resultados del alineamiento vertical nos permiten fijar las posiciones de la cara y sus componentes en el eje Y. En particular, podemos calcular la altura de los ojos; después, proyec-

tamos horizontalmente esa región para relocalizar la cara en el eje X. En la figura 5.17 se muestra este paso del seguidor.

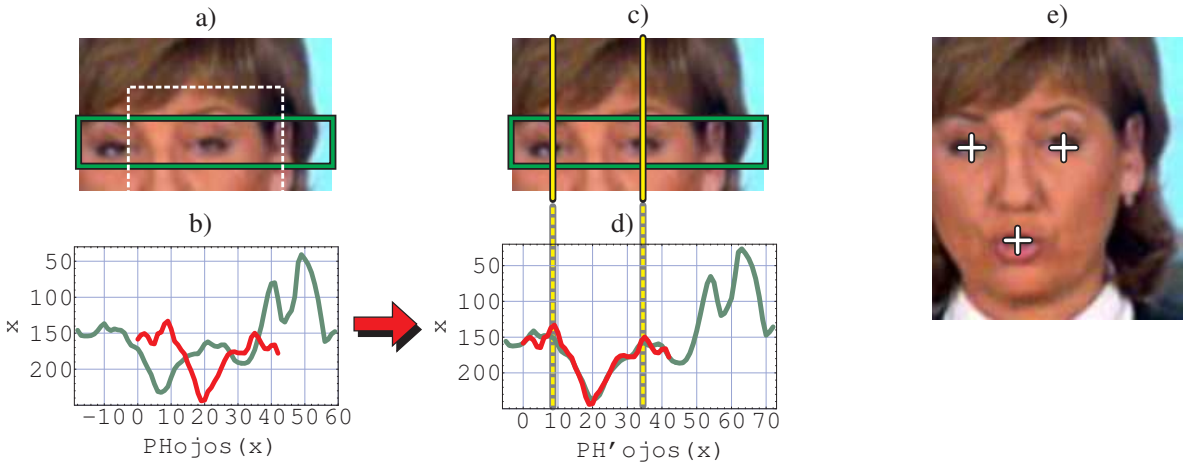


Figura 5.17: *Paso de alineamiento horizontal en el proceso de seguimiento. a) Región de ojos en la cara extraída, i_{ext}, según los resultados del alineamiento vertical. b) Proyección horizontal de la región de ojos, PH_{ojos} (en verde), incluyendo los márgenes laterales; y el modelo, MH_{ojos} (en rojo). d) Las mismas proyecciones después del alineamiento. c) Con los resultados del alineamiento, se reajusta la posición X de la cara. e) Posiciones de ojos y boca, después del ajuste horizontal y vertical. La cara presenta una inclinación muy ligera.*

Los detalles matemáticos de este procedimiento se estudiaron en el apartado 4.3.4 (a partir de la página 195). Resumimos los cálculos aplicados:

1. Usando el desplazamiento d y escala e del paso anterior, y los parámetros $y_{ojosmin}$, $y_{ojosmax}$ del modelo de cara, se calcula la región de ojos sobre la imagen extraída, i_{ext}. Esta región incluye los márgenes laterales.

2. Se calcula la proyección horizontal de la región asociada a los ojos, PH_{ojos}.

3. Se aplica el algoritmo 2.4 para el alineamiento rápido de proyecciones, sobre PH_{ojos} y el modelo MH_{ojos} creado para el seguimiento.

4. Como en el proceso anterior, los parámetros resultantes del alineamiento indicarán la posición y escala del rostro en sentido horizontal.

También igual que en el paso anterior, la distancia resultante del alineamiento puede ayudar a decidir cuándo ha desaparecido la cara seguida. Aunque, debido a su mayor variabilidad, los umbrales deberían ser mayores para esta proyección.

Paso 3. Estimación de la inclinación

El último paso del seguimiento consiste en estimar la inclinación de la cara en la imagen extraída, i_{ext}. Debemos recordar que lo que estamos calculando, realmente, es la variación de ángulo respecto a la imagen anterior de la secuencia. De esta forma, es posible seguir caras

con cualquier orientación –incluso aunque estén invertidas– si la velocidad de giro está entre los márgenes admisibles.

Veamos el proceso de forma muy sintética. Se puede consultar una descripción más detallada en el apartado 4.3.2 (página 189).

1. Usando los desplazamientos y escalas resultantes de los dos pasos anteriores, se calculan las regiones contenedoras de los ojos, $ojo1$ y $ojo2$, según las proporciones del modelo de cara, en la imagen i_{ext}.

2. Se obtienen las proyecciones verticales de ambas regiones, PV_{ojo1} y PV_{ojo2}.

3. Sobre las proyecciones se aplica el algoritmo 4.1 para el alineamiento de proyecciones usando sólo desplazamientos.

4. Con el desplazamiento resultante, $desp$, y la distancia entre los ojos, $dist_{ojos}$, se estima la variación del ángulo de inclinación con: $\arctan(desp/dist_{ojos})$.

5. Se aplica la inversa de la transformación afín (la obtenida en el paso 1) sobre las posiciones alineadas de ojos y boca, obteniendo así las localizaciones resultantes.

En la figura 5.18 se presenta un ejemplo de este paso.

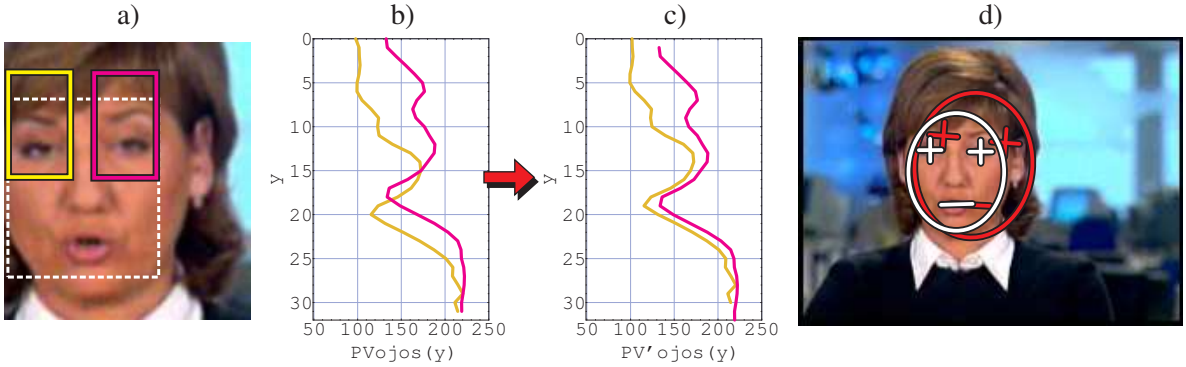


Figura 5.18: *Estimación de la orientación en el proceso de seguimiento. a) La cara extraída, i_{ext}, y las regiones calculadas para ambos ojos (en amarillo y violeta). b) Proyecciones verticales de las regiones de ojos, PV_{ojo1} y PV_{ojo2}. c) Las mismas proyecciones después de alinearlas entre sí. d) Resultado final del seguimiento (en blanco) y la posición que había sido predicha (en rojo).*

5.3.4. Políticas de seguimiento

A diferencia de los algoritmos de detección de caras y localización de componentes, el seguimiento es un proceso que se ejecuta de forma continua y reiterada. Por ello, se hace necesario definir una política de seguimiento que establezca qué hacer frente a cualquier situación posible: cómo detectar la pérdida de una cara, cuántas caras se pueden seguir como máximo, cómo permitir que aparezcan caras nuevas, etc. La mayoría de las cuestiones no tienen una respuesta universal, sino que están ligadas a los requisitos de la aplicación. Por ello, vamos

a destacar los principales aspectos a considerar, pero sin fijar cuál es el valor o el modo de funcionamiento óptimo.

Se entiende que estas políticas serán llevadas a cabo por un proceso de nivel superior, que controla la ejecución de los algoritmos de detección, localización y seguimiento, además de la captura de imágenes de la entrada de vídeo, como ya vimos en la sección 5.1. En principio, el sistema debería permitir configurar los distintos modos de funcionamiento que desarrollamos a continuación.

Detección de pérdida de la cara

Posiblemente, tan importante como seguir las caras es conocer cuándo éstas han desaparecido de la escena, están ocultas, o simplemente ha fallado el seguimiento. En cualquier caso, debemos disponer de un mecanismo que nos indique cuándo se ha perdido un rostro.

A lo largo del apartado 5.3.3 hemos visto cómo las distancias resultantes de los alineamientos vertical y horizontal (que denominaremos d_{pv} y d_{ph}, respectivamente) se pueden utilizar como un criterio para señalar la situación de pérdida. Mientras el seguimiento sea correcto, se espera que ambas medidas sean pequeñas. Y cuando dejen de verse las caras, la distancia aumentará de manera súbita. Por lo tanto, podemos fijar un umbral máximo de distancia admitida. En concreto, utilizamos la suma de d_{pv} y d_{ph}; cuando supere el tope establecido, decimos que se ha perdido el seguimiento.

La figura 5.19 contiene los resultados de una secuencia de ejemplo. El vídeo es un fragmento de un programa de noticias, capturado de televisión analógica, en el que aparecen tres planos con una persona diferente en cada uno. En total consta de 418 *frames*.

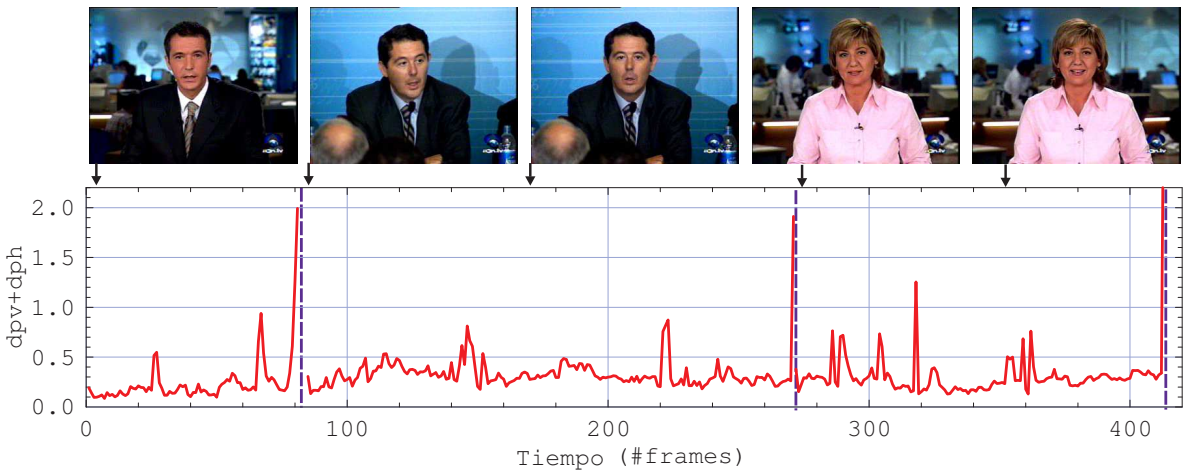


Figura 5.19: *Distancias de alineamiento y detección de pérdida en el seguimiento de caras. La gráfica representa las distancias de alineamiento ($d_{pv} + d_{ph}$) resultantes del seguimiento, para una secuencia de vídeo de 418* frames. *La línea azul discontinua marca el final de cada seguimiento, detectado cuando la distancia supera el umbral. Los cambios de plano tienen lugar en los* frames *80, 270 y 412. En la parte superior se muestran algunas imágenes de la secuencia.*

En el ejemplo de la figura 5.19, los distintos cortes son identificados de forma precisa me-

diante un umbral estándar con valor 1,5. En general, el criterio de la suma de distancias funciona bastante bien, como veremos en los experimentos. Pero, lógicamente, la técnica también puede incurrir en errores, que pueden ser de dos tipos:

- Se supera el umbral cuando la cara no ha desaparecido. El motivo más frecuente para esta situación es un excesivo giro lateral (perfil izquierdo o derecho) o vertical (mirada arriba o abajo). En menor proporción puede deberse a una variación significativa de la iluminación, y raramente a cambios de la expresión facial.

- La cara se pierde pero las medidas de distancia siguen siendo bajas. La causa de este error puede ser un movimiento rápido de la cabeza y que, por circunstancias casuales, el fondo de la escena sea similar a una cara (al menos, desde el punto de vista de las proyecciones).

El segundo tipo de fallo es más grave, ya que implica seguir una no-cara por cierto lapso de tiempo, mientras que el primero puede solucionarse en la siguiente ejecución del detector. Por fortuna, en la práctica, el segundo tipo es muchísimo menos habitual.

Aplicación periódica del detector de caras

En el esquema original (mostrado en la figura 5.1), se aplica la detección y localización en el instante inicial, y después se ejecuta el algoritmo de seguimiento, propiamente dicho, mientras que no se pierda la cara. Pero, ¿qué ocurre si aparece un nuevo individuo en la escena? Simplemente, no se llegaría a encontrar hasta que no se vuelva a recurrir a la detección.

En consecuencia, parece más que conveniente ejecutar de forma periódica el detector facial, incluso aunque no ocurran fallos de seguimiento. La primera cuestión que se plantea es decidir el **tiempo entre detecciones**, es decir, el número de *frames* –o el tiempo, en caso de vídeo en tiempo real–, entre los cuales se aplica el algoritmo de detección de caras. Obviamente, siempre que el seguimiento falle el detector se ejecutará sin esperar este intervalo.

La ejecución periódica del detector da lugar a nuevas cuestiones, que surgen de las posibles incoherencias entre la información que aporta el detector y la del seguidor. El mecanismo que controla la política de seguimiento debe tratar los siguientes casos:

- **Caras no detectadas.** Es posible que alguna de las caras que están siendo seguidas no sea encontrada por el detector. Si hacemos caso del seguidor, la cara debería conservarse; si hacemos caso al detector, debería eliminarse. Puesto que ninguno de los dos es perfecto, no existe una elección ideal. El seguidor puede acabar situado erróneamente en una posición de no cara, por lo que sería mejor la segunda opción; pero también el detector puede incurrir en falsos negativos, de manera que sería preferible la primera.

- **Nuevas caras encontradas.** Si la detección produce nuevas caras, la opción más lógica es añadirlas al seguimiento. No obstante, si el detector está ajustado a un umbral con alto número de detecciones, y por lo tanto también de falsos positivos, podría ocurrir un

aumento incontrolado del número de falsas caras seguidas, que se iría incrementando con cada ejecución del detector.

- **Diferentes posiciones de las caras.** La situación ideal es que tanto el detector como el seguidor coincidan en las caras existentes. Sin embargo, es prácticamente imposible que ambos den las mismas posiciones para las caras y los componentes faciales. De nuevo, surge de la cuestión de decidir qué método es más fiable; en este caso, cuál ofrece una mayor precisión de localización: el seguimiento o el detector/localizador.

En la tabla 5.1 se destacan las ventajas e inconvenientes de cada posible elección en la estrategia de seguimiento. No nos decantaremos aquí por ninguna opción, sino que ésta se debe fijar según los requisitos de cada aplicación concreta.

Parámetro	Valor	Ventajas
Tiempo entre detecciones	Bajo	Disminuye la latencia para encontrar caras nuevas en el vídeo, es decir, se detectan más rápidamente las caras que aparecen cuando hay otras que están siendo seguidas.
	Alto	Reduce el tiempo de ejecución total, ya que la detección es, normalmente, mucho más costosa que el seguimiento. Esto es más importante con vídeo en tiempo real, donde aplicar la detección puede hacer que se pierdan algunos *frames*.
Caras no detectadas	Eliminar	Se pueden eliminar errores de funcionamiento del seguidor, donde se esté siguiendo una falsa cara o haya un gran desajuste en la posición localizada del rostro.
	Conservar	Evita que un falso negativo del detector (sobre una cara que está siendo seguida) haga que se pierda el seguimiento. Esto puede suceder en ciertos casos como inclinación de la cabeza o giros 3D, que resultan difíciles de detectar.
Nuevas caras encontradas	Añadir	Permite la inclusión de nuevos individuos en el seguimiento. Es la opción más lógica, y necesaria cuando se pueden seguir varias caras.
	No añadir	Impide que las falsas alarmas en la detección aumenten el número de instancias de no cara seguidas. Se debe usar si sólo se sigue una cara.
Diferentes posiciones	Actualizar	Permite corregir desviaciones en la posición introducidas con el tiempo. Las posiciones serán más próximas a las ofrecidas por el localizador.
	Conservar	Evita discontinuidades, de un *frame* al siguiente, en las posiciones de la cara y los componentes faciales, que se podrían producir en cada aplicación de detector/localizador.

Tabla 5.1: *Ventajas de las posibles políticas en el proceso de detección periódica, dentro del seguimiento de caras. Para cada valor de los parámetros, se muestran las principales ventajas. En este contexto, los inconvenientes serían las ventajas de la opción opuesta.*

Modo "cara única"

El mecanismo de seguimiento que hemos desarrollado en esta sección es capaz de manejar un número ilimitado de caras, que son seguidas de forma independiente. Pero, a veces, el

número de rostros que deben ser seguidos es conocido o está limitado a priori. Un ejemplo claro son los videojuegos multijugador que usan visión artificial; el número de jugadores fija de antemano el número de objetos de interés.

Es más, son muchas las aplicaciones donde sólo tiene sentido seguir una persona. Por ejemplo, en un sistema de identificación biométrica con vídeo, un individuo interacciona con el ordenador a través de su rostro; no es necesario seguir otras caras, aunque puedan aparecer en la escena.

En definitiva, el proceso de seguimiento debe poder configurarse para admitir un modo de trabajo de "cara única". Esto tiene varias implicaciones en el mecanismo de control de las políticas. Por un lado, el detector debe seleccionar sólo una cara resultante, la más fiable. Por otro lado, en el seguimiento se puede omitir la ejecución continua del detector propuesta en el punto anterior.

5.4. Resultados experimentales

Repasando superficialmente los vídeos manejados por diferentes grupos de investigación para la evaluación de los seguidores, observamos que algunos presentan variaciones de orientación, pero no de posición ni expresión facial; otros incluyen grandes cambios de iluminación y expresión, pero no de orientación; a veces hay una sola cara presente en toda la secuencia, y otras veces se añaden oclusiones, desapariciones y reapariciones; algunos trabajos manejan escenas de exteriores y otros de interiores, donde la cámara está centrada en la cara del usuario o bien ocupa una pequeña fracción de las imágenes.

En los experimentos que describimos a continuación hemos intentado poner a prueba la mayor variedad posible de situaciones. Para ello, se han utilizado vídeos creados por otros investigadores y disponibles públicamente, capturas de televisión analógica, extractos de DVD, y algunas secuencias propias obtenidas con cámaras de videoconferencia de bajo coste.

Todas las pruebas han sido llevadas a cabo en una aplicación creada para este propósito, utilizando Borland C++ Builder y las librerías Intel IPL y OpenCV [35]. El aspecto gráfico del entorno de experimentación se muestra en la figura 5.20.

La aplicación desarrollada permite elegir cualquier combinación de técnicas de detección y localización entre las analizadas en los capítulos anteriores. Normalmente aplicaremos los métodos Haar+IP o IP+Haar para detectar las caras, y el localizador de componentes faciales basado en integrales proyectivas.

Los resultados concretos de los seguidores pueden variar ligeramente cambiando estos procesos previos. No obstante, debemos señalar dos cuestiones a este respecto: (1) en todas las pruebas sobre una misma secuencia se usa siempre la misma combinación detector/localizador; y (2) aunque los resultados de los seguidores se modifiquen un poco al usar otra combinación, la comparación entre métodos (cuál es mejor o peor) se suele mantener casi siempre, es decir, todos mejoran o empeoran proporcionalmente.

En los siguientes apartados vamos a presentar los resultados de los experimentos llevados

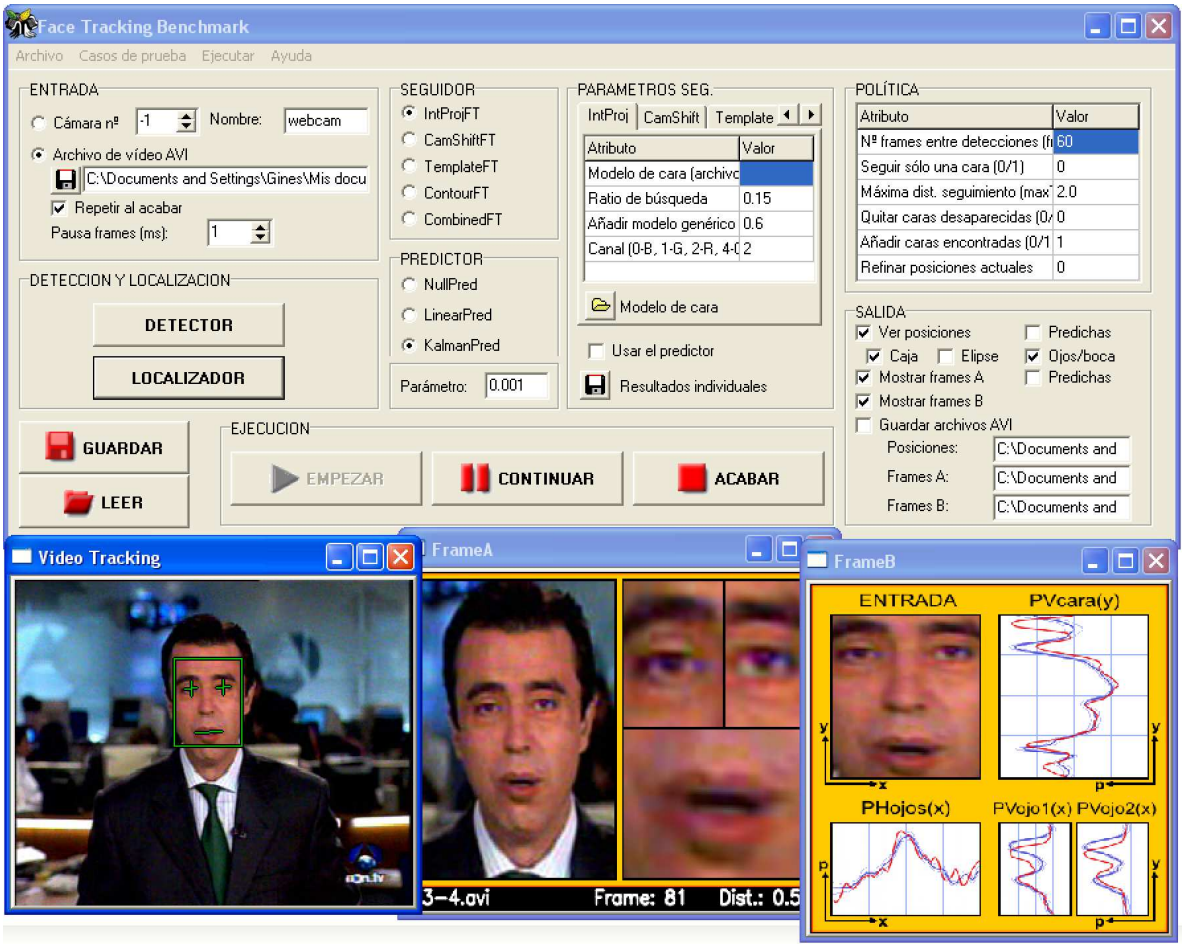


Figura 5.20: *Aplicación creada para la ejecución de los experimentos de seguimiento. En la parte izquierda de la ventana, la entrada (de cámara o de archivo de vídeo) y el detector y localizador utilizados. En el medio, la selección de la técnica de seguimiento, de predicción y los parámetros ajustables de cada método. A la derecha, definición de la política de seguimiento. Abajo, en ventanas separadas, visualización del estado actual del proceso.*

a cabo, comparando el seguimiento basado en proyecciones con otros métodos alternativos. En primer lugar, en el apartado 5.4.1, detallamos esas técnicas alternativas incluidas en la comparativa. Las pruebas han sido organizadas en varios grupos, orientados a analizar la precisión, eficiencia y robustez de los diferentes seguidores. Éstas son descritas en los apartados 5.4.2, 5.4.3 y 5.4.4, respectivamente. Por último, como en los capítulos previos, sintetizamos las conclusiones más importantes de los experimentos en la sección 5.5.

5.4.1. Métodos alternativos de seguimiento

Con el propósito de contrastar los resultados de la técnica de seguimiento propuesta, hemos incorporado algunos de los métodos disponibles en las librerías de procesamiento de imagen y visión artificial Intel OpenCV [35]. Algunos de ellos son genéricos y otros están orientados al dominio específico de las caras humanas. En concreto, manejaremos el algoritmo

CamShift [16], el método iterativo de Lucas y Kanade [115, 15], y un proceso de seguimiento basado en contornos. Estos métodos no utilizan los filtros de predicción introducidos en el apartado 5.3.2 (lineal básica, con filtros de Kalman, y mediante color), ya que disponen de sus propios mecanismos; tales predictores serán aplicados exclusivamente sobre el seguidor basado en proyecciones.

Para disponer de un rendimiento base con el que comparar, hemos creado también desde cero un proceso elemental de seguimiento basado en una simple búsqueda de patrones. A continuación, vamos a señalar algunos de los aspectos más relevantes en relación a la implementación de estas técnicas de seguimiento facial.

IntProy - Seguimiento de caras con integrales proyectivas

Los distintos parámetros del algoritmo desarrollado en la sección 5.3 han sido ajustados para producir un funcionamiento óptimo en un caso genérico, y serán usados en la mayoría de los experimentos, a menos que se indique lo contrario. En concreto, los modelos genéricos de proyección asociados a la cara son como los mostrados en la figura 3.33 (véase la página 132), siendo PV_{cara} de 60 puntos y PH_{ojos} de 48. Tal y como se detalla en el apartado 5.3.3, los modelos aplicados en el seguimiento se obtienen mediante una media ponderada entre el modelo genérico y las proyecciones en el *frame* inicial. En particular, el primero se multiplica por 0,4 y el segundo por 0,6. El margen de tolerancia en el algoritmo de alineamiento de proyecciones es típicamente del 15 % del ancho de las señales.

Para referirnos a las variantes del proceso que aplican el predictor nulo, el lineal básico, el de filtros de Kalman y el basado en color, usaremos los acrónimos **IntProyN**, **IntProyL**, **IntProyK** e **IntProyC**, respectivamente. El segundo y el tercero disponen de un parámetro de inercia, α, cuyo valor será indicado cuando se pongan a prueba estos métodos.

CamShift - Seguimiento de objetos basado en color

Esta alternativa utiliza la implementación del algoritmo CamShift [16], ofrecida por su propio creador, Gary Bradsky, en las librerías OpenCV [35]. La operación disponible toma en la entrada una imagen en escala de grises –calculada por el usuario–, que contiene la probabilidad de color de piel de cada píxel. Es decir, se separan los problemas de modelar el color y de obtener la nube de puntos, estando resuelto el segundo.

Para modelar el color de piel nos basamos en el esquema propuesto en [16] y descrito en el apartado 5.3.2. De forma resumida, se utiliza el canal H del espacio de color HSV, descartando los píxeles muy oscuros (V menor que 80) o próximos al gris (S menor que 30); el histograma de H es discretizado a 16 celdas. Por otro lado, el modelo de color se obtiene siempre del *frame* previo y se aplica al siguiente de la secuencia. Esta estrategia evita los posibles inconvenientes debidos a los cambios graduales o bruscos de la iluminación.

El algoritmo devuelve como resultado una elipse que describe la nube de puntos con máxima probabilidad de piel. Sin embargo, la elipse no siempre está centrada exactamente

en la cara del usuario. Puede existir cierta desviación, por ejemplo, debido a la inclusión del cuello en la región de piel. Para paliar este problema se calcula el desplazamiento, escala y orientación relativos entre la cara y la nube de puntos. Esto consigue reducir parte del problema, aunque provoca malos resultados cuando ocurre una rotación 3D de la cara. También las posiciones de ojos y boca se obtienen en relación a la nube de puntos.

LKTracker - Seguimiento con el método iterativo de Lucas y Kanade

El método iterativo de Lucas y Kanade para el cálculo del flujo óptico [115], es una de las técnicas clásicas para el seguimiento de objetos en general, como ya discutimos en la sección 5.2. OpenCV ofrece una implementación piramidal de este algoritmo, propuesta en [15], destinada a seguir un conjunto disperso de características en una imagen.

En nuestro caso, las características seguidas son las posiciones centrales de los ojos y la boca. Utilizamos 4 niveles en la búsqueda piramidal, y el tamaño de la ventana en cada nivel es de 10×10 píxeles. En principio, esto permite un desplazamiento máximo de 80 píxeles –más que suficiente en la mayoría de las pruebas–. Por otro lado, cada imagen i_t, es comparada con la inmediatamente anterior, i_{t-1}. La detección de pérdida del seguimiento se lleva a cabo mediante una umbralización de la suma de residuos para los tres puntos: $\epsilon(ojo1) + \epsilon(ojo2) + \epsilon(boca)$.

En el ajuste de los parámetros del método hemos intentado crear un buen seguidor, aunque sin entrar en refinamientos de gran calado. Existen, por lo menos, dos debilidades en nuestro modo de utilizar el algoritmo de Lucas y Kanade en el seguimiento facial: en primer lugar, el posible problema de *drifting*, puesto que la comparación se hace siempre entre imágenes sucesivas; en segundo lugar, cada elemento seguido se busca por separado, sin garantizar la coherencia en el aspecto de la cara. Fundamentalmente, el segundo inconveniente afecta más gravemente a la posición de la boca. Sería posible introducir algunas heurísticas sencillas para detectar y corregir estas situaciones.

TemMatch - Seguimiento mediante búsqueda de patrones

Este algoritmo utiliza una estrategia de *fuerza bruta*: para cada nueva imagen de la secuencia, encontrar patrones de ojos y boca extraídos de imágenes anteriores. El tamaño por omisión de los fragmentos es del 60 % de la distancia interocular; y las regiones están centradas en el ojo izquierdo, el ojo derecho y la boca. Para evitar el problema de *drifting*, los patrones son extraídos siempre del primer *frame*.

Dada la nueva imagen, se localiza la mejor aparición de cada patrón usando una medida de correlación normalizada en torno a cierta vecindad local de la posición previa, normalmente de la mitad del ancho de la cara. En el caso de las imágenes a color, se toma la suma de correlaciones de los tres canales, RGB. Las posiciones de máxima correlación se asocian a las nuevas localizaciones de ojos y boca. Es más, el valor de correlación se aplica para establecer el criterio de cuándo se ha perdido el seguimiento.

Esta técnica sencilla funciona bien en condiciones de uso triviales. En cierto sentido, se puede considerar como una medida de rendimiento base, que debería ser mejorada por los mecanismos más avanzados; si bien en circunstancias realistas presenta numerosos y evidentes inconvenientes.

Cont - Seguimiento mediante localización de contornos

El seguimiento de caras mediante localización de contornos es otra de las operaciones disponibles en las librerías Intel OpenCV [35], dentro de un conjunto de funcionalidades experimentales. La idea del método es parecida a la del proceso de localización con contornos, pero aplicada aquí sobre cierta vecindad local de las posiciones previas de los componentes faciales.

Debemos recordar que esta parte de las librerías se encuentra aún en proceso de depuración y mejora, por lo que es posible encontrar diversos problemas. Uno de ellos es la detección de la pérdida de la cara. Aunque se incluye un mecanismo para señalar el final del seguimiento, su funcionamiento no es muy fiable, como veremos en los experimentos. Otro inconveniente no solucionado es la necesidad de mantener una estructura coherente del rostro, que en ocasiones puede degenerarse por completo.

5.4.2. Pruebas de precisión y estabilidad

En este bloque de pruebas nos centramos en la precisión y estabilidad de los distintos métodos, es decir, su capacidad de afinar en la localización de los componentes a lo largo de las secuencias de vídeo. Analizamos en primer lugar el efecto de las distintas técnicas de predicción sobre el seguimiento basado proyecciones; a continuación, comparamos los resultados obtenidos con los seguidores alternativos.

Medidas de precisión de los seguidores

Al igual que en el problema de localización, la *precisión del seguimiento* se define en relación a una referencia que se da por cierta. En nuestro caso, esa referencia proviene de un etiquetado manual que, obviamente, está sujeto a cierto error implícito a su obtención.

Los resultados producidos por un seguidor pueden estar influidos por las localizaciones iniciales[11]. Para evitar la influencia de este factor, tomamos como medida del error la *desviación estándar* de las posiciones en relación al etiquetado manual, y en proporción al ancho de la cara. De esta forma, para cada *frame* de la secuencia, se normalizan las posiciones obtenidas de ojos y boca –según el etiquetado manual–, rectificando la cara y tomando como referencia la distancia interocular; se calcula la varianza de las posiciones normalizadas de cada componente a lo largo de la secuencia; y, finalmente, a partir de ellas se obtiene la

[11] Por ejemplo, una técnica extremadamente precisa y fiable, pero ejecutada con una posición inicial inadecuada de la cara, puede dar lugar a resultados desviados siempre en la misma dirección.

desviación en X, en Y y la total. Como en el caso de la localización, fijamos una distancia máxima en los ojos para declarar que se ha perdido el seguimiento de una cara.

En definitiva, las medidas de precisión utilizadas en las pruebas son las siguientes:

- **Ratio de seguimiento**: proporción de caras existentes en la secuencia, contadas imagen por imagen, que son encontradas por el seguidor, produciendo para los ojos una diferencia máxima del 30 % de la distancia interocular (ecuación 5.1).
- **Ratio de falsos positivos**: porcentaje de regiones encontradas o seguidas por el sistema, contadas imagen por imagen, que no corresponden realmente a caras –o en las cuales la distancia máxima a los ojos supera el 30 % de la distancia interocular–, en proporción al número total de imágenes de la secuencia (ecuación 5.2).
- **Desviación en X y en Y, de ojo izquierdo, derecho y boca**: desviación estándar de cada componente facial en posiciones normalizadas, a lo largo del eje horizontal y vertical, respectivamente, en proporción a la distancia interocular del etiquetado manual.
- **Desviación total de ojo izquierdo, derecho y boca**: se obtiene mediante la raíz cuadrada de la suma de varianza en X y en Y, también en relación a la distancia interocular.

Volveremos a usar aquí las gráficas de densidades de localizaciones, para mostrar de manera más detallada el tipo de imprecisiones en que incurre cada técnica analizada. A estos datos añadimos el tiempo de ejecución medio por imagen de la secuencia, aunque este aspecto se estudia en profundidad en las pruebas del apartado 5.4.3. A menos que se diga lo contrario, los tiempos incluyen todos los procesos involucrados: lectura y descompresión de los ficheros de vídeo AVI, aplicación del detector/localizador –en caso necesario–, seguimiento, medición del resultado y visualización.

Evaluación de IntProy con diversos mecanismos de predicción

El propósito de este primer experimento es contrastar los resultados de los diferentes métodos de predicción introducidos en el apartado 5.3.2: nula, lineal básica, filtros de Kalman y con color. Para ello utilizamos el vídeo propio "ggm2.avi" que se describe en la tabla 5.2 y se muestra en la figura 5.25. La secuencia incluye una muestra variada de expresiones, inclinación, giros y movimientos de la cabeza por la escena a una velocidad media/baja. La estimación del error de etiquetado manual está alrededor del 3,6 % de la distancia entre ojos.

Secuencia	Fuente	Resoluc.	Duración	Compresión	Variación
ggm2.avi	Creative Webcam NX Pro	320 × 240 30 fps	1 min 13,8 s 2214 frames	DivX Pro 5.0.5 6,59 Kbytes/s	Movimiento, inclin., giro 3D, expresión

Tabla 5.2: *Descripción de la secuencia de prueba "ggm2.avi". El vídeo está disponible públicamente en:* `http://dis.um.es/profesores/ginesgm/fip`

En este caso aplicamos el detector IP+Haar, que garantiza un tiempo de ejecución rápido y un número muy reducido de falsos positivos. La política de seguimiento consiste en seguir

una sola cara en la secuencia, y el número de *frames* entre detecciones se establece a un valor alto de 600. En estas condiciones, se ejecuta el proceso de seguimiento con integrales proyectivas, variando el mecanismo de previsión subyacente; además, para las técnicas lineal básica y Kalman se utilizan distintos valores del parámetro que controla la inercia del proceso.

Los resultados de este ensayo se exponen en la tabla 5.3.

Método de predicción	Ratio segui.	Ratio f.pos.	Desv. ojo izq. Total	Desv. ojo izq. X / Y	Desv. ojo der. Total	Desv. ojo der. X / Y	Desv. boca Total	Desv. boca X / Y	Tmp. (ms)
Nulo	**100**	**0**	**6,1**	**5,2**/3,3	**6,7**	**5,6/3,7**	**10,7**	**6,1/8,8**	**8,2**
Lineal, α=0,25	95,7	3,7	11,0	10,2/4,2	13,2	12,2/4,9	31,8	11,6/29,6	8,9
Lineal, α=0,5	96,0	3,9	8,0	6,8/4,3	9,4	8,0/4,9	20,2	7,1/18,9	8,6
Lineal, α=0,75	93,9	4,7	10,2	9,0/4,8	12,0	10,6/5,6	30,2	9,1/28,8	9,1
Kalman, α=10^{-1}	97,3	2,7	6,6	5,3/4,0	7,7	6,3/4,5	15,6	6,8/14,0	8,9
Kalman, α=10^{-2}	96,6	3,4	7,4	5,8/4,5	8,5	6,9/4,9	17,5	7,2/15,9	9,2
Kalman, α=10^{-3}	96,2	3,8	9,5	8,1/5,0	10,0	8,3/5,7	22,6	7,7/21,2	9,4
Kalman, α=10^{-4}	95,5	4,3	9,4	8,4/4,2	11,1	9,8/5,0	22,7	9,2/20,8	10,1
CamShift	**100**	**0**	6,3	5,5/**3,2**	7,3	6,2/3,9	12,2	7,1/9,9	13,2

Tabla 5.3: *Resultados del seguimiento con proyecciones sobre la secuencia "ggm2.avi" para distintos métodos de predicción (definidos en el apartado 5.3.2). La secuencia es descrita en la tabla 5.2. Las medidas de desviación estándar están en proporción a la distancia interocular.*

Los valores obtenidos son bastante esclarecedores: los métodos de predicción más avanzados no sólo no mejoran los resultados del predictor nulo, sino que los empeoran de manera significativa. Sólo el mecanismo basado en color es capaz de mantener las máximas tasas de seguimiento con un reducido error de precisión. Estos datos vienen a confirmar lo que adelantamos en el apartado 5.3.2: **predecir la posición de la cara resulta extremadamente difícil**. Tanto el método lineal básico como el de Kalman se basan en velocidad uniforme, pero el modelo subyacente resulta claramente inadecuado; aunque el movimiento de la cara está limitado, la aceleración puede ser relativamente grande. De hecho, la mayoría de los fallos ocurren al cambiar la cabeza de dirección. En tales situaciones, la predicción *aleja* las posiciones de partida respecto de las reales, y el predictor nulo resulta la elección más conservadora. En relación con esto, es también significativo el hecho de que el estimador de Kalman produzca mejores resultados cuanto menores son los valores de inercia (con α mayor).

Hemos podido comprobar que este fenómeno se repite para la gran mayoría de las secuencias de prueba utilizadas. Los estimadores lineal y de Kalman aportan muy poco en la solución del problema, empeorando los resultados de partida. Por este motivo, serán omitidos de los experimentos descritos en adelante. El estimador mediante color mantiene unos niveles de error similares al método nulo, aunque su coste computacional es más elevado.

Comparación de la precisión en los diferentes seguidores

La bondad de un seguidor no sólo está relacionada con su capacidad de seguir caras en situaciones extremas, sino también con la estabilidad y precisión de sus resultados. En este segundo experimento aplicamos los métodos alternativos de seguimiento –seis en total,

descartando IntProyL e IntProyK– sobre un conjunto de 5 secuencias capturadas de televisión analógica. Los datos de las mismas se pueden consultar en la tabla 5.4. Todas ellas corresponden a fragmentos de programas de noticias, donde el presentador aparece de medio cuerpo hacia arriba, hablando, y en algunos casos con ligeros cambios de posición y orientación.

Secuencia	Fuente	Resoluc.	Duración	Compresión	Variación
a3-5.avi	ATI All-in-Wonder Pro 128	320 × 240 25 fps	6,72 s 169 frames	DivX Pro 5.0.5 5,92 Kbytes/s	Expresión
tve1-3.avi	″	320 × 240 25 fps	10,96 s 274 frames	DivX Pro 5.0.5 5,07 Kbytes/s	Expresión
a3-03.avi	″	640 × 480 25 fps	10,32 s 258 frames	DivX Pro 5.0.5 9,33 Kbytes/s	Expresión, posición
a3-06.avi	″	640 × 480 25 fps	13,12 s 328 frames	DivX Pro 5.0.5 8,27 Kbytes/s	Expresión, giro
tl5-00.avi	″	640 × 480 25 fps	7,2 s 180 frames	DivX Pro 5.0.5 11,16 Kbytes/s	Giros, expresión

Tabla 5.4: *Descripción de las secuencias de prueba capturadas de televisión. El sistema de adquisición es un sintonizador/digitalizador doméstico de televisión analógica. Los vídeos están disponibles públicamente en:* `http://dis.um.es/profesores/ginesgm/fip`.

En este caso se ha cuidado especialmente la exactitud del etiquetado manual, cuyo error estimado es aproximadamente del 2 % de la distancia entre los ojos. El detector utilizado es el método combinado Haar+IP, que junto con la aplicación del localizador basado en proyecciones garantiza una inicialización muy precisa de los seguidores. Los resultados conseguidos por las distintas técnicas sobre las 5 secuencias se presentan en la tabla 5.5.

Método	Ratio segui.	Ratio f.pos.	Desv. ojo izq. Total	Desv. ojo izq. X / Y	Desv. ojo der. Total	Desv. ojo der. X / Y	Desv. boca Total	Desv. boca X / Y	Tmp. (ms)
IntProyN	**100**	**0**	**3,2**	**2,6**/**1,8**	**3,8**	**3,4**/**1,8**	**5,1**	**4,5**/**2,5**	18,8
IntProyC	**100**	**0**	**3,2**	2,7/**1,8**	4,0	3,5/**1,8**	5,5	**4,5**/3,1	35,4
LKTracker	99,8	**0**	7,0	6,4/2,7	9,1	8,7/2,8	13,7	13,4/3,1	**18,1**
TemMatch	94,5	5,5	5,0	4,0/3,0	5,3	4,0/3,5	9,7	9,3/2,7	21,0
CamShift	66,9	33,1	10,7	9,3/5,3	9,6	7,7/5,7	15,3	13,7/6,7	31,5
Cont	98,8	1,2	10,5	4,9/9,3	9,6	4,3/8,6	16,6	5,9/15,6	55,6

Tabla 5.5: *Resultados del seguimiento sobre las secuencias de televisión: "a3-5.avi", "tve1-3.avi", "a3-03.avi", "a3-06.avi" y "tl5-00.avi". Las secuencias están descritas en la tabla 5.4. Se indican los valores acumulados para cada método sobre las 5 secuencias.*

Con el fin de mostrar las imprecisiones en las que incurre cada algoritmo, la figura 5.21 contiene las densidades de localizaciones producidas en proporción a una cara estándar. Cuanto más dispersa aparezca la nube de puntos asociada a cada componente, más impreciso es un método, y viceversa.

En las figuras 5.22 y 5.23 se pueden ver algunos resultados de *frames* concretos de las secuencias, comparando los algoritmos basados en proyecciones con el seguimiento de Lucas y Kanade, y con el algoritmo CamShift, respectivamente.

Recordemos que esta prueba trata de medir la precisión del seguimiento en condiciones no complejas. En general, los resultados de los dos métodos basados en proyecciones mejoran

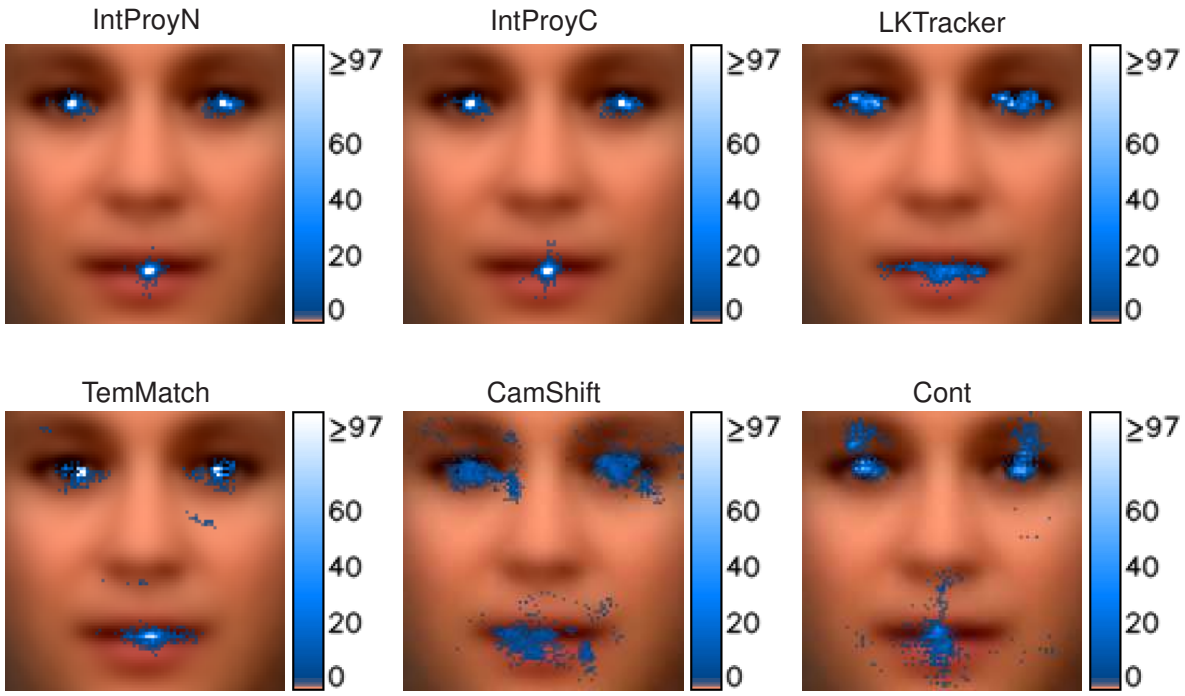


Figura 5.21: *Localizaciones resultantes de los distintos seguidores para los vídeos de televisión: "a3-5.avi", "tve1-3.avi", "a3-03.avi", "a3-06.avi" y "tl5-00.avi". Se muestran las densidades acumuladas para cada método.*

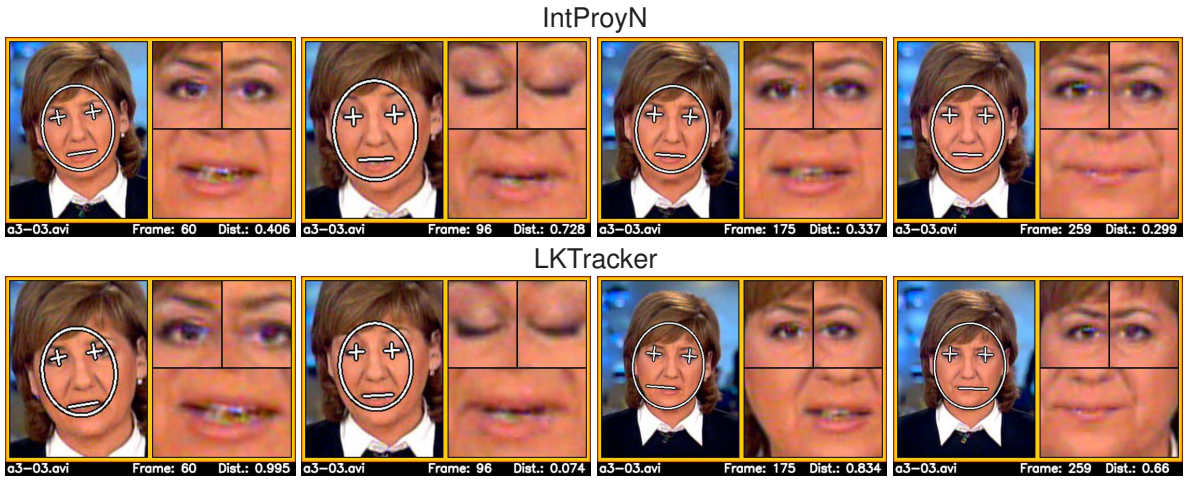


Figura 5.22: *Ejemplos de resultados del seguimiento sobre la secuencia "a3-03.avi". Arriba, posiciones obtenidas por el método basado en proyecciones con predicción nula. Abajo, resultados del algoritmo iterativo de Lucas y Kanade; obsérvese el efecto progresivo de deriva (drifting) en ojos y boca.*

sensiblemente los alcanzados por las técnicas alternativas, tanto para los errores de los ojos como para los de la boca. Las nubes de puntos de IntProyN e IntProyC en la figura 5.21 están concentradas en las posiciones centrales de los componentes, y son muy pocos los casos en los que se desvían de manera notable.

Los restantes métodos provocan unos niveles de varianza muy superiores, y especialmente en la boca. Esta imprecisión era previsible para CamShift, no diseñado específicamente

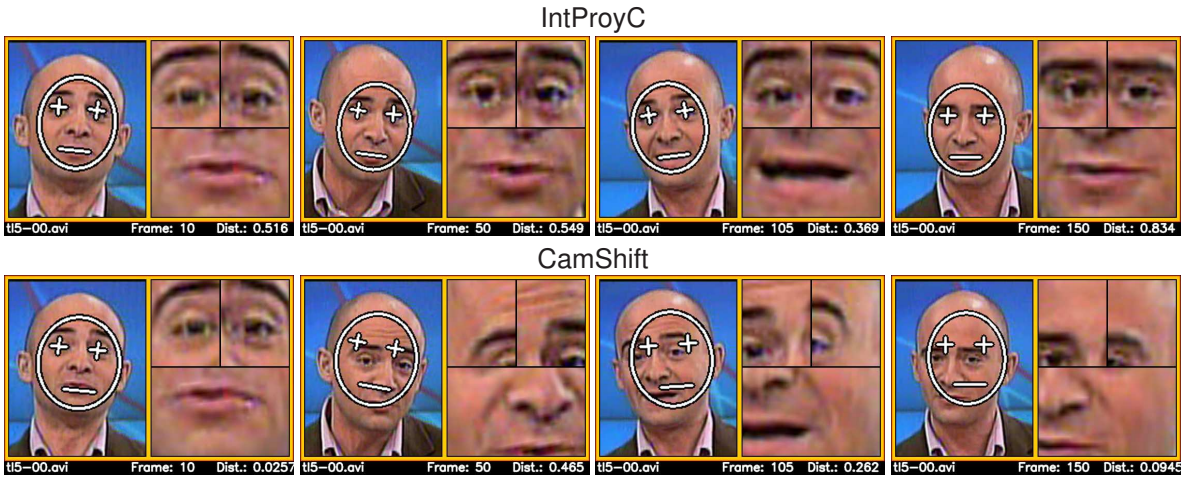


Figura 5.23: *Ejemplos de resultados del seguimiento sobre la secuencia "tl5-00.avi". Arriba, posiciones obtenidas por el método basado en proyecciones con predicción mediante color. Abajo, resultados del algoritmo CamShift.*

para la localización de los componentes faciales. Pero en las otras técnicas denota una dificultad para mantener estable el seguimiento de los componentes.

Por ejemplo, se puede observar que la técnica de Lucas y Kanade genera una gran ambigüedad en sentido horizontal; la desviación en X es entre 3 y 4 veces mayor que en Y. Paradójicamente, la simple búsqueda de patrones consigue menores errores en todos los componentes sin un excesivo aumento del tiempo. Esto es debido a la progresiva degradación de las posiciones en LKTracker, el *drifting*: puesto que el algoritmo se aplica entre el frame t y el $t-1$, los pequeños errores se van acumulando a lo largo del tiempo. El problema se acrecienta con secuencias largas. Este hecho no ocurre con TemMatch, ya que los patrones de ojos y boca son los obtenidos en el instante inicial. Por su parte, el seguidor basado en contornos presenta el mismo tipo de dificultades que las que aparecían en las pruebas del capítulo anterior: alta imprecisión, ambigüedad cejas/ojos, nariz/boca, etc.

Comparación de la precisión con giros y movimientos

El mismo experimento del punto anterior, orientado al estudio de la precisión, ha sido repetido sobre la secuencia "ggm2.avi", que incluye un mayor número de situaciones complejas, fundamentalmente giros laterales, verticales y movimientos. Los datos de la secuencia fueron expuestos en la tabla 5.2.

En este caso, los resultados obtenidos no sólo están influidos por la precisión de las técnicas, sino también por la robustez frente a las variaciones mencionadas. Esto hace que los errores cometidos sean en general mucho mayores, aunque no se pierdan caras en el seguimiento. Los valores alcanzados por los seguidores se muestran en la tabla 5.6.

Las gráficas de localizaciones de esta prueba se encuentran en la figura 5.24.

Globalmente, podemos confirmar las mismas conclusiones extraídas para las secuencias de televisión. Los algoritmos basados en proyecciones alcanzan excelentes niveles de pre-

Método	Ratio segui.	Ratio f.pos.	Desv. ojo izq. Total	Desv. ojo izq. X / Y	Desv. ojo der. Total	Desv. ojo der. X / Y	Desv. boca Total	Desv. boca X / Y	Tmp. (ms)
IntProyN	**100**	**0**	**6,1**	**5,2**/3,3	**6,7**	**5,6**/**3,7**	**10,7**	**6,1**/8,8	8,2
IntProyC	**100**	**0**	6,3	5,5/**3,2**	7,3	6,2/3,9	12,2	7,1/9,9	13,2
LKTracker	**100**	**0**	11,6	6,4/9,7	17,4	13,0/11,5	11,8	7,9/**8,7**	**6,5**
TemMatch	**100**	**0**	10,2	6,8/7,6	12,4	9,2/8,3	41,7	21,7/35,6	14,7
CamShift	70,2	29,8	20,6	14,0/15,1	25,2	17,1/18,5	36,2	28,6/22,2	10,4
Cont	97,9	0,2	10,3	5,6/8,6	11,8	7,5/9,1	29,5	16,8/24,3	41,3

Tabla 5.6: *Resultados del seguimiento sobre la secuencia "ggm2.avi" para distintos métodos de seguimiento. La secuencia es descrita en la tabla 5.2. Las medidas de desviación estándar están en proporción a la distancia interocular.*

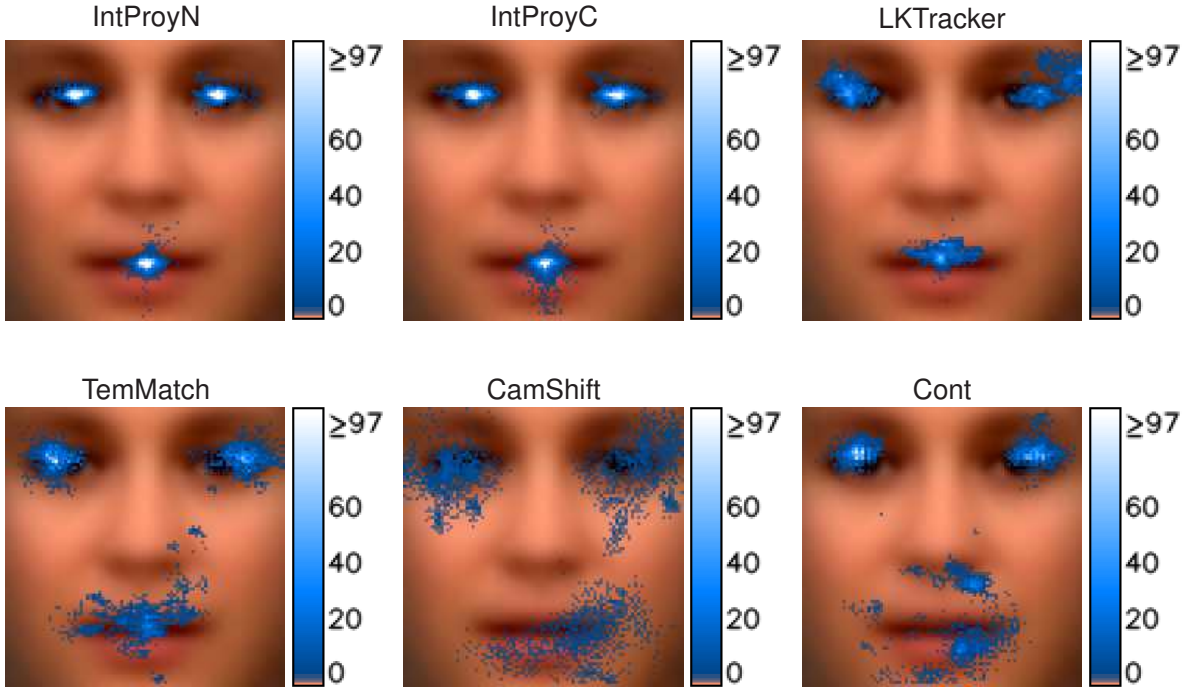


Figura 5.24: *Localizaciones resultantes de los distintos seguidores para el vídeo propio: "ggm2.avi". Se muestran las densidades de puntos localizados en cada posición de una cara estándar.*

cisión, aunque menores que los documentados en la tabla 5.5. Por ejemplo, los errores para los ojos de IntProyN e IntProyC son alrededor de la mitad que los producidos por las restantes técnicas. No obstante, las gráficas de la figura 5.24 evidencian que la imprecisión de los ojos es mayor en el eje X que en Y. Lo contrario sucede para la boca.

También vuelve a ocurrir que TemMatch supera ligeramente a LKTracker en la localización de los ojos. No obstante, su imprecisión para la boca es mucho mayor, ya que son numerosos los casos donde se coloca erróneamente en la nariz o en la barbilla. El seguidor LKTracker presenta nuevamente problemas de deriva progresiva en las posiciones de los componentes, como se puede deducir de la figura 5.24.

Evolución temporal del seguimiento

Todos los métodos de seguimiento analizados ofrecen una medida de fiabilidad del resultado conseguido (residuos, correlación, distancias a los modelos de proyección, etc.). En relación a ella, se fija un umbral para decidir cuándo se ha perdido la cara. Pero la medida también puede ser mala cuando el seguimiento es difícil, causando la pérdida de una cara que estaba siendo seguida de forma correcta.

Para analizar el comportamiento del seguidor basado en proyecciones, en la figura 5.25 mostramos su evolución a lo largo del tiempo para la secuencia "ggm2.avi". Esta gráfica debe ser comparada con la de la figura 5.19, donde ocurren cambios de plano que originan la desaparición de las caras. Añadimos la representación del error en los ojos para cada instante. Se muestra comparativamente la misma evolución para el seguidor LKTracker.

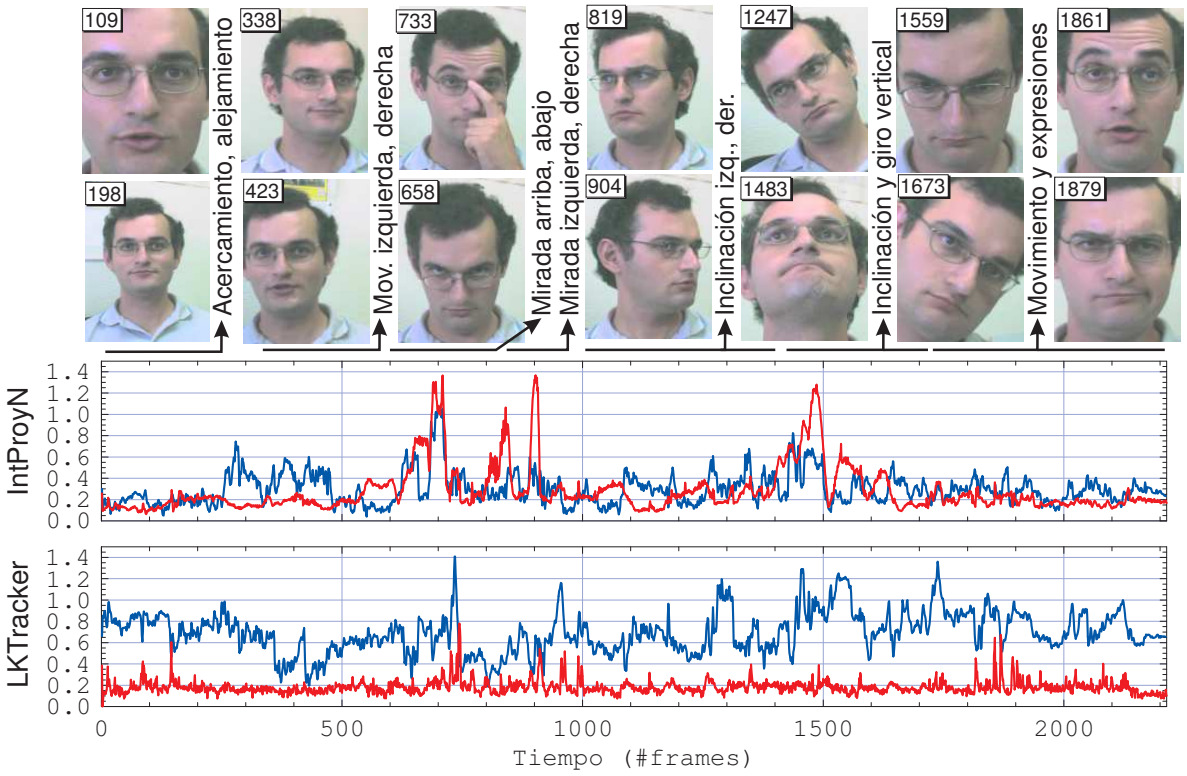


Figura 5.25: *Evolución temporal del seguimiento en la secuencia de webcam "ggm2.avi". En la parte superior aparece una descripción del contenido de la secuencia, y algunos extractos de los fragmentos más significativos (se indica el número de frame). En la parte inferior, los errores de localización de los ojos (en azul) y la distancia o fiabilidad (en rojo) devuelta por el seguidor basado en proyecciones (arriba) y el método de Lucas y Kanade (abajo). Los errores medios en los ojos se miden en proporción a la distancia interocular, y están divididos por 10.*

Es interesante observar que existe una cierta correlación entre el error en los ojos y la medida de fiabilidad de IntProy. Recordemos que esta medida se obtiene mediante la suma de los errores de alineamiento de PV_{cara} y PH_{ojos}. Siempre que el error existente toma valores grandes, el parámetro devuelto por el algoritmo aumenta consecuentemente. En condiciones normales, la medida de fiabilidad está por debajo de 0,6. Un valor mayor indica que, aunque la

cara puede ser seguida, existe un cambio de apariencia que dificulta el seguimiento. Aun así, los valores en casos de desaparición se encuentran casi siempre muy por encima. Obsérvese, por ejemplo, que en la figura 5.19 todas las desapariciones producen como mínimo una distancia de 1,8. En consecuencia, es sencillo establecer un umbral para detectar la pérdida, sin cortar el seguimiento de un caso complejo.

La decisión resulta más difícil para el seguidor LKTracker, donde no está tan clara la relación entre el error existente y la medida de fiabilidad. El inconveniente que subyace es que sólo se tiene en cuenta la información del instante anterior. Si no se ha podido detectar la desaparición, el proceso acabaría siguiendo una no cara indefinidamente, o hasta que se volviera a aplicar la detección/localización periódica.

5.4.3. Medidas de eficiencia computacional

Ya hemos señalado la importancia de reducir el coste computacional en los algoritmos de seguimiento. En la práctica, el procesamiento *off-line* de secuencias es poco habitual, y la mayoría de las aplicaciones del seguimiento de caras requieren manejar vídeo en tiempo real. La ejecución del seguidor no sólo debería permitir ratios del orden de los 30 *frames* por segundo, sino que debe dejar tiempo adicional para otros procesos.

Para analizar el coste de ejecución de las alternativas disponibles, utilizamos diversos vídeos con resoluciones típicas. Sobre estos vídeos se aplican los seguidores, midiendo los tiempos requeridos para cada *frame*; las medidas incluyen la lectura y descompresión de los archivos de vídeo en formato AVI[12]. El ordenador utilizado es un Pentium IV a 2,60GHz.

La tabla 5.7 resume los tiempos medios de los métodos de seguimiento en función del tamaño de las imágenes del vídeo. Debemos aclarar que en este caso no se incluye el tiempo consumido en los procesos de detección/localización de las caras, que pueden ser aplicados con mayor o menor periodicidad, de acuerdo con la política de seguimiento.

Tamaño (píxeles)	**T. lectura (ms)**	**Tiempo de seguimiento por frame (ms)**					
		IntProyN	**IntProyC**	**LKTracker**	**TemMatch**	**CamShift**	**Cont**
160×120	0,9	4,1	5,3	**2,0**	2,4	2,3	25,1
320×240	4,7	8,2	13,2	**8,1**	10,3	10,0	31,1
640×480	16,4	**19,8**	43,2	22,2	28,2	41,1	54,8

Tabla 5.7: *Tiempo de ejecución medio de los seguidores según el tamaño de las imágenes. La entrada en todos los casos son archivos en formato AVI, con una cara en cada frame. Los tiempos indicados incluyen la lectura del archivo de vídeo, que se muestra en la columna "T. lectura".*

En la figura 5.26 se representan gráficamente estos tiempos. Se añade información adicional sobre los tiempos máximos y mínimos de cada ejecución concreta.

Señalamos algunas conclusiones relevantes sobre estos resultados:

1. En general, casi todas las técnicas admiten frecuencias teóricas por encima de los 100 fps para una resolución habitual de 320 × 240 píxeles. El único método que está claramente

[12] En cualquier caso, se indican también esos tiempos de lectura para los ejemplos de prueba.

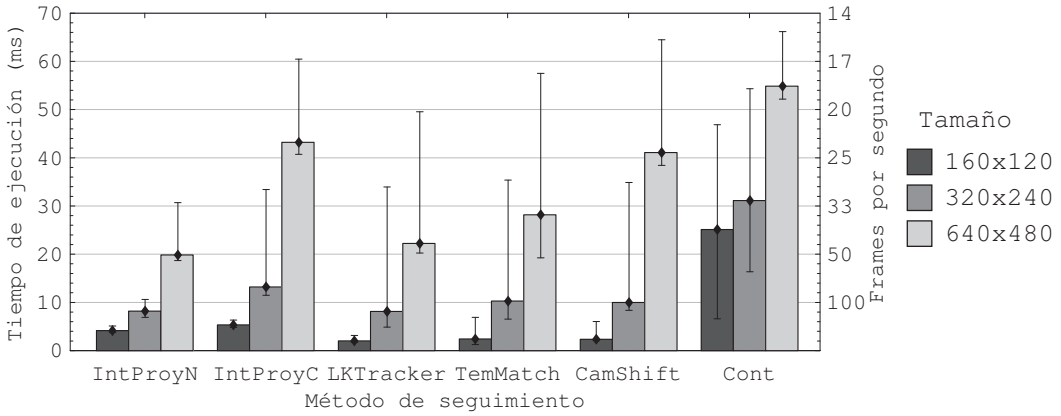


Figura 5.26: *Tiempos de ejecución máximo, mínimo y promedio de los algoritmos de seguimiento de caras, en función del tamaño de las imágenes de la secuencia, con 1 cara por imagen.*

por encima de la media es el basado en contornos. El algoritmo más rápido es el de Lucas y Kanade –recordemos que se usa una implementación piramidal muy eficiente–, seguido de cerca por IntProyN y TemMatch. Hay que tener en cuenta que en caso de utilizar captura de cámara los tiempos de lectura serán distintos (posiblemente algo mayores).

2. Si nos fijamos en la variación del tiempo en función del tamaño de la imagen, descontando los tiempos de lectura, podemos apreciar que IntProyN añade siempre unos 4 milisegundos al coste de obtener la imagen. Es decir, no depende de la resolución de entrada. Esto es debido a que el primer paso del proceso extrae la cara a una imagen de tamaño estándar y reducido, sobre la que se aplican los cálculos posteriores. Gracias a ello, consigue mejorar la eficiencia de LKTracker para tamaños grandes.

3. Los tiempos de IntProyC están limitados, lógicamente, por los de CamShift, en cuanto que el segundo se usa como mecanismo de predicción del primero. Teniéndolo en cuenta, la aplicación de la relocalización mediante proyecciones supone añadir unos 3 milisegundos al coste total del proceso. Para tamaños grandes, la mayor parte de la carga es debida a la transformación de RGB a HSV. Una posible estrategia para reducir este requisito es escalar previamente las imágenes: la relocalización trabajaría con la resolución original, pero el algoritmo CamShift puede ser aplicado sobre versiones más pequeñas. De forma estimativa, si las imágenes de 640×480 se reducen a la mitad, el tiempo de IntProyC podría situarse fácilmente sobre los 25 milisegundos.

El coste de los procesos de detección y localización no debe ser obviado, ya que normalmente es mucho mayor que el de seguimiento propiamente dicho. Esos tiempos no se contemplan en la tabla 5.7, aunque sí en las restantes pruebas. Es más, puesto que la detección se aplica al perder el seguimiento, el incremento afectará de forma desigual a los diferentes métodos. De hecho, más adelante veremos un caso donde LKTracker se sitúa por detrás de IntProyC debido a las numerosas pérdidas de la cara seguida.

5.4.4. Robustez frente a resolución, movimientos y expresiones

Cualquier fuente de variación en la apariencia de las caras, llevada a un extremo, puede conducir a un fallo del proceso de seguimiento. Ya hemos analizado el comportamiento de los diferentes métodos en condiciones típicas de trabajo, con grados de variación moderados. En este apartado vamos a estudiar la robustez de los seguidores frente a ejemplos más complejos de expresión facial, iluminación, baja resolución de entrada y movimientos rápidos de la cabeza. Para cada caso usamos las secuencias más significativas.

Robustez frente a expresiones faciales e iluminación

Todas las secuencias introducidas hasta ahora exhiben variaciones naturales, y nada artificiosas, de la expresión facial. Sin embargo, en este experimento estamos interesados en poner a prueba los diferentes métodos frente a gestos más exagerados. En concreto, manejamos dos vídeos, uno propio y el otro presentado en el artículo de Buenaposada y otros [19] y disponible públicamente. El segundo incluye, además, una fuerte modificación en las condiciones de iluminación de la escena, que genera la aparición de sombras y situaciones de iluminación deficiente. La tabla 5.8 describe ambas secuencias.

Secuencia	Fuente	Resoluc.	Duración	Compresión	Variación
ggm4.avi	Logitech QuickCam Pro 5000	640 × 480 15 fps	32,6 s 489 frames	3ivX D4 4.0.4 19,46 Kbytes/s	Expresión, movimiento
case2.avi	N.D.	320 × 240 15 fps	1 min 7 s 1008 frames	JPEG ∼10 Kbytes/frame	Expresión, iluminación

Tabla 5.8: *Descripción de las secuencias de prueba de la expresión facial, "ggm4.avi" y "case2.avi". La primera está disponible públicamente en:* `http://dis.um.es/profesores/ginesgm/fip`*. La segunda se puede encontrar en:* `http://www.dia.fi.upm.es/~pcr/face_tracking.html`*.*

En la secuencia "ggm4.avi" se aplica el detector combinado Haar+IP –con una reducción previa de las imágenes a la mitad de su tamaño– y el localizador mediante proyecciones. Por su parte, en "case2.avi" el detector es IP+Haar –que consigue resultados ligeramente mejores– y el localizador es el basado en redes neuronales; la existencia de sombras hace que la localización de los ojos con proyecciones sea imprecisa, por lo que se usa esa alternativa. En todos los seguidores, el umbral para dar por finalizado el seguimiento se fija a valores altos, para evitar posibles pérdidas prematuras de la cara.

Los resultados de este estudio se encuentran en la tabla 5.9. Además de los ratios de seguimiento, de falsos positivos, y de la desviación en los componentes, se añaden dos medidas adicionales. Por un lado, mostramos el **número de cortes** del proceso, es decir, el número de veces que se arranca y se pierde el seguimiento. Idealmente, para las dos secuencias de prueba, debería tomar valor 1, indicando que existe un solo tramo de seguimiento. Por otro lado, añadimos un hipotético seguidor de caras –denominado "Detector"– consistente en aplicar los algoritmos de detección y localización independientemente para todos los *frames*. El objetivo es medir el grado de fiabilidad y precisión de los mismos; lógicamente, el número de

cortes aquí coincide con la cantidad total de detecciones.

Secuencia: "ggm4.avi"

Método	Número de cortes	Ratio seguim.	Ratio f.pos.	Desviación Ojo izq.	Desviación Ojo der.	Desviación Boca	Tmp. (ms)
Detector	432	432 (88,3 %)	0 (0,0 %)	4,5	4,3	8,9	180,9
IntProyN	**1**	478 (97,8 %)	11 (2,2 %)	**5,6**	4,5	**6,2**	**21,3**
IntProyC	**1**	**484 (99,0 %)**	4 (0,8 %)	5,8	4,8	7,5	51,5
LKTracker	5	444 (90,8 %)	**1 (0,2 %)**	6,8	**4,4**	41,0	35,4
TemMatch	**1**	460 (94,1 %)	29 (5,9 %)	6,5	5,3	21,0	38,7
CamShift	**1**	224 (45,8 %)	265 (54,2 %)	12,7	10,8	15,0	44,9
Cont	**1**	471 (96,3 %)	18 (3,7 %)	6,0	5,8	19,0	72,9

Secuencia: "case2.avi"

Método	Número de cortes	Ratio seguim.	Ratio f.pos.	Desviación Ojo izq.	Desviación Ojo der.	Desviación Boca	Tmp. (ms)
Detector	989	882 (87,5 %)	107 (10,6 %)	8,4	6,4	13,6	249,2
IntProyN	**1**	969 (96,1 %)	38 (3,8 %)	8,6	**7,1**	11,3	7,4
IntProyC	**1**	996 (98,8 %)	12 (1,2 %)	8,8	7,6	**10,7**	14,5
LKTracker	**1**	**998 (99,0 %)**	**10 (1,0 %)**	7,9	9,7	35,7	**6,7**
TemMatch	**1**	964 (95,6 %)	44 (4,4 %)	**5,6**	9,0	45,0	16,9
CamShift	**1**	881 (87,4 %)	126 (12,5 %)	12,3	9,9	12,0	12,3
Cont	6	439 (43,6 %)	512 (50,8 %)	16,1	12,8	40,9	85,3

Tabla 5.9: *Resultados del seguimiento con grandes variaciones de la expresión facial. Las secuencias son descritas en la tabla 5.8. Las medidas de desviación estándar están en proporción a la distancia interocular.*

La figura 5.27 muestra algunos ejemplos comparativos sobre la secuencia "ggm4.avi". Se omite el método IntProyC, cuyos resultados son prácticamente idénticos a los de IntProyN. Se incluyen varios casos de oclusión parcial –gesto de quitar y poner las gafas–, en los cuales se pierden algunos algoritmos de seguimiento.

Por su parte, la figura 5.28 presenta varios ejemplos del seguidor IntProyC para la secuencia "case2.avi", a intervalos de 100 *frames*. En la parte inferior se pueden apreciar algunos resultados extraídos de [19], en el que se contrastan varios seguidores basados en apariencia sobre la secuencia mencionada.

Globalmente, los algoritmos basados en proyecciones siguen manteniendo un buen comportamiento, tanto en los ratios de seguimiento como en la precisión de las localizaciones obtenidas. Ambos producen el mínimo número de cortes para ambas secuencias. En ciertos casos se llega a aplicar la detección/localización; pero al encontrarse la cara de forma inmediata, no se produce un nuevo corte[13]. Esto no sucede con LKTracker, que provoca un número muy elevado de pérdidas para la secuencia "ggm4.avi"; fundamentalmente –aunque no únicamente– son debidas a los casos de oclusión parcial.

Para la prueba "case2.avi" el método de Lucas y Kanade resulta el más fiable, seguido de cerca por IntProyC; si bien la precisión de LKTracker para la boca es muy escasa, ya que se

[13]Es decir, el detector de caras consigue recuperar la pérdida momentánea del seguimiento, sin que se llegue a dejar de encontrar el rostro del sujeto en ningún momento.

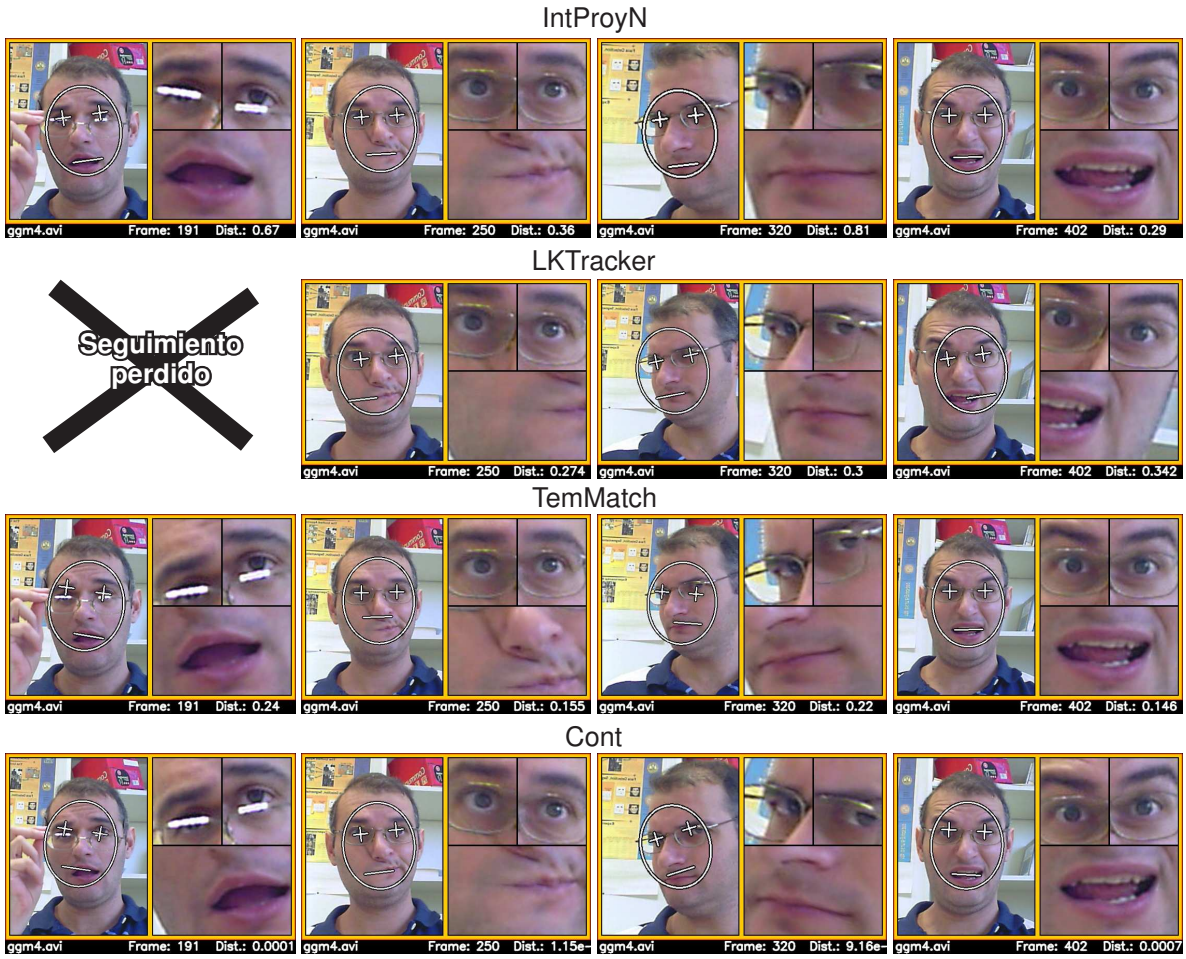


Figura 5.27: *Ejemplos de resultados del seguimiento sobre la secuencia "ggm4.avi". De arriba abajo, se muestran los resultados de los métodos IntProyN, LKTracker, TemMatch y Cont.*

mueve de forma incontrolada por la cara.

Es destacable el comportamiento del algoritmo basado en contornos, que funciona muy bien para el vídeo "ggm4.avi", pero fracasa rotundamente para "case2.avi". La razón se encuentra en la existencia de sombras y cambios de iluminación en el segundo ejemplo, que imposibilita la extracción fiable de contornos asociados a los componentes faciales de interés.

Por otro lado, en comparación con los seguidores basados en apariencia documentados en [19], los resultados de IntProyN e IntProyC sobre la secuencia "case2.avi" son bastante positivos. Tanto el método de Hager y Belhumeur [74], como el de Matthews y Baker [121], provocan inestabilidades, como se puede apreciar en la parte inferior de la figura 5.28. Es más, el primero pierde el seguimiento durante una tercera parte de la secuencia. La imprecisión de ambos también es elevada frente a gestos como levantar las cejas o cerrar los ojos.

En las técnicas basadas en proyecciones, la aparición de sombras (principalmente en los lados de la nariz) da lugar a una ligera desviación hacia el lado donde ocurre la sombra, como sucede en los *frames* 300, 600 y 800 de la figura 5.28. Además, el algoritmo subyacente de alineamiento de proyecciones debe trabajar con márgenes de tolerancia pequeños, no superiores al

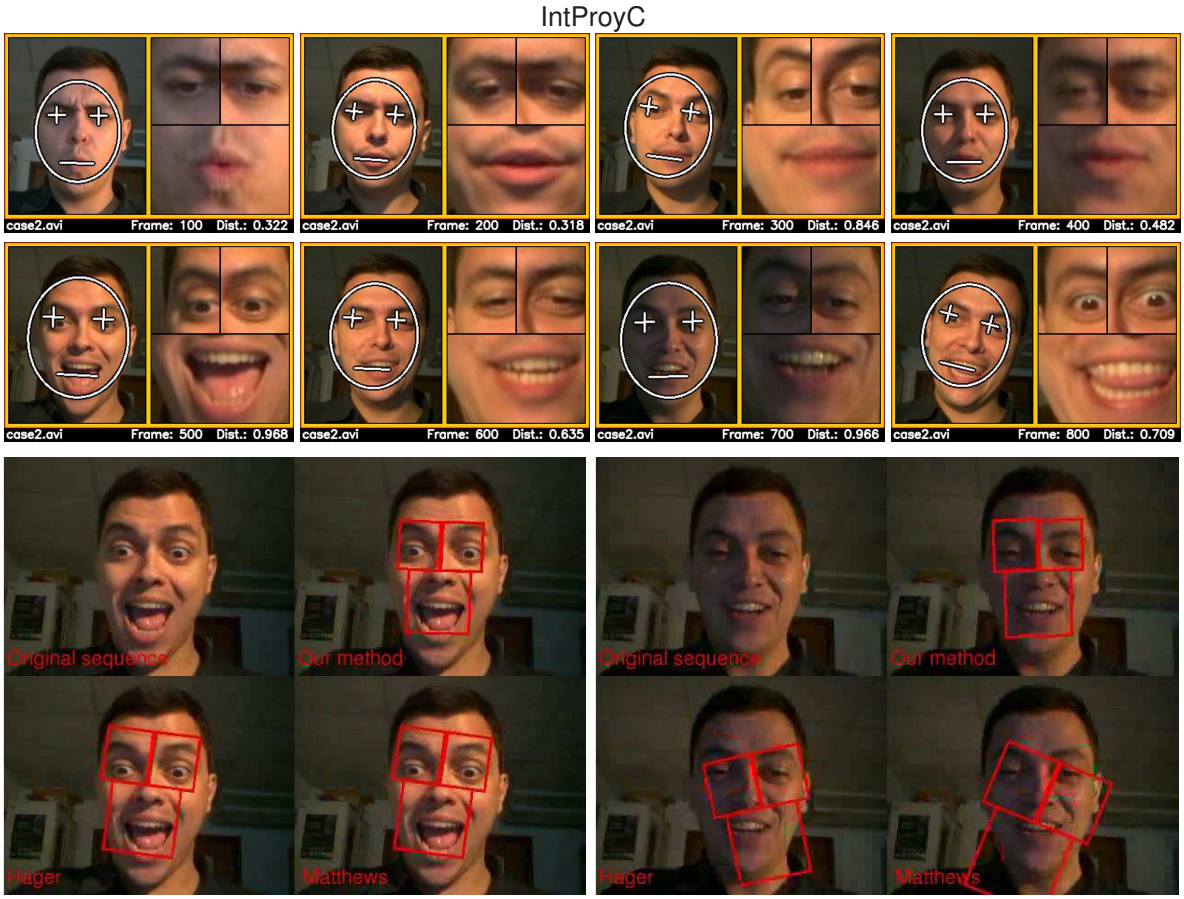


Figura 5.28: *Ejemplos de resultados del seguimiento sobre la secuencia "case2.avi". En la parte superior, se muestran los resultados de la técnica IntProyC, en intervalos de 100 frames. En la parte inferior, extractos de la secuencia original y varios métodos basados en apariencia: Buenaposada y otros [19] (etiquetado como "our method"), Hager y otros [74], y Matthews y otros [121].*

10 %. Lógicamente, el método propuesto por Buenaposada y otros [19], diseñado y entrenado específicamente para manejar este tipo de dificultades, resulta el más fiable e invariante a las sombras que aparecen en la secuencia "case2.avi".

Evaluación de la robustez con baja resolución de entrada

La falta de resolución y calidad de la fuente de captura puede provocar una completa pérdida de detalle en los rostros que están siendo seguidos, como puede verse en la figura 5.29. Sin embargo, una aplicación orientada a usuarios domésticos debe ser capaz de abordar la posible existencia de este inconveniente. En tales condiciones, el objetivo de la robustez prevalece frente a la precisión del mecanismo.

El conjunto de secuencias utilizadas para este experimento, que denominaremos la base NRC-ITT, está documentado en [70, 69]. Originalmente los vídeos fueron creados para la evaluación de sistemas de reconocimiento de personas en vídeo (FRiV); así, en [70, 69], las caras

son extraídas mediante la aplicación del detector de Haar sobre cada *frame* por separado. No obstante, el uso de los procesos de seguimiento puede aumentar el número de caras encontradas, reduciendo al mismo tiempo el coste computacional.

La base NRC-ITT consta en total de 22 vídeos de 11 personas (2 archivos por persona). Para nuestras pruebas tomamos las 12 primeras secuencias. Los datos de las mismas se pueden ver en la tabla 5.10. Todas ellas están disponibles en la web.

Secuencias	Fuente	Resolución	Duración total	Compresión	Variación
00-1.avi 00-2.avi	Webcam	160 × 120 20 fps	23,85 s 479 frames	Intel Indeo v5.10 4,84 Kbytes/s	Movim., giros, oclusión
01-1.avi 01-2.avi	″	160 × 120 20 fps	28,3 s 568 frames	Intel Indeo v5.10 5,32 Kbytes/s	Expres., giros, desaparición
02-1.avi 02-2.avi	″	160 × 120 20 fps	29,8 s 598 frames	Intel Indeo v5.10 5,49 Kbytes/s	Movimiento, inclinación
03-1.avi 03-2.avi	″	160 × 120 20 fps	44,3 s 888 frames	Intel Indeo v5.10 4,58 Kbytes/s	Giros, oclusión, escasa resoluc.
04-1.avi 04-2.avi	″	320 × 240 20 fps	37,85 s 759 frames	Intel Indeo v5.10 8,96 Kbytes/s	Giros 3D, posición
05-1.avi 05-2.avi	″	160 × 120 20 fps	22,3 s 448 frames	Intel Indeo v5.10 5,12 Kbytes/s	Movimiento, escasa resoluc.

Tabla 5.10: *Descripción de las secuencias de prueba de la base NRC-ITT [70, 69]. La información está agrupada por parejas. Por ejemplo, "00-1.avi" consta de 229 frames y "00-2.avi" de 250. Los vídeos están disponibles públicamente en:* `http://synapse.vit.iit.nrc.ca/db/video/faces/cvglab`.

A grandes rasgos, la base NRC-ITT contiene muestras de los principales inconvenientes en el procesamiento de caras en vídeo: escasa resolución, desenfoque, inclinación y giros fuera del plano de imagen, expresiones faciales, oclusiones, desaparición, movimientos rápidos. La figura 5.29 presenta algunos extractos representativos del conjunto.

Figura 5.29: *Ejemplos de imágenes de las secuencias usadas de la base NCR-ITT [70, 69]. La resolución de las imágenes es de* 160 × 120 *píxeles, excepto en "04-{1,2}.avi" que es de* 320 × 240.

Para la realización de esta prueba utilizamos el detector de Haar, que es ajustado en un modo de operación permisivo, es decir, con alto número de detecciones. Es más, en algunos casos se deben bajar los umbrales del detector, ya que resulta muy difícil encontrar las caras. Las principales medidas aquí son el ratio de seguimiento y el número de falsas detecciones.

De forma excepcional, la diferencia máxima para declarar un resultado como correcto o como falso positivo se fija en el 40 % de la distancia interocular.

La tabla 5.11 contiene los resultados de los diferentes vídeos, agrupados de dos en dos (se ponen juntos los correspondientes al mismo individuo). Como en el estudio anterior, se han incluido los resultados de aplicar repetidamente el detector en todas las imágenes de las secuencias.

	Secuencias: 00-1.avi, 00-2.avi			Secuencias: 01-1.avi, 01-2.avi		
Método	**Seguidas**	**Falsos pos.**	**Tiempo**	**Seguidas**	**Falsos pos.**	**Tiempo**
Detector	318 (66,7 %)	24 (5,0 %)	23,5	136 (24,6 %)	118 (20,8 %)	20,6
IntProyN	**445 (93,5 %)**	**5 (1,1 %)**	6,6	458 (83,0 %)	55 (9,7 %)	7,2
IntProyC	435 (91,4 %)	26 (5,5 %)	7,2	**485 (87,9 %)**	**27 (4,8 %)**	8,1
LKTracker	408 (85,7 %)	57 (12,0 %)	**2,4**	447 (81,0 %)	83 (14,7 %)	**3,9**
TemMatch	421 (88,4 %)	20 (4,2 %)	6,7	218 (39,5 %)	321 (56,8 %)	7,5
CamShift	242 (50,8 %)	239 (50,2 %)	3,7	186 (33,7 %)	364 (64,4 %)	6,2

	Secuencias: 02-1.avi, 02-2.avi			Secuencias: 03-1.avi, 03-2.avi		
Método	**Seguidas**	**Falsos pos.**	**Tiempo**	**Seguidas**	**Falsos pos.**	**Tiempo**
Detector	353 (59,2 %)	49 (8,2 %)	21,7	464 (53,0 %)	97 (10,9 %)	35,8
IntProyN	425 (71,3 %)	182 (30,5 %)	7,9	685 (78,4 %)	89 (10,1 %)	9,3
IntProyC	507 (85,1 %)	**43 (7,2 %)**	8,1	**706 (80,8 %)**	**68 (7,7 %)**	10,8
LKTracker	**532 (89,3 %)**	64 (10,7 %)	**2,7**	610 (69,8 %)	248 (28,1 %)	**3,1**
TemMatch	407 (68,3 %)	158 (26,5 %)	6,9	577 (66,0 %)	243 (27,5 %)	6,0
CamShift	370 (62,1 %)	225 (37,8 %)	3,8	477 (54,6 %)	407 (46 %)	3,4

	Secuencias: 04-1.avi, 04-2.avi			Secuencias: 05-1.avi, 05-2.avi		
Método	**Seguidas**	**Falsos pos.**	**Tiempo**	**Seguidas**	**Falsos pos.**	**Tiempo**
Detector	430 (57,3 %)	**41 (5,4 %)**	132,9	331 (74,2 %)	39 (8,7 %)	23,0
IntProyN	507 (67,6 %)	138 (18,2 %)	14,4	373 (84,0 %)	24 (5,4 %)	8,5
IntProyC	**529 (70,5 %)**	133 (17,6 %)	21,7	**388 (87,4 %)**	**3 (0,7 %)**	9,4
LKTracker	478 (63,7 %)	216 (28,5 %)	13,2	353 (79,5 %)	38 (8,6 %)	**5,1**
TemMatch	399 (53,2 %)	191 (25,2 %)	32,8	365 (82,2 %)	14 (3,2 %)	8,1
CamShift	371 (49,5 %)	378 (49,9 %)	**11,1**	259 (58,3 %)	150 (33,8 %)	6,6

Tabla 5.11: *Resultados individuales del seguimiento para las secuencias de la base NCR-ITT [70, 69]. Las secuencias son descritas en la tabla 5.10. Se indica el número total de caras seguidas, de falsos positivos, y los porcentajes en relación al total.*

Los resultados globales del experimento aparecen en la tabla 5.12. La figura 5.30 representa gráficamente los ratios de seguimiento y de falsas alarmas de cada seguidor. En la figura 5.31 se pueden ver algunos ejemplos de cómo la técnica desarrollada puede funcionar en condiciones muy adversas de oclusión parcial y baja calidad de imagen.

Como era de esperar, los ratios de seguimiento caen en relación a los mostrados para los anteriores experimentos. El mejor método, IntProyC, no pasa del 84 % con un alto número de falsos positivos. Incluso, en "04-{1,2}.avi" se pierde el 30 % de las instancias existentes. El hecho de que el detector no llegue al 56 % total de detección es una muestra evidente de la dificultad implícita de esta prueba.

A pesar de ello, los métodos basados en proyecciones consiguen aumentar siempre el número de caras que encontraría el detector de Haar por sí solo; y en casi todos los casos su-

Método	Caras seguidas	Falsos positivos	Tiempo (ms)
Detector	2032 (55,8 %)	368 (9,8 %)	42,9
IntProyN	2893 (79,6 %)	493 (12,5 %)	8,9
IntProyC	**3050 (83,9 %)**	**300 (7,3 %)**	10,8
LKTracker	2828 (78,2 %)	706 (17,1 %)	**5,1**
TemMatch	2387 (66,3 %)	947 (23,9 %)	11,3
CamShift	1905 (51,5 %)	1763 (47,0 %)	5,8

Tabla 5.12: *Resultados totales de los seguidores para las secuencias utilizadas de la base NCR-ITT. Los tiempos y los porcentajes son el promedio de los 6 valores de la tabla 5.11.*

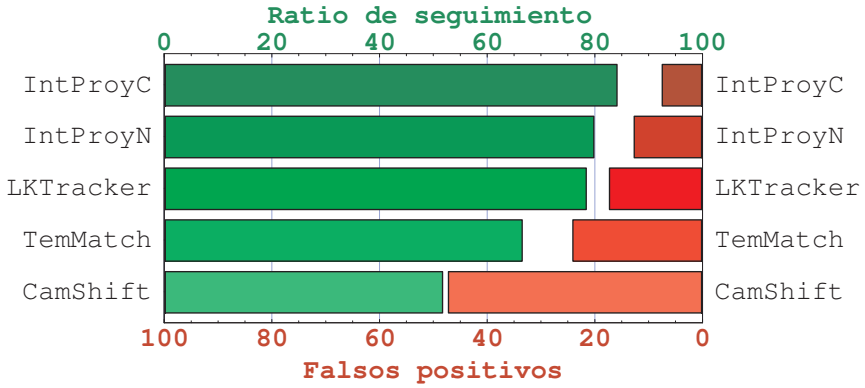


Figura 5.30: *Ratios de seguimiento (en verde) y de falsas alarmas (en rojo) de los seguidores, para las secuencias de la base de vídeos NRC-ITT. Los valores concretos se pueden consultar en la tabla 5.12.*

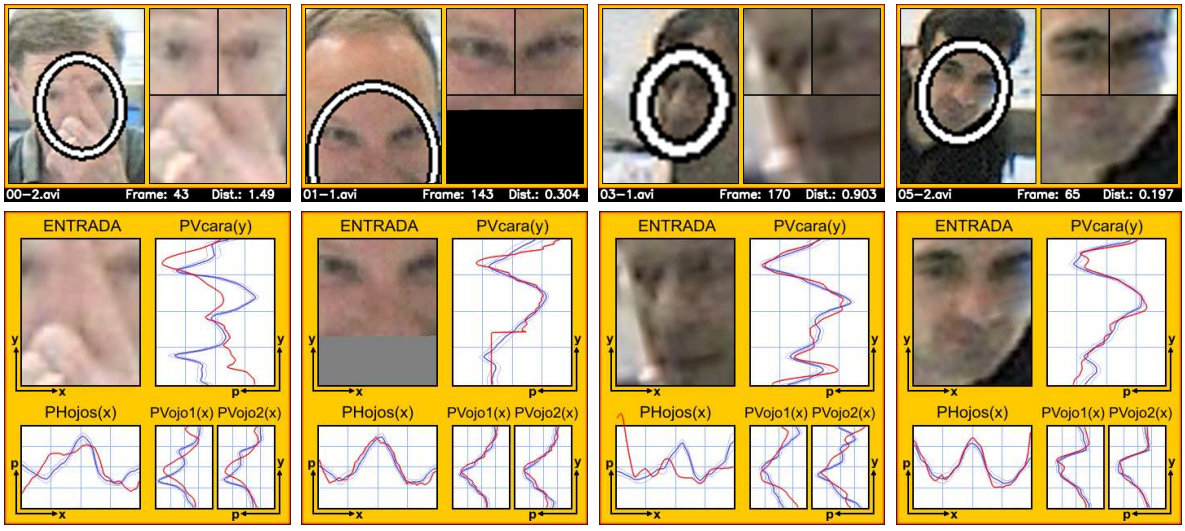


Figura 5.31: *Ejemplos de seguimiento en IntProyN con oclusión y baja calidad de imagen, en secuencias de la base NRC-ITT. De izquierda a derecha, fragmentos de "00-2.avi", "01-1.avi", "03-1.avi" y "05-2.avi". En la parte superior, extractos de la secuencia con las posiciones seguidas por el algoritmo. En la parte inferior, la entrada del paso de relocalización y las proyecciones calculadas en el proceso.*

peran a los restantes sistemas. Sólo en el par "02-{1,2}.avi" LKTracker obtiene un mejor ratio de seguimiento que IntProyN e IntProyC. Además, otra ventaja de ambos es que producen un bajo número de falsos seguimientos. Por ejemplo, si nos fijamos en la figura 5.30, los por-

centajes totales de caras seguidas en IntProyN y LKTracker son muy parecidos; sin embargo, el primero genera un 30 % menos de posiciones de no caras. En cuanto a los tiempos de ejecución, al tener que aplicar la detección/localización muchas veces, las diferencias entre los métodos más rápidos y más lentos se reducen.

Aunque los porcentajes de caras seguidas son escasos, el número de cortes de las secuencias no suele ser muy elevado, permitiendo un seguimiento continuado para tramos donde no ocurren variaciones extremas. Por ejemplo, en "00-1.avi" con IntProyC sólo suceden 2 cortes de la secuencia, y 3 cortes en "00-2.avi". Es más, muchas veces los falsos positivos corresponden a un error en la posición de los ojos que supera ligeramente el 40 % establecido, más que al seguimiento de una falsa cara como tal.

El método basado en contornos ha sido excluido de este experimento por su mal comportamiento con la mayoría de las secuencias. A modo de ejemplo, en los vídeos "00-{1,2}.avi" el ratio de seguimiento es de sólo un 35 % para un 63 % de falsos positivos.

Evaluación de la robustez frente a movimientos rápidos

La adaptación a los movimientos rápidos y esporádicos del usuario es otra de las características deseables de un buen seguidor. Normalmente, este factor está relacionado inversamente con la precisión obtenida: los métodos más precisos suelen ser los más limitados en la velocidad de la cara. Una estrategia mixta puede resultar la elección más acertada para estas situaciones. Recordemos que la velocidad observada no sólo depende del movimiento de la cara y de la cámara, sino también del número de imágenes por segundo de la secuencia.

En la tabla 5.13 se detallan las características del vídeo usado en la evaluación de los seguidores frente al movimiento. Además del desplazamiento, existen variaciones de la expresión facial y un tramo donde desaparece el rostro, al salirse por un extremo de la imagen.

Secuencia	Fuente	Resol.	Duración	Compresión	Variación
ggm5.avi	Logitech QuickCam Pro 5000	640 × 480 15 fps	14,53 s 219 frames	3ivx D4 4.0.4 19,3 Kbytes/s	Mov. rápidos, expresión, oclusión

Tabla 5.13: *Descripción de la secuencia de prueba "ggm5.avi". El vídeo está disponible públicamente en:* `http://dis.um.es/profesores/ginesgm/fip`.

La velocidad media de la cabeza[14] en "ggm5.avi" es de unos 0,36 m/s, con un pico de unos 1,5 m/s. En comparación, la secuencia "ggm2.avi", por ejemplo, presenta un movimiento medio de 0,06 m/s, y en "case2.avi" no sobrepasa los 0,02 m/s, con un valor máximo de sólo 0,16 m/s. La figura 5.32 representa la evolución de la velocidad en "ggm5.avi" a lo largo del tiempo.

Los valores típicos de velocidad máxima están en torno a los 0,8 m/s. Esto significa que de un *frame* al siguiente la cara se puede desplazar hasta un 77 % de la distancia entre los ojos –suficiente para que un ojo se sitúe, más o menos, en la posición del otro ojo en el *frame*

[14]Esta velocidad ha sido estimada a partir del etiquetado manual, suponiendo que la distancia entre los ojos del sujeto es de unos 70 mm.

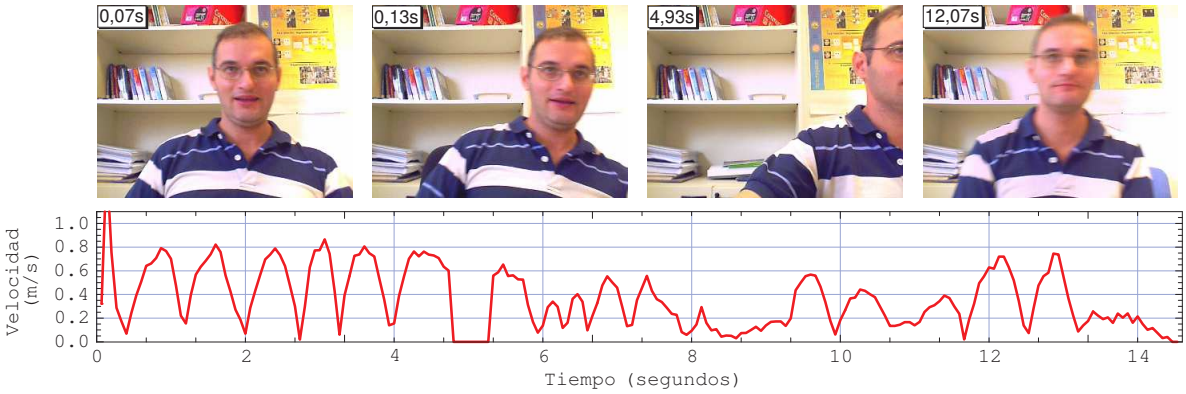


Figura 5.32: *Velocidades estimadas de la cara en la secuencia de prueba "ggm5.avi". En la parte superior se muestran algunas imágenes de la secuencia. La estimación de la velocidad supone que la distancia interocular es de 70 mm. La cara desaparece de la escena durante unos 0,5 s alrededor del segundo 5.*

anterior–. Otro problema asociado con la velocidad es la aparición de desenfoque por movimiento, como se puede ver en el ejemplo más a la derecha de la figura 5.32.

En todos los métodos de seguimiento, los parámetros relacionados con el radio de búsqueda de la cara han sido aumentados de forma significativa. Así, por ejemplo, en LKTracker se usan 5 niveles en la pirámide y tamaño de ventana de 18×18 píxeles (lo que permite un máximo desplazamiento teórico de 288 píxeles); en IntProy el tamaño de tolerancia en el algoritmo de alineamiento pasa a valer un 25 % del ancho de las señales.

El algoritmo de detección en este caso es Haar+IP, aplicado sobre las imágenes reducidas a la mitad del tamaño original. La localización, como en la mayoría de las restantes pruebas, utiliza el algoritmo basado en integrales proyectivas. En definitiva, los resultados del experimento se pueden consultar en la tabla 5.14. Se han añadido los métodos de predicción lineal básica y filtros de Kalman, con los parámetros de inercia ajustados a valores óptimos.

Método	Número de cortes	Ratio seguim.	Ratio f.pos.	Desviación Ojo izq.	Desviación Ojo der.	Desviación Boca	Tmp. (ms)
Detector	177	174 (82,5 %)	3 (1,4 %)	4,5	5,2	15,4	248,4
IntProyN	24	140 (66,4 %)	33 (15,1 %)	7,0	9,0	19,3	55,0
IntProyL(0,3)	18	160 (75,8 %)	12 (5,5 %)	6,5	10,0	18,5	54,5
IntProyK(0,1)	17	155 (73,5 %)	26 (11,9 %)	7,9	9,2	21,7	53,5
IntProyC	**2**	**202 (95,7 %)**	**6 (2,8 %)**	6,6	6,9	**7,0**	50,8
LKTracker	8	184 (87,2 %)	23 (10,6 %)	5,7	**6,6**	25,7	58,8
TemMatch	18	133 (63,0 %)	47 (21,6 %)	8,1	8,1	24,6	83,0
CamShift	1	91 (43,1 %)	120 (55,0 %)	17,9	19,1	21,8	**49,1**
Cont	21	66 (31,3 %)	112 (51,4 %)	**4,8**	7,7	23,1	92,7

Tabla 5.14: *Resultados del seguimiento sobre la secuencia "ggm5.avi" para distintos métodos de predicción. La secuencia es descrita en la tabla 5.13. Las medidas de desviación estándar están en proporción a la distancia interocular.*

Lo más destacable de los resultados de la tabla 5.14 es la enorme superioridad de IntProyC sobre el resto de métodos, tanto en los ratios de caras seguidas y falsos positivos, como en las medidas de precisión y los tiempos de ejecución. El seguimiento sólo se corta durante el

tramo en el que desaparece la cara, dando lugar al número óptimo de 2 cortes. El porcentaje de instancias encontradas es casi 10 puntos superior a las demás técnicas; y las falsas detecciones corresponden realmente a ligeros errores en la posición de los ojos. De forma colateral, los tiempos medios de ejecución de IntProyC se sitúan entre los más reducidos –de hecho, es el segundo método más rápido, sólo por detrás de CamShift– debido a que usa pocas veces el proceso de detección/localización.

Estos valores confirman lo que ya adelantamos. La estrategia mixta de seguimiento (color+proyecciones), consigue tomar lo mejor de ambos métodos: la adaptación a **movimientos rápidos** del primero, y la **precisión** del segundo. Con el esquema utilizado, los dos métodos se complementan; no es necesario renunciar a un buen ajuste del seguimiento cuando la cabeza se mueve con gran velocidad.

Merece la pena valorar el beneficio que aportan IntProyL e IntProyK sobre el mecanismo de predicción nula. La mejora se produce fundamentalmente en los tramos con velocidad más o menos uniforme. Pero estos trozos no son muchos ni muy largos, como se refleja en la figura 5.32; más bien, la velocidad cambia constantemente y con una gran aceleración. Así, en realidad, IntProyL sólo consigue encontrar 20 caras más que IntProyN. En cualquier caso, tanto el método lineal básico como el de Kalman mantienen un número muy elevado de cortes, que denota lo inadecuado del modelo de movimiento uniforme.

La buena precisión aparente del seguimiento con contornos resulta meramente casual. Al tener ratios de seguimiento tan bajos para un elevado número de cortes, las posiciones devueltas están muy próximas a la combinación de detector/localizador, que es más precisa por sí sola que cualquiera de los métodos de seguimiento analizados. A todos los efectos, el algoritmo basado en contornos es inviable en condiciones de movimientos rápidos.

Otros aspectos de interés en los seguidores

En los capítulos 3 y 4 comprobamos que, en el caso de las imágenes en color, el canal rojo es el que aporta más información para los problemas de detección de caras y localización de componentes faciales. La misma hipótesis se puede plantear también ahora al seguimiento. Para verificarla, repetimos la prueba de seguimiento sobre la secuencia "ggm4.avi", con IntProyN, tomando proyecciones de los canales R, G, B y del valor de intensidad. Los resultados se muestran en la tabla 5.15.

Canal utilizado	Número de cortes	Ratio seguim.	Ratio f.pos.	Desviación Ojo izq.	Desviación Ojo der.	Desviación Boca
Rojo	**1**	**478 (97,8 %)**	**11 (2,2 %)**	**5,6**	**4,5**	**6,2**
Verde	**1**	449 (91,8 %)	40 (8,2 %)	6,0	4,7	6,7
Azul	**1**	445 (91,0 %)	44 (9,0 %)	6,8	5,4	7,1
Intensidad	**1**	447 (91,4 %)	42 (8,6 %)	6,1	4,6	6,6

Tabla 5.15: *Resultados del seguimiento de caras mediante proyecciones, IntProyN, sobre la secuencia "ggm4.avi" utilizando distintos canales de color. El valor de "Intensidad" se calcula con:* $0,3R + 0,59G + 0,11B$.

A la vista de los valores de la tabla 5.15, la superioridad del canal rojo parece más que manifiesta. Las diferencias son grandes en los ratios de seguimiento, pero no lo son tanto en las medidas de precisión. Se pueden obtener conclusiones parecidas con otras secuencias, y también contrastando los restantes algoritmos de seguimiento.

Aunque en todas las pruebas documentadas hasta ahora aparece como máximo una cara por imagen, se han llevado a cabo algunos otros ensayos con hasta 8 rostros por imagen. En concreto se ha usado una plantilla de imágenes dibujadas sobre un papel, como puede verse en la figura 5.33. Los resultados demuestran que el seguimiento de una cara no afecta al de las vecinas. En IntProyN, cada nueva instancia que es seguida aumenta unos 3 ms el tiempo de ejecución del proceso.

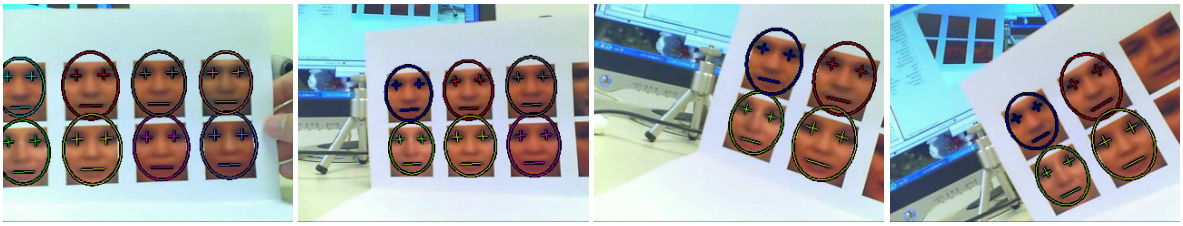


Figura 5.33: *Ejemplo de seguimiento de caras con integrales proyectivas, en una secuencia con múltiples instancias de caras. El vídeo utilizado es el archivo "8caras.avi". Cada instancia seguida se representa con un color diferente.*

Otra propiedad interesante del algoritmo desarrollado es la capacidad de mantener el seguimiento en situaciones de **oclusión o desaparición parcial** de la cara. En la figura 5.34 se muestra un ejemplo extraído de DVD, y documentado en [57]. Casi la mitad del rostro se sale por la parte inferior de las imágenes. A pesar de ello, el proceso es capaz de relocalizar los componentes faciales gracias a que los ojos siguen estando visibles. Otros ejemplos de oclusión incluyen el gesto de quitar las gafas, como se vio con la secuencia "ggm4.avi", o pasar las manos por delante de la cara, como ocurre en la base NRC-ITT.

Por último, y sin ánimo de ser exhaustivos, en la figura 5.35 mostramos otro ejemplo, también presentado en [57], que trata de evaluar la **genericidad** del método basado en proyecciones y su adaptación al seguimiento de otro tipo de objetos. El proceso es aplicado sobre un rostro *humanoide* generado por ordenador, cuyas posiciones iniciales son indicadas manualmente. El seguidor demuestra una buena capacidad de seguimiento en esta secuencia, aunque con una ligera imprecisión en la posición de la boca. Como es evidente, en este caso no se suma el modelo genérico de cara a las señales usadas en el proceso.

5.5. Conclusiones y valoración finales

Haciendo una valoración global de los experimentos, podemos concluir que el seguimiento de caras basado en proyecciones consigue alcanzar excelentes medidas de rendimiento, para **diferentes criterios** y en una amplia variedad de **entornos de trabajo**. En un escenario típico, de complejidad media/baja, donde el usuario maneja una cámara de videoconferen-

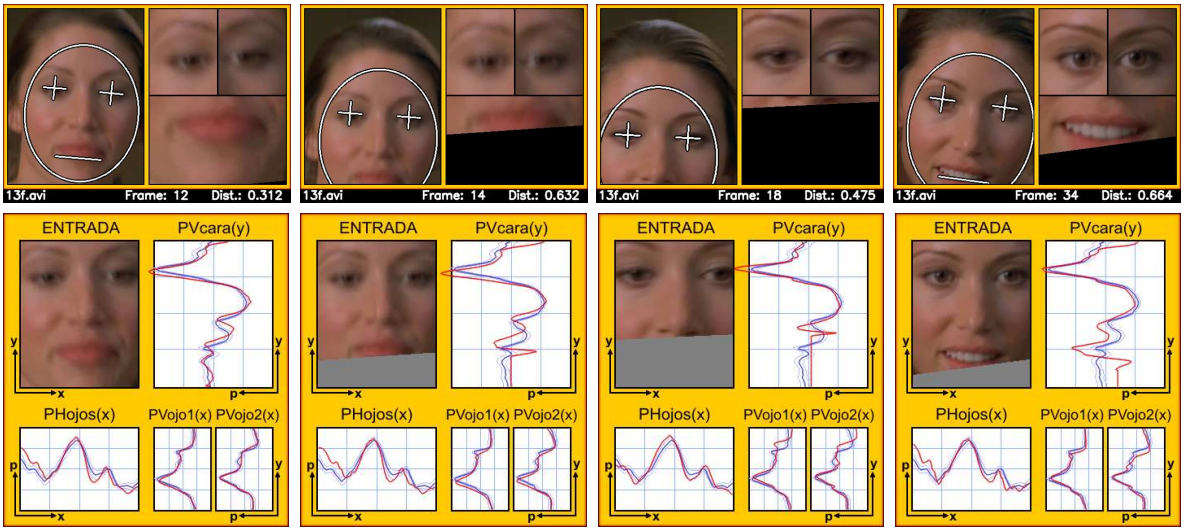


Figura 5.34: *Ejemplo de seguimiento de caras con integrales proyectivas, en una secuencia con desaparición parcial de la cara. El vídeo es un fragmento de la película en DVD "Trece fantasmas". En la parte superior, extractos de la secuencia con las posiciones seguidas por el algoritmo. En la parte inferior, la entrada del paso de relocalización y las proyecciones calculadas en el proceso.*

Figura 5.35: *Ejemplo de seguimiento de caras humanoides con integrales proyectivas. El vídeo es un fragmento de la película en DVD "La Guerra de las Galaxias: Episodio 1". Se muestran extractos de la secuencia, con las posiciones seguidas por el algoritmo.*

cia, se logra una **precisión y estabilidad** en la localización de los componentes muy superior a las restantes técnicas analizadas. Las proyecciones pueden ser usadas por sí solas, siendo el coste computacional comparable al de las técnicas más eficientes.

Si consideramos condiciones de uso más complejas, el método propuesto demuestra una **gran robustez** frente a expresiones faciales, baja resolución y oclusión parcial. La adaptación a **movimientos rápidos** del usuario se consigue combinando las integrales proyectivas con información de color. La invarianza frente a los cambios de iluminación resulta más reducida, aunque manteniendo unos niveles aceptables de seguimiento. Debemos recordar que el método propuesto no incluye ningún mecanismo específico para compensar el efecto de las sombras, por lo que los casos extremos pueden provocar la inestabilidad del proceso.

Una de las propiedades esenciales del mecanismo de seguimiento diseñado, en relación con los otros métodos analizados, es que la **relocalización** de la cara se realiza de forma **holística** y no independiente para cada elemento facial. Esta es, posiblemente, una de las claves de sus buenos resultados. Aunque tal esquema se puede aplicar también para métodos basados en patrones 2D (por ejemplo, el método de Lucas y Kanade aplicado a todo el rostro), el cambio de buscar dos ojos a relocalizar toda la cara hace que se multiplique la complejidad del problema: pasamos de 2 problemas con 2 grados de libertad (traslación de ambos ojos), a 1 problema con 4 grados de libertad (posición, escala e inclinación global del rostro). Las proyecciones permiten una relocalización global basada en 2 problemas de 2 grados de libertad (alinear las señales PV_{cara} y PH_{ojos}), más otro grado de libertad para la inclinación. Por otro lado, la manera de aprovechar la **simetría facial** ha demostrado ser capaz de manejar condiciones no triviales –como iluminación lateral de la cara o gestos asimétricos–, donde un método de simetría 2D presentaría mayores dificultades.

La mayor debilidad de la técnica desarrollada es la forma de obtener los **modelos de proyección** en los que se basa el proceso, MV_{cara} y MH_{ojos}. En el método básico, éstos se calculan exclusivamente a partir de la primera imagen de la secuencia. Una localización inicial deficiente o un gesto poco significativo pueden conducir a un mal seguimiento posterior. Para reducir este inconveniente, hemos propuesto que los modelos de proyección se calculen como una media ponderada entre la instancia inicial y los modelos genéricos de MV_{cara} y MH_{ojos}. Obviamente, se trata de una solución sencilla que podría ser mejorada. En particular, sería interesante estudiar un mecanismo de actualización continua del modelo, según las variaciones observadas en la propia secuencia. Otra alternativa pasaría por aprender los modos de variación de las proyecciones mediante un entrenamiento previo. No obstante, esto limitaría la gran flexibilidad del método propuesto.

5.6. Resumen

Existen infinidad de variantes del problema de seguimiento de caras humanas, desde las que se resuelven con una simple búsqueda de la posición central del rostro, hasta las que requieren el ajuste de una malla 3D deformable. Todas ellas tienen sus ventajas, sus inconvenientes y sus limitaciones. La aplicación final será la que determine la versión de interés en cada situación particular.

En este capítulo hemos abordado el caso del seguimiento facial con localización de los puntos medios de los ojos y la boca. El método propuesto hace uso de las integrales proyectivas y presenta las siguientes características:

- De forma genérica, el seguimiento de objetos es un proceso iterativo que consta de detección inicial, predicción y relocalización. Hemos planteado varias alternativas para **predecir la nueva posición** de la cara en la siguiente imagen de una secuencia. Pero los experimentos han demostrado que las trayectorias típicas del rostro no se pueden

describir adecuadamente mediante modelos de movimiento lineal. La alternativa más conservadora es la predicción nula, consistente en usar las posiciones del instante previo. Frente a ella, la **utilización del color de piel** puede aportar información relevante para una predicción más fiable y adaptable a movimientos rápidos.

- El funcionamiento del algoritmo de relocalización presenta muchas **similitudes** con el de **localización** de componentes faciales usando proyecciones, que se derivan de la estrecha relación entre ambos problemas. La principal diferencia es que los modelos de proyección se obtienen ahora de la propia secuencia.

- En concreto, el proceso se compone de **tres grandes pasos**. En primer lugar, se reajusta la posición de la cara en el eje Y a través del alineamiento de la proyección vertical de la cara, PV_{cara}, con el modelo correspondiente. En segundo lugar, se estima la posición en X haciendo uso de la proyección horizontal de los ojos, PH_{ojos}. Finalmente, se alinean las proyecciones verticales asociadas a los ojos para calcular la inclinación facial.

- Una de las claves del método propuesto es el tratamiento **separado** de los **grados de libertad** del problema, que son resueltos en distintas etapas: 2 grados para el ajuste vertical, 2 para el horizontal, y 1 para la inclinación. El elemento que hace posible, y efectiva, esta separación es la utilización de integrales proyectivas.

- Los resultados de los experimentos confirman ampliamente el excelente comportamiento de la técnica propuesta, tanto en la **localización precisa** de secuencias típicas, como en la **robustez** frente condiciones extremas de expresión facial, resolución de entrada, movimientos rápidos e inclinación. Además, el coste computacional permite claramente un funcionamiento en tiempo real en un ordenador medio.

- Las mayores dificultades del seguimiento con proyecciones se encuentran en los **giros fuera del plano** de la imagen y las condiciones extremas de **iluminación**. Creemos que ambos se pueden solventar, en gran medida, con la utilización de métodos más avanzados de modelado de las proyecciones. No obstante, esto puede conducir también a un mayor coste en el entrenamiento de los mismos. En cualquier caso, el desarrollo de esos modelos se postula como una prometedora línea de investigación futura.

CAPÍTULO 6

"Unfinished portrait", Tullio Pericoli, 1985

Reconocimiento de Personas

"Es un hecho de admiración común entre las personas cómo,
entre tantos millones de caras, no existen dos iguales."
SIR THOMAS BROWN, *Religio Medici*, 1643

En cierto sentido, podemos decir que los problemas abordados hasta ahora tienen un carácter básicamente *instrumental*. La detección de caras, la localización de componentes faciales y el seguimiento en vídeo no son *metas* en sí mismos, sino que sirven como base para la resolución de otros tipos de problemas con caras humanas. En el presente capítulo vamos tratar un problema de innegable aplicación inmediata: el reconocimiento facial de personas. Realmente, como veremos, no existe un único problema de reconocimiento, sino toda una familia asociada a la identificación y autentificación biométrica mediante la cara.

Uno de los grandes acercamientos clásicos a los problemas de reconocimiento visual –no sólo de personas sino de cualquier otro tipo de objetos– consiste en la reducción de las imágenes de entrada a subespacios lineales de pequeña dimensionalidad; sobre ellos se definen y aplican medidas de distancia, en muchos casos asociadas a la estimación de las funciones de densidad de probabilidad para cada clase. La clasificación se basa, normalmente, en criterios de menor distancia o máxima probabilidad. Encontrar los subespacios más adecuados y las mejores formas de modelar las funciones de probabilidad condicionadas son las cuestiones básicas dentro de este enfoque.

En este capítulo analizamos la viabilidad de utilizar integrales proyectivas en los diversos problemas del reconocimiento facial de personas. La propuesta puede ser enmarcada dentro de los métodos basados en reducción a subespacios lineales, en cuanto que el resultado final de la proyección será un vector de tamaño reducido, donde cada valor de salida es una combinación lineal de la entrada. Nuestro objetivo es cuantificar hasta qué punto las proyecciones retienen la información específica que caracteriza a un individuo, independientemente

de los factores externos (iluminación, expresión facial, pose, etc.), y en comparación con otros métodos de subespacios más elaborados.

El resto de este capítulo está estructurado de la siguiente forma. Comenzamos haciendo una introducción al contexto del reconocimiento biométrico de personas, en la sección 6.1, analizando los subproblemas que aparecen, y los protocolos y criterios de evaluación de los diversos escenarios. La sección 6.2 repasa brevemente algunos de los trabajos más interesantes en este ámbito de investigación. A continuación, en la sección 6.3 abordamos la utilización de integrales proyectivas en el reconocimiento facial. El punto clave se encuentra en seleccionar las proyecciones que mejor retienen la información específica de cada individuo; la clasificación se basa en medidas de distancia sobre el espacio de las proyecciones. Los experimentos se detallan en la sección 6.4, comparando el método propuesto con otros tipos de reconocedores. Para acabar, se hace un rápido resumen del capítulo en la sección 6.5.

6.1. El contexto del reconocimiento de personas

Sin duda alguna, los problemas de identificación biométrica constituyen una de las grandes áreas de expansión de la percepción artificial en los últimos años. En cierto sentido, este hecho no es más que el reflejo de una mayor preocupación, tanto en el ámbito público como en el privado, por la cuestiones relacionadas con la seguridad [108, 212].

Hasta la fecha, son muchos los tipos de muestras biométricas que han sido estudiadas y utilizadas [3]: las huellas dactilares, el iris, la firma, la voz, la palma de la mano, las orejas, las venas de la retina o de la mano, el ADN, la forma de caminar, la dinámica en el uso del teclado, etc.; además, lógicamente, del rostro humano. Cada tipo tiene sus propias ventajas e inconvenientes, de los cuales se deduce su ámbito de aplicación específico. Por ejemplo, el reconocimiento mediante ADN es ideal en aplicaciones de análisis forense, pero es completamente inviable en un sistema de vigilancia de aeropuertos; por su parte, las huellas dactilares o las caras pueden usarse en el control de accesos a un edificio, mientras que en una aplicación bancaria es preferible la firma, puesto que se requiere una constancia por escrito.

En general, la bondad de los sistemas de reconocimiento biométrico puede cuantificarse en función de una serie de factores y criterios tales como [114]:

- la mayor o menor **intrusividad**, desde el punto de vista del usuario;
- la **precisión** esperada para ese tipo de medidas;
- el **coste** de implantación, a todos los niveles; y
- el **esfuerzo**, esto es, la mayor o menor facilidad de obtención de las muestras.

En la figura 6.1 se muestra el resultado de un análisis comparativo de estos parámetros para algunos de los tipos de biométricas con mayor presencia, no sólo en los ámbitos puramente de investigación sino también en el mundo comercial.

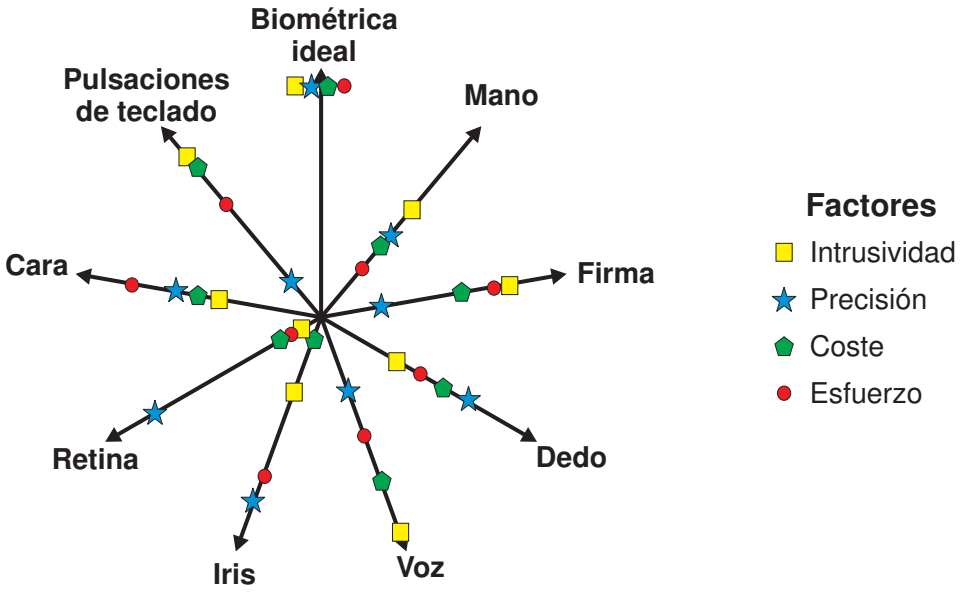


Figura 6.1: *Análisis de la bondad de diversos tipos de sistemas biométricos. Para cada tipo de biométrica se ponderan 4 factores en una escala relativa. El centro del diagrama significa un valor malo para ese factor. El análisis es de la compañía Zephyr Biometrics, (c) International Biometric Group [11].*

Ninguna de las alternativas existentes se aproxima en todos sus parámetros al sistema biométrico ideal. No obstante, podemos observar que el reconocimiento facial presenta un buen compromiso entre los diferentes criterios. Se pueden encontrar conclusiones similares en [80], donde se estudia la compatibilidad de diferentes biométricas con los sistemas denominados de "documentación de viajeros con lectura mecanizada"[1]; combinando diferentes factores de interés, las caras se sitúan en primer lugar, justo por delante de las huellas dactilares y el análisis ocular. Aunque si nos fijamos en datos objetivos como las cuotas de mercado, según el International Biometric Group [11], en 2006 la tecnología de caras se encontraba en segundo lugar, justo por detrás de las basadas en huellas y a gran distancia de las mismas (en concreto, un 19 % de cuota frente a un 44 % para las huellas del dedo).

Ciñéndonos al contexto académico, también es claro el lugar destacado que ocupa la investigación en reconocimiento automático de caras humanas, tanto en el número de publicaciones, como de grupos de trabajo y congresos específicos sobre el tema. En la actualidad, este campo es uno de los más *maduros* en el ámbito de la visión artificial, y especialmente en relación con los otros problemas sobre caras humanas. Todo esto se deja ver en la mayor estandarización de los criterios y protocolos de evaluación, como podremos comprobar en los siguientes apartados.

6.1.1. Problemas asociados al reconocimiento de caras

Existen diferentes escenarios posibles para los sistemas de autentificación biométrica mediante la cara. No es lo mismo, por ejemplo, una aplicación de control de accesos a un edi-

[1] En inglés, *machine readable travel documents* (MRTD).

ficio –que ciertas personas utilizan para identificarse y entrar al recinto–, que un sistema de videovigilancia en un lugar público –donde se trata de identificar criminales que pasan aleatoriamente frente a la cámara–. Existe una diferencia fundamental entre ambos: en el primero, el individuo *quiere ser* reconocido; en el segundo, *quiere no ser* reconocido.

Tampoco es comparable una aplicación que maneje varias docenas de personas con otra que tenga que tratar cientos de miles. A medida que aumenta el tamaño del problema, se deben abordar cuestiones operativas de naturaleza muy diversa. Y, dentro de un mismo tamaño, pueden presentarse casos donde el usuario tenga la posibilidad de *acreditarse* mediante algún mecanismo alternativo, como por ejemplo una tarjeta identificativa.

Para modelar todo este amplio espectro de posibles aplicaciones del reconocimiento de caras, se definen tres problemas típicos asociados a escenarios estándar [108, 114]: identificación en conjunto abierto, identificación en conjunto cerrado, y verificación. Vamos a describir las características de cada uno de ellos, empezando por los más sencillos. Pero antes, introducimos un poco de nomenclatura sobre el tema.

Definiciones y terminología en el contexto de las biométricas

Existen algunos términos de uso común en el contexto del reconocimiento biométrico, que conviene aclarar antes de proseguir. Nos basamos aquí en la notación presentada en el capítulo 14 de [108], utilizada en la mayoría de los trabajos existentes.

Una *muestra biométrica* (o, simplemente, una *muestra*) es una medición obtenida de cierta persona y que permite distinguirla de otras. Puede ser una imagen de la cara o de su huella dactilar, una grabación de su voz, una secuencia de vídeo, etc. Típicamente, las muestras pueden entenderse como vectores de números de tamaño dado, $x = (x[1], x[2], \ldots, x[k])$.

Al conjunto de muestras biométricas de personas para las que se conoce su identidad se le denomina la *galería*, y se denota por $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$. En principio, una galería puede contener varias muestras por individuo[2]. Representamos la identidad asociada a cada muestra de $\mathcal{G}$ por la función *id*, siendo $id(g_i)$ un identificador único asociado a la persona de la que se ha obtenido la muestra g_i.

En otros contextos, la galería es lo que se denomina el "conjunto de entrenamiento". Los "conjuntos de test" se llaman aquí *conjuntos de pruebas*. En concreto, una *prueba* es una muestra biométrica que debe clasificar el sistema de reconocimiento. Se distinguen, en general, dos posibles conjuntos de pruebas: $\mathcal{P}_\mathcal{G}$ y $\mathcal{P}_\mathcal{N}$. El conjunto $\mathcal{P}_\mathcal{G}$ está formado por muestras de personas que están en $\mathcal{G}$; mientras que $\mathcal{P}_\mathcal{N}$ contiene muestras de individuos que no están en $\mathcal{G}$, es decir, son personas "desconocidas" para el sistema.

Otro término interesante es el de *puntuación* –en inglés, *score*–. Dada una prueba, p_j (ya sea de $\mathcal{P}_\mathcal{G}$ o de $\mathcal{P}_\mathcal{N}$), y una muestra de la galería, g_i, la puntuación es una medida de similitud entre ambas muestras, y se denota por s_{ij}. Cuanto mayor sea la puntuación, se supone que

[2]Debemos notar que en este aspecto no seguimos la formalización propuesta en [108], que considera que en la galería sólo puede existir una muestra por individuo. No obstante, esto no afecta sustancialmente a los conceptos y criterios sucesivos.

existe más probabilidad de que ambas muestras pertenezcan al mismo individuo. En términos numéricos, el concepto de puntuación sería opuesto al de distancia; dada una distancia, se puede obtener una puntuación tomando el inverso, y viceversa.

Es interesante observar que en muchas ocasiones el diseño de una técnica de reconocimiento se reduce a definir un método de obtención de las puntuaciones, dando por hecho que a partir de las mismas los diferentes problemas se resuelven de forma más o menos directa. A continuación vamos a ver exactamente cómo.

Descripción de los problemas de reconocimiento

Los tres problemas estándar que se encuentran habitualmente en el ámbito del reconocimiento biométrico de personas son los siguientes:

- **Identificación en conjunto cerrado**

  Este problema es el que se ha entendido tradicionalmente al hablar de "reconocimiento de personas". En esencia, dada una galería $\mathcal{G}$ y una prueba p_j de una persona del conjunto $\mathcal{P}_\mathcal{G}$, la identificación en conjunto cerrado consiste en encontrar cuál es su identidad más probable. En la figura 6.2 se representa gráficamente este escenario.

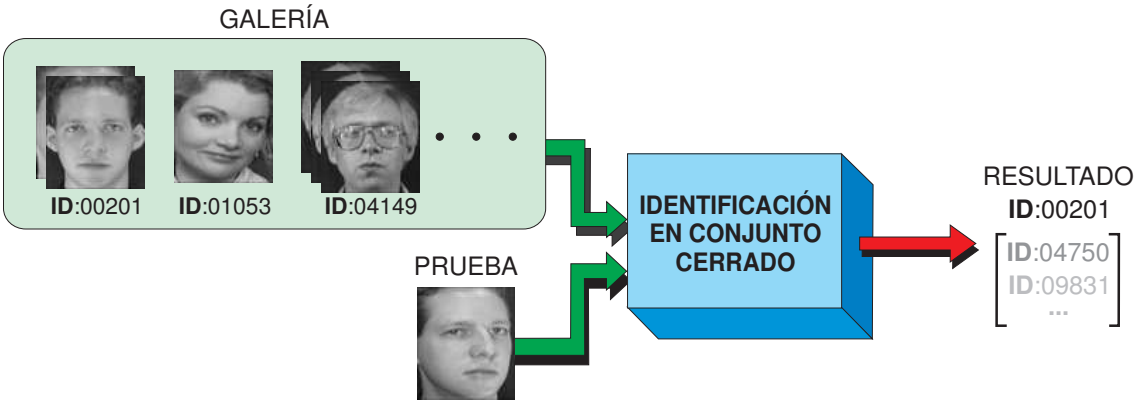


Figura 6.2: *Esquema de la identificación en conjunto cerrado. La entrada es la galería de muestras y la prueba biométrica a ser reconocida. El resultado es una identidad de la galería. Opcionalmente, se pueden devolver también las n identidades más probables. Las imágenes de ejemplo están tomadas de la base de caras ORL [159].*

  Posiblemente, esta es la categoría que más se aproxima al problema genérico de clasificación de patrones. La galería es el conjunto de entrenamiento y las pruebas biométricas son los patrones nuevos a los que se debe asignar una clase. Por lo tanto, es posible aplicar todos los avances en el extenso campo del reconocimiento de patrones. No obstante, una vez más se impone la especialización debida a las particularidades del dominio de aplicación. Por ejemplo, ¿qué beneficio aportan las máquinas de vectores de soporte (SVM) cuando existen varios miles de clases y sólo un ejemplo por clase?

  Existe otra forma de interpretar el problema, desde el punto de vista de las puntuaciones. Suponiendo definida la función de similitud, s_{ij}, la cuestión se reduce a encon-

trar el elemento g^* de $\mathcal{G}$ con mayor puntuación respecto de p_j, es decir, maximizar s_{*j}. El resultado del sistema sería simplemente $id(g^*)$.

Aunque, en el fondo, ambos planteamientos son análogos, el segundo enfoque está más centrado en la representación de las muestras biométricas y la definición de medidas de distancia/similitud, mientras que el primero parece más orientado al estudio de los mecanismos de clasificación.

- **Verificación**

  La característica específica del problema de verificación –también llamado a veces *autentificación*– reside en que el usuario dispone de un mecanismo alternativo para indicar su identidad al sistema, como podría ser una tarjeta o un número identificativo. De esta manera, la entrada al proceso es la identidad aducida por la persona, e, y una muestra biométrica, p_j, utilizada como una prueba –en sentido literal– de esa identidad. Se puede ver un ejemplo de esta situación en la figura 6.3.

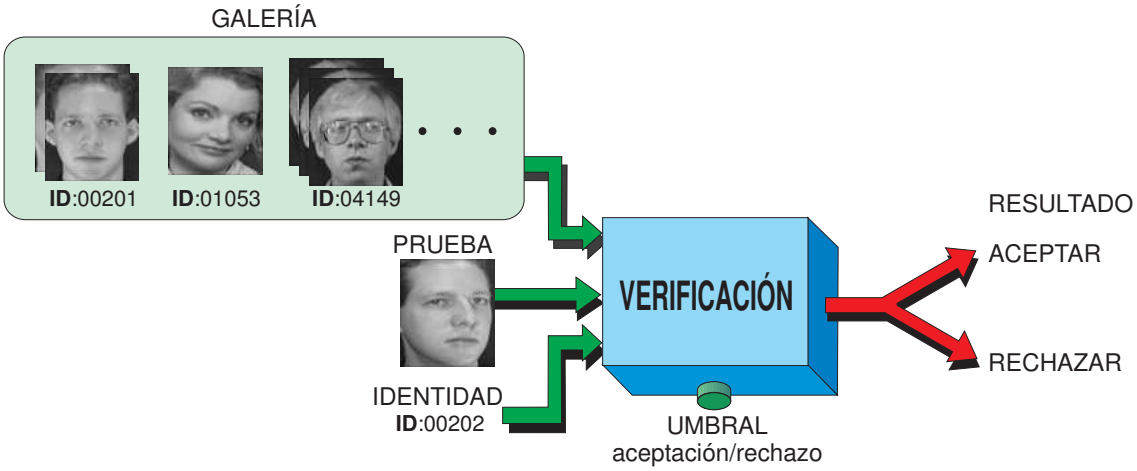


Figura 6.3: *Esquema de la verificación biométrica. La entrada es la galería de muestras, la identidad aducida por la persona y una prueba biométrica de la misma. El resultado es la decisión de aceptar o rechazar la prueba. Existe un parámetro que controla el nivel de tolerancia en las decisiones aceptación/rechazo. Imágenes de ejemplo tomadas de la base ORL [159].*

Llamamos *impostor* a un individuo que reclama una identidad que no le corresponde. Si un impostor recibe una respuesta positiva del sistema, decimos que ha ocurrido una *falsa aceptación*. Por su parte, si un no impostor es rechazado, lo denominamos *falso rechazo*. Todas las demás situaciones son *verificaciones correctas*. Obviamente, aparecerán cuestiones relativas al compromiso entre falsos rechazos y falsas aceptaciones.

En términos del reconocimiento de patrones, la verificación se puede plantear como un problema de clasificación binaria clase/no clase. A partir de la galería, se deberían entrenar tantos clasificadores binarios como número de individuos existan. De esta forma, dado un patrón nuevo y una identidad supuesta, se selecciona y aplica el clasificador correspondiente, obteniendo una respuesta booleana: pertenece a la clase o no pertenece.

También es posible observar el problema desde la perspectiva de las puntuaciones. Dada p_j y la identidad aducida e, se trataría de encontrar la mayor puntuación $s_{i'j}$ para todo i' con $id(g_{i'}) = e$. Si la puntuación resultante está por encima de cierto umbral, se acepta la prueba y en otro caso se rechaza.

La distinción entre los impostores que pertenecen a la galería y los que no pertenecen puede ser importante. Para los primeros, es de esperar que exista alguna puntuación alta, s_{ij}, asociada a la verdadera identidad. Por lo tanto, se pueden aplicar técnicas de *normalización* de las puntuaciones. Por ejemplo, se podrían calcular todas las puntuaciones de p_j con la galería $\mathcal{G}$, y dividir por el máximo global. La situación no está tan clara cuando la persona no sólo es un impostor sino que tampoco pertenece a $\mathcal{G}$ (lo que se llama comúnmente un *verdadero impostor*).

- **Identificación en conjunto abierto**

  En cierto sentido, la identificación en conjunto abierto se puede entender como una combinación de los dos problemas anteriores. Dada una galería $\mathcal{G}$ y una prueba p_j, se trata de resolver dos cuestiones: en primer lugar, determinar si la persona es conocida o no (es decir, si está en $\mathcal{G}$ o no está); y, en caso afirmativo, seleccionar su identidad más probable. La figura 6.4 representa esquemáticamente este escenario.

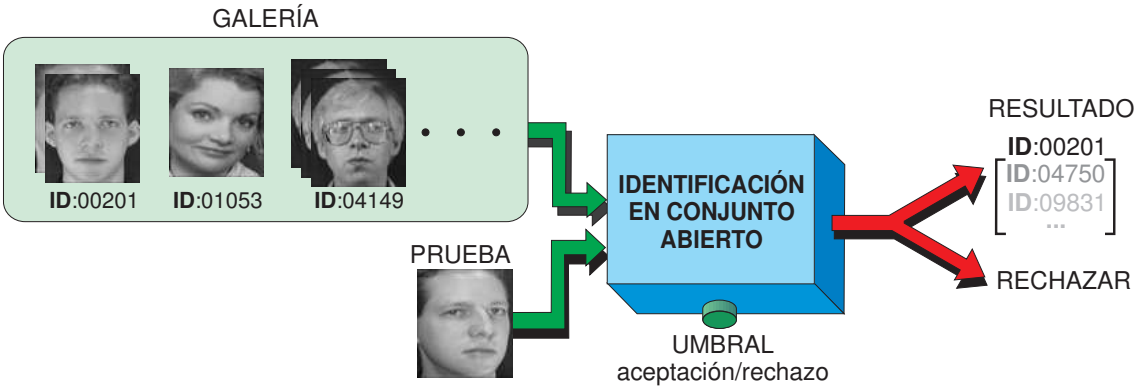


Figura 6.4: *Esquema de la identificación en conjunto abierto. La entrada es la galería de muestras y la prueba biométrica a ser reconocida. El resultado puede ser aceptar la prueba (la persona está en la galería) o rechazarla (no está en la galería). En el primer caso, se devuelve también la identidad, o identidades, más probables. Existe un parámetro que controla el nivel de tolerancia en las decisiones aceptación/rechazo. Imágenes de ejemplo tomadas de la base ORL [159].*

Los sistemas de videovigilancia y detección de criminales son ejemplos de aplicaciones dentro de esta categoría. Existe una base de datos de delincuentes –o, en general, de personas buscadas por la ley– utilizada en el reconocimiento, pero la mayoría de los individuos que pasan frente a la cámara serán desconocidos. En cualquier caso, los "usuarios" no presentan su identidad al sistema, y posiblemente no sean conscientes de que están siendo controlados.

Esta categoría representa un paso adicional de dificultad, ya que existen más posibles situaciones de error. Por un lado, una persona que pertenezca a la galería puede ser

identificada correctamente, puede serle asignada una identidad equivocada, o puede ser rechazada erróneamente. De forma parecida, para una prueba del conjunto $\mathcal{P}_{\mathcal{N}}$ sería un error atribuirle cualquier identidad de $\mathcal{G}$.

Usando clasificación de patrones, el problema se puede plantear de dos maneras. En la primera –en analogía al problema de verificación–, existirían tantos clasificadores binarios como personas en $\mathcal{G}$. La prueba, p_j, se sometería a todos ellos, declarándose rechazada si la respuesta es negativa para todos. Habría que articular algún mecanismo si más de un clasificador admite la prueba. La segunda alternativa sería una extensión de la identificación en conjunto cerrado; se trataría de añadir una clase adicional al clasificador multiclase, que representara a todos los posibles individuos desconocidos.

Basándose en las puntuaciones, también es posible trazar una analogía con la identificación en conjunto cerrado. Como en el caso anterior, se debe buscar el elemento g^* de $\mathcal{G}$ que maximice s_{*j}. Además, es necesario establecer un umbral de rechazo, τ; si s_{*j} es mayor que τ, se declara la identidad $id(g^*)$, y si está por debajo se rechaza la prueba. La selección del umbral puede ser compleja, ya que diferentes individuos pueden producir puntuaciones en rangos muy variados. En tal situación, sería necesario aplicar algún tipo de normalización[3].

No existe un orden de preferencia, o de mayor relevancia, entre los diferentes problemas de reconocimiento. Simplemente, cada uno tiene su propio ámbito de aplicación, asociado al escenario de uso del sistema biométrico dado.

Una ventaja de utilizar las puntuaciones es que los tres problemas quedan resueltos automáticamente una vez definido el mecanismo de obtención de los *scores*. De hecho, la mayoría de los protocolos de evaluación estandarizados [52, 143, 144, 13, 139], toman como entrada la matriz de puntuaciones, entre cada muestra de la galería y cada prueba de entrada. Las diferentes medidas de rendimiento se obtienen automáticamente, como vamos a ver a continuación.

6.1.2. Evaluación de los métodos de reconocimiento

La necesidad de definir métodos y protocolos estándar de evaluación de los sistemas biométricos surgió ya en los primeros años de la disciplina. Debido al gran número de propuestas existentes, resultaba necesario establecer no sólo los parámetros y medidas objeto de estudio, sino también el propio mecanismo de evaluación. El protocolo *de facto* fue establecido por el programa FERET [52, 143, 144], y utilizado después en otras evaluaciones públicas, como las del FRVT (*Face Recognition Vendor Test*) [13, 139]. El primero fue patrocinado por el ejército de los Estados Unidos, y dio lugar a tres evaluaciones en 1994, 95 y 96; mientras que el segundo corresponde a una iniciativa privada de empresas especializadas, y se llevó a cabo

[3]Pero obsérvese que, en este caso, la normalización de "dividir por el máximo" propuesta para la verificación, no sería válida, ya que siempre devolvería 1 para s_{*j}.

en 2000 y 2002. Podemos entender estas evaluaciones como "concursos" en los que varios participantes pusieron a prueba sus sistemas sobre unos mismos conjuntos de datos. Los protocolos que se definieron, los criterios de evaluación utilizados y las conclusiones obtenidas son una referencia obligada en cualquier trabajo de reconocimiento facial.

La metodología de las pruebas estaba basada en una serie de preceptos, articulados en [140], que pretenden garantizar una competición justa y objetiva. Podemos resumirlos en los cuatro siguientes [108]:

1. Las evaluaciones son diseñadas y administradas por grupos independientes a los desarrolladores de los algoritmos y las empresas que los venden o patrocinan.

2. Los datos no pueden ser accesibles por los participantes antes de la evaluación.

3. El diseño, protocolo y metodología de las pruebas de evaluación deben ser públicos.

4. Las medidas de rendimiento deben ofrecer diferencias significativas entre los distintos competidores.

El último precepto está relacionado con el denominado "problema de los tres osos[4]": las pruebas no deben ser ni muy fáciles ni muy difíciles, porque si los valores son todos altos o todos bajos, es imposible comparar las ventajas e inconvenientes de los diferentes métodos.

Obviamente, en unas pruebas no públicas, como las que presentamos en la sección 6.4, muchos de estos principios no son aplicables. Sólo podemos intentar conseguir la mayor imparcialidad en la comparación entre el método que proponemos y las otras técnicas alternativas disponibles. Las imágenes usadas en las evaluaciones del programa FERET están disponibles públicamente [52], y serán utilizadas en algunos de nuestros experimentos. También haremos uso, a efectos comparativos, de algunos resultados de estas evaluaciones.

Debemos aclarar que la mayoría de las evaluaciones públicas –entre ellas, las más relevantes, como FERET y FRVT– utilizan como entrada para el análisis las puntuaciones, o *scores*, ofrecidas por cada método de reconocimiento participante. En concreto, parten de la matriz de puntuaciones, s_{ij}, que relaciona cada muestra de la galería, g_i, con cada prueba, p_j. A partir de ellas se obtienen de forma automatizada las distintas medidas de rendimiento. Vamos a describir las más relevantes.

Medidas de rendimiento para la identificación en conjunto cerrado

Supongamos que tenemos una galería, $\mathcal{G}$, un conjunto de pruebas, $\mathcal{P}_\mathcal{G}$[5], y queremos cuantificar el rendimiento de cierto método, que se caracteriza por sus puntuaciones, s_{ij}. La medida básica consistiría en contar el número de pruebas de $\mathcal{P}_\mathcal{G}$ que son reconocidas correctamente. Dado un p_j y $g^* = \arg\max_{\forall i} s_{ij}$, el reconocimiento es correcto si $id(p_j) = id(g^*)$.

[4] En referencia al cuento infantil de autor desconocido "*Ricitos de oro y los tres ositos*".

[5] Recordemos que en la identificación en conjunto cerrado las pruebas son siempre de individuos que están en $\mathcal{G}$, es decir, $\mathcal{P}_\mathcal{N}$ sería el conjunto vacío.

Sin embargo, en algunos casos, también se podría considerar un buen resultado si la puntuación obtenida fuera la segunda mayor, o la tercera... En general, se dice que una prueba p_j tiene rango n, $rango(p_j) = n$, si la puntuación para la identidad correcta es la n-ésima mayor de entre los individuos de la galería. Así, por ejemplo, decimos que la prueba se reconoce correctamente cuando su rango es 1.

Usando el concepto de rango, se define el *ratio o porcentaje de identificación* para rango n, $P_I(n)$, como el cociente:

$$P_I(n) = \frac{C(n)}{|\mathcal{P}_\mathcal{G}|} \tag{6.1}$$

Siendo $C(n)$ el número acumulado de pruebas con rango n o menor, es decir:

$$C(n) = |\{p_j \,/\, \forall p_j \in \mathcal{P}_\mathcal{G};\ rango(p_j) \leq n\}| \tag{6.2}$$

El valor $P_I(1)$ se denomina también el "ratio de identificación correcta" o el "ratio para la puntuación máxima" (en inglés, *top match score*).

La representación de la función $P_I(n)$ para distintos valores de n es lo que se conoce como la **curva CMC** (*cumulative match characteristic*, o curva característica de correspondencia acumulada). Esta curva nos permite observar rápidamente la bondad de una técnica de reconocimiento. Será mejor cuanto mayores sean los valores que toma a partir rangos bajos. Obviamente, la curva es no decreciente y para el rango máximo, $|\mathcal{P}_\mathcal{G}|$, toma valor 1. En la figura 6.5 se muestran dos ejemplos de curvas CMC, para una ejecución del método que proponemos en el apartado 6.3 y para una técnica básica.

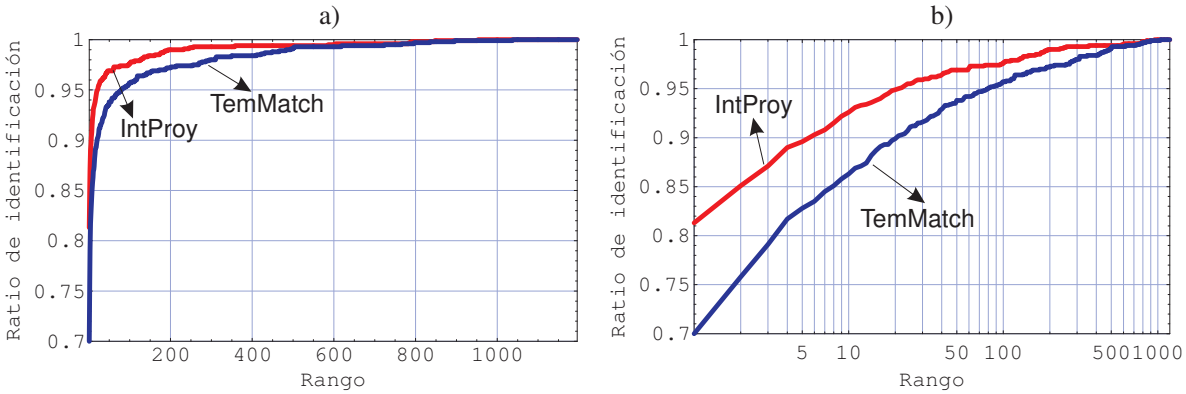


Figura 6.5: *Ejemplo de curvas CMC para el rendimiento de la identificación en conjunto cerrado. La galería contiene 1196 imágenes de la base FERET [52], con una imagen por persona. El conjunto de pruebas contiene 1195 imágenes de distintas personas de la galería. a) Curvas CMC del método basado en proyecciones (en rojo) y en* template matching *(en azul). b) Las mismas curvas utilizando una escala logarítmica para el rango.*

Obsérvese que en la figura 6.5b) se ha utilizado una escala logarítmica en el eje horizontal, con el propósito de destacar el comportamiento en los valores bajos del rango (que son normalmente los de mayor interés). También es posible expresar el rango proporcionalmente al

tamaño del conjunto de prueba, $\mathcal{P}_\mathcal{G}$. Esto tendrá sentido, por ejemplo, si queremos comparar dos experimentos con distintos tamaños de galería.

Medidas de rendimiento para la verificación

Como ya hemos adelantado, el rendimiento en el problema de verificación está relacionado con los falsos rechazos y las falsas aceptaciones. En primer lugar, definimos el *ratio de verificación* como el porcentaje de pruebas del conjunto $\mathcal{P}_\mathcal{G}$ que son aceptadas por el sistema para su identidad correcta.

En términos de las puntuaciones, la prueba p_j será aceptada correctamente si su puntuación con alguna muestra de la galería $g_i \in \mathcal{G}$, siendo $id(p_j) = id(g_i)$, está por encima de cierto umbral fijado de antemano, τ. Luego el ratio de verificación para umbral τ se puede expresar formalmente como:

$$P_V(\tau) = \frac{|\{p_j \;/\; \forall p_j \in \mathcal{P}_\mathcal{G},\; \exists g_i \in \mathcal{G};\; id(p_j) = id(g_i) \wedge s_{ij} \geq \tau\}|}{|\mathcal{P}_\mathcal{G}|} \tag{6.3}$$

El ratio de falsos rechazos sería simplemente $1 - P_V(\tau)$.

La otra medida estándar es el *ratio de falsas aceptaciones*. Existen varias formas de obtener este valor. Si sólo disponemos del conjunto de pruebas $\mathcal{P}_\mathcal{G}$, sería el número de puntuaciones s_{ij} que superan el umbral τ cuando las identidades de p_j y g_i no coinciden; es decir, un individuo de la galería es aceptado con otra identidad distinta a la suya. El ratio se puede obtener con la fórmula[6]:

$$P_{FA}(\tau) = \frac{|\{s_{ij} \;/\; \forall p_j \in \mathcal{P}_\mathcal{G},\; \forall g_i \in \mathcal{G};\; id(p_j) \neq id(g_i) \wedge s_{ij} \geq \tau\}|}{(|\mathcal{P}_\mathcal{G}| - 1)|\mathcal{G}|} \tag{6.4}$$

Este método suele criticarse porque supone que los impostores son personas de la misma galería. Se argumenta que deberían utilizarse verdaderos impostores, esto es, muestras del grupo $\mathcal{P}_\mathcal{N}$. En ese caso, el porcentaje de falsas aceptaciones sería:

$$P_{FA}(\tau) = \frac{|\{s_{ij} \;/\; \forall p_j \in \mathcal{P}_\mathcal{N},\; \forall g_i \in \mathcal{G};\; s_{ij} \geq \tau\}|}{|\mathcal{P}_\mathcal{N}| \cdot |\mathcal{G}|} \tag{6.5}$$

En definitiva, cada ajuste de τ da lugar a un valor de P_V y otro de P_{FA}. La variación de $P_V(\tau)$ frente a $P_{VA}(\tau)$ para distintos valores de τ es lo que se conoce como la **curva ROC** (*receiver operating characteristic*). Desde el punto de vista del cliente del sistema biométrico, es una especificación de los diferentes modos de funcionamiento del sistema. La figura 6.6 contiene algunos ejemplos de estas curvas.

En las gráficas de la figura 6.6 se han representado las curvas de dos métodos sobre unos mismos datos. Como en la figura 6.5, se usa una escala logarítmica para precisar el comportamiento en los valores pequeños del eje horizontal.

[6] En esta fórmula sí que se requiere una sola muestra en $\mathcal{G}$ por persona. Equivalentemente, podemos suponer que los s_{ij} son los máximos de todos los $s_{i'j}$ con $id(g_i) = id(g_{i'})$.

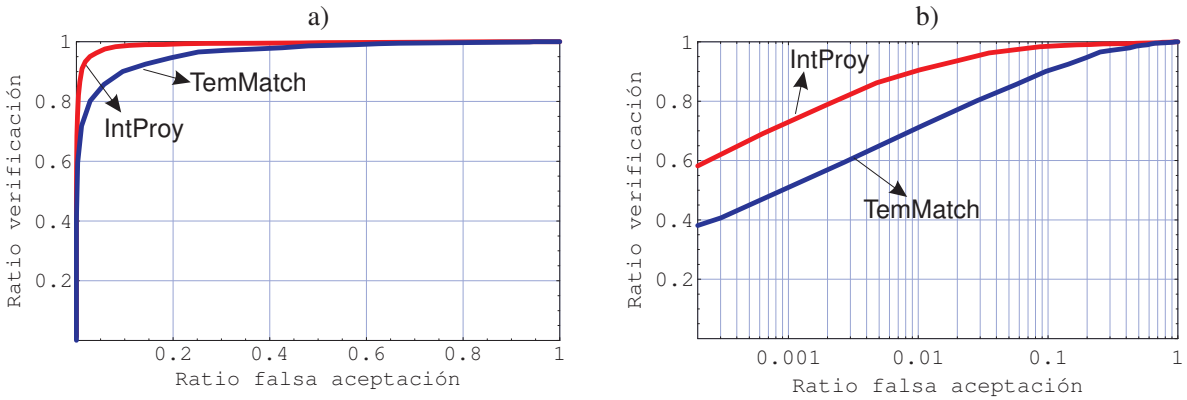


Figura 6.6: *Ejemplo de curvas ROC para el rendimiento de la verificación. La galería y las pruebas son las mismas que las de la figura 6.5. a) Curvas ROC del método basado en proyecciones (en rojo) y en* template matching *(en azul). b) Las mismas curvas utilizando una escala logarítmica para el rango.*

Medidas de rendimiento para la identificación en conjunto abierto

Para una evaluación adecuada de la identificación en conjunto abierto resulta conveniente utilizar los dos conjuntos de pruebas, $\mathcal{P}_\mathcal{G}$ y $\mathcal{P}_\mathcal{N}$. Si sólo disponemos de $\mathcal{P}_\mathcal{G}$, la estrategia más conservadora consistiría en no rechazar nunca las pruebas, de manera que las métricas obtenidas pueden ser poco realistas.

En este problema, una prueba p_j se rechaza cuando su máxima puntuación, s_{*j}, no supera el umbral τ. Por lo tanto, el *ratio de falsas aceptaciones* será la proporción de verdaderos impostores cuya puntuación máxima está por encima de τ:

$$P_{FA}(\tau) = \frac{|\{p_j \,/\, \forall p_j \in \mathcal{P}_\mathcal{N};\ \text{máx}_{\forall i}\, s_{ij} \geq \tau\}|}{|\mathcal{P}_\mathcal{N}|} \tag{6.6}$$

Cuando una prueba de $\mathcal{P}_\mathcal{G}$ no es rechazada y es reconocida correctamente, decimos que ha ocurrido una detección e identificación correcta. El *ratio de detección e identificación* es la proporción de muestras de $\mathcal{P}_\mathcal{G}$ que cumplen ambos criterios. De esta manera, no basta con que la puntuación obtenida sea la máxima para la identidad correcta (esto es, lo que denominamos $rango(p_j) = 1$), sino que además esa puntuación debe ser mayor o igual que el umbral establecido, τ. Como en el caso de conjunto cerrado, podemos considerar identificaciones hasta cierto rango n.

En definitiva, el cálculo del *ratio de detección e identificación* para rango n y umbral τ es parecido a la fórmula 6.1, pero añadiendo la condición asociada al umbral. Es decir:

$$P_{DI}(\tau, n) = \frac{|\{p_j \,/\, \forall p_j \in \mathcal{P}_\mathcal{G};\ rango(p_j) \leq n \wedge s_{*j} \geq \tau\}|}{|\mathcal{P}_\mathcal{G}|} \tag{6.7}$$

Existen varias formas comunes de mostrar los resultados de la identificación en conjunto abierto. Puesto que la función P_{DI} tiene dos dimensiones y P_{FA} una, se necesitaría idealmente una representación tridimensional. En su lugar, se puede fijar un parámetro y observar cómo varían los otros. Por ejemplo, fijando el rango a $n = 1$ podemos obtener una gráfica del ratio

de detección e identificación frente al número de falsas alarmas en una curva ROC. Se puede ver un ejemplo en la figura 6.7. Llegado a cierto punto del eje horizontal, el ratio de detección e identificación prácticamente se mantiene constante. Esto es debido a que, aunque se reduzca mucho el umbral, τ, la puntuación de la clase correcta no es la máxima. En consecuencia, el máximo valor alcanzable en esta curva es, precisamente, $P_I(1)$.

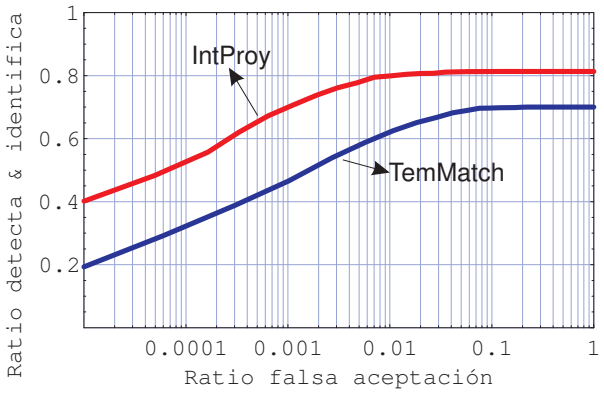


Figura 6.7: *Ejemplo de curvas ROC para identificación en conjunto abierto. La galería y las pruebas son las mismas que las de la figura 6.5. Como no hay un conjunto de verdaderos impostores, se ha usado la fórmula 6.4 para las falsas alarmas. Se muestra el ratio de falsas alarmas frente al de detección e identificación correcta, para el método basado en proyecciones (rojo) y en* template matching *(azul).*

También podemos obtener una curva CMC fijando el número de falsas aceptaciones a un valor dado. Por ejemplo, las curvas CMC de identificación en conjunto cerrado (como las de la figura 6.5) se pueden interpretar como casos donde el ratio de falsas aceptaciones es de 1.

6.1.3. Los grandes desafíos del reconocimiento

A pesar de la relativa madurez del reconocimiento automático de caras, siguen existiendo grandes cuestiones abiertas, factores adversos que hacen que los ratios de identificación bajen hasta órdenes del 60 %, 50 % o aún menores, inaceptables en muchas aplicaciones. Por ejemplo, aunque hay implantados desde hace años algunos sistemas para la identificación de criminales –ver por ejemplo los casos de Newham, Londres [123], o el aeropuerto de Logan, Montana [191]–, en la práctica, todavía no se ha informado de que hayan sido capaces de identificar a algún sospechoso.

Realmente, el caso de la identificación de personas *perseguidas por la ley* se encuentra entre las aplicaciones más complejas del reconocimiento facial. Por un lado, como ya hemos mencionado, porque las personas de la galería *no quieren* ser identificadas. Por otro lado, porque las imágenes captadas típicamente por los sistemas CCTV[7] ofrecen una calidad muy limitada. Pero, paradójicamente, no existen muchas referencias donde se valore el rendimiento en situaciones donde el usuario intenta ocultar o disfrazar su rostro; posiblemente, porque tal rendimiento es extremadamente reducido, si no nulo. No obstante, sigue habiendo un gran

[7]Sistemas de televisión en circuito cerrado (*closed circuit TV*).

número de aplicaciones potenciales del reconocimiento facial. Prueba de ello es la gran cantidad de empresas dedicadas a la comercialización de este tipo de sistemas, [31].

Vamos a señalar los principales obstáculos que deben ser abordados en la construcción de sistemas biométricos basados en caras. Buena parte de esta discusión está basada en las conclusiones resultantes de las evaluaciones de los programas FERET y FRVT.

- **Los efectos del tiempo**

  En todos los sistemas de reconocimiento de caras ocurre un descenso notable del rendimiento a medida que aumenta el tiempo transcurrido entre la obtención de la muestra de la galería y la captura de la cara de prueba. Comprender y modelar los efectos del tiempo sobre los rostros humanos es una de las grandes cuestiones pendientes de la disciplina. La base de datos FERET contiene varios conjuntos de imágenes de las mismas personas tomadas en diferentes fechas –los denominados *duplicados*–, con hasta 3 años de diferencia; la figura 6.8 muestra un ejemplo. En la evaluación del FRVT 2002 se estimó que el efecto del tiempo supone un descenso aproximado en el rendimiento del 5 % por cada año transcurrido [108]. Este problema supone que la información almacenada en los sistemas biométricos debe ser actualizada con cierta periodicidad.

Figura 6.8: *Imágenes de la base de caras FERET. Las imágenes están etiquetadas por categorías. a,b) Los conjuntos "fa" y "fb" están tomados el mismo día, cambiando la expresión facial. c) El conjunto "fc" es también del mismo día, pero con diferente iluminación. d) Los duplicados "dup1" son normalmente dentro de un año de diferencia. e) En "dup2" la diferencia es de más de un año.*

- **La escalabilidad de los sistemas**

  Otra de las grandes cuestiones de interés es la influencia del tamaño de las galerías en el rendimiento de los sistemas. En la actualidad, manejando una galería con unos cuantos cientos de usuarios y con condiciones más o menos controladas, se pueden esperar ratios de identificación muy próximos al 100 %. Pero, a medida que aumenta el tamaño, el rendimiento de los sistemas se degrada de forma progresiva.

  El interés por manejar problemas cada vez mayores se puede constatar en el tamaño de las bases de caras disponibles. Uno de los conjuntos *clásicos*, la base ORL (Olivetti

Research Laboratory) [159], contenía 40 individuos, con 10 imágenes por persona. Posteriormente aparecen otras muchas, como la base AR [120], con 116 sujetos, o la base ESSEX [82], creada entre 1994 y 1996, que incluye unas 400 personas. En la actualidad, una de las mayores bases de caras disponibles de forma pública es la utilizada en las evaluaciones FERET [52], abierta a la investigación en 2001, y con casi 1200 individuos diferentes. Superando a todas ellas –aunque de uso no público–, la base del programa FRVT 2002 [13], consta de más de 120.000 imágenes de unas 37.000 personas. Precisamente, usando esos datos se estableció que el descenso del rendimiento es aproximadamente de 2 ó 3 puntos cada vez que se duplica el tamaño de la galería.

- **Número de muestras por individuo**

  A medida que ha ido aumentando el número y el tamaño de las bases disponibles, ha disminuido también el número de muestras por persona usado para la galería. Por ejemplo, aunque la base FERET contiene cerca de 10 imágenes de cada sujeto, las galerías predefinidas se limitan a 1 imagen por individuo. Existe un consenso implícito en el sentido de que el número requerido de muestras por persona debería ser el menor posible, idealmente una, ya que en la práctica la cantidad de información disponible es muy reducida. Este hecho puede hacer que se descarten muchos métodos tradicionales de clasificación de patrones, que requieren un elevado número de ejemplos por clase.

- **Cambios de iluminación**

  La mayor dificultad relacionada con los cambios de iluminación son las variaciones debidas al paso de condiciones de interior a exterior, y viceversa. Refiriéndonos a datos de algunas de las evaluaciones antes mencionadas, el uso de distintas condiciones de iluminación en interior –véase, por ejemplo, el contraste entre las figuras 6.8a) y 6.8c)– puede suponer una reducción desde el 95 % hasta el 82 % (algoritmo USC [129], sobre la base FERET); mientras que para un cambio interior/exterior puede bajar hasta el 55 %.

- **Cambios de pose**

  Las variaciones debidas a una modificación en la posición 3D relativa de la cara son especialmente complejas cuando sólo se dispone de una muestra por individuo. El rendimiento de las técnicas existentes no resulta alterado hasta ángulos de $\pm 25^{o}$. Pero más allá, los porcentajes de reconocimiento caen de forma significativa. Para reducir el problema se han propuesto modelos deformables y técnicas similares, que tratan de compensar los giros, obteniendo una vista virtual de la cara en una posición normalizada.

Por otro lado, algunas fuentes de variabilidad parecen estar más controladas. Las diferencias de expresión facial, la resolución y la pérdida de información por compresión no afectan significativamente a la mayoría de las técnicas existentes [108].

6.2. El estado del arte en reconocimiento de personas

Es difícil sintetizar con precisión el ingente número de publicaciones que en más de treinta años han aparecido en relación al reconocimiento de caras humanas. Por número de artículos disponibles[8], el volumen casi duplica al de la detección de caras, y multiplica por diez al de seguimiento. Pero pretender listar de forma exhaustiva los trabajos existentes se sale claramente de los objetivos de esta tesis. En su lugar, nos limitamos a hacer un repaso somero de los principales acercamientos al problema y algunos de los trabajos originales y más relevantes dentro de cada uno.

Tradicionalmente, el reconocimiento visual de caras ha usado imágenes estáticas, mayoritariamente en escala de grises. Recientemente, algunas variantes del problema han suscitado un creciente interés, como el reconocimiento en secuencias de vídeo (véase, por ejemplo, el capítulo 8 de [108]), el uso de imágenes de infrarrojos (se puede encontrar un repaso de métodos en [100]), y el reconocimiento tridimensional de las caras. Uno de los trabajos con mayor impacto de este último grupo es el de los hermanos Bronstein y otros [17], que en 2003 demostraron por primera vez la capacidad de distinguir dos gemelos idénticos[9]. No obstante, en lo sucesivo nos vamos a centrar en el reconocimiento de imágenes estáticas 2D.

En la completa revisión de la literatura de Zhao y otros [212], se distinguen tres grandes **categorías de técnicas de reconocimiento**, en función de que las imágenes se traten de forma global o por partes:

- **Métodos holísticos.** La cara se representa como un todo, sin distinguir explícitamente partes dentro de la misma. Las propuestas van desde la clásica técnica de autocaras [183], hasta el uso de redes neuronales [111], SVM [91], y proyecciones en subespacios mediante LDA [9], ICA [6], o PCA probabilístico [125], ente otras.

- **Métodos basados en características.** El reconocimiento se realiza en base a características extraídas de la cara o de partes de la misma (de los componentes faciales). Estas características pueden ser distancias entre componentes faciales [93], respuestas a filtros de Sobel [56], filtros de *wavelet* [103, 192], filtros frecuenciales [127], integrales proyectivas [190], autoespacios asociados a ojos y boca [137], etc.

- **Métodos híbridos.** Incorporan al mismo tiempo información global de la cara y propiedades asociadas a los elementos faciales. Puede ser, por ejemplo, a través de una simple ponderación relativa [137], o con mecanismos que consideren al mismo tiempo ambos tipos de propiedades, como los modelos deformables 2D [105], y 3D [14].

Otros autores introducen categorizaciones ligeramente diferentes. Por ejemplo, Lu [114], clasifica los métodos según estén *basados en apariencia* o *en modelos*. Grosso modo, los primeros equivalen a los que Zhao y otros denominan holísticos; mientras que los métodos híbridos y

[8] Datos obtenidos de: `http://scholar.google.com`

[9] Curiosamente, los dos gemelos utilizados en los experimentos eran los propios hermanos Bronstein.

los que usan características están mayoritariamente basados en modelos de cara predefinidos. Por su parte, en otra revisión interesante y muy reciente, Kong y otros [100], realizan una agrupación similar, señalando el papel dominante de las técnicas basadas en apariencia.

En el resto de esta sección adoptamos la clasificación en 3 grupos de Zhao y otros [212]. Vamos a profundizar en cada uno de ellos, mostrando los trabajos más significativos que han sido propuestos hasta la fecha. En el apartado 6.2.1 tratamos los métodos holísticos o basados en apariencia; en el apartado 6.2.2 revisamos los basados en características; y, finalmente, describimos las técnicas híbridas más relevantes en el apartado 6.2.3.

6.2.1. Métodos holísticos de reconocimiento

Al trabajar con las imágenes en sí mismas, uno de los desafíos de los métodos holísticos es la necesidad de manejar espacios de elevada dimensionalidad. Por ello, típicamente las resoluciones usadas para el reconocimiento no sobrepasan los 20×20 píxeles. Para reducir aún más el espacio de entrada, las técnicas holísticas suelen apoyarse en la proyección a subespacios lineales o no lineales. La comparación entre muestras se realiza dentro de estos espacios proyectados.

Reconocimiento mediante autocaras

Alrededor de finales de la década de 1980 [168], se comprobó la capacidad de obtener buenas reconstrucciones de las caras en espacios de reducida dimensionalidad, usando *análisis de componentes principales* (PCA). Turk y Pentland [183], son los primeros en aprovechar esta idea para construir un sistema completo de detección y reconocimiento facial. El método que plantean es bastante elemental. A partir de las imágenes de la galería, se obtienen los autovectores de la matriz de covarianzas –que denominan las *autocaras*–, descartando los de menor autovalor asociado (ver una muestra en la figura 6.9a). Los ejemplos de la galería se representan como puntos en ese autoespacio, dados por la proyección en la base de autocaras. Dada una imagen nueva, se proyecta igualmente en el espacio de caras. La clasificación se realiza por simple distancia euclídea mínima en el autoespacio.

Este trabajo supuso una de las primeras propuestas de gran calado en el campo del reconocimiento, y ha sido la inspiración de muchas investigaciones posteriores, que han mejorado ampliamente sus resultados. A pesar de ello, aún se suele aplicar como un método base para la comparación, como veremos en los experimentos de la sección 6.4.

Posteriormente, Moghaddam y Pentland [125], extendieron el método básico de autocaras. La mejora fundamental consistía en sustituir el simple criterio de vecino más próximo en el autoespacio de caras, por una medida probabilística. En esencia, para cada clase se modela la función de densidad de probabilidad de las variaciones *intra-clase* y las *entre-clases* (o *inter-clases*), suponiendo distribuciones gaussianas en autoespacios separados para cada uno (los llamados *autoespacios duales de caras*). Se puede ver un ejemplo de estos autoespacios duales en la figura 6.9; la parte 6.9b) contiene el asociado a las variaciones *intra-clase* y la figura 6.9c)

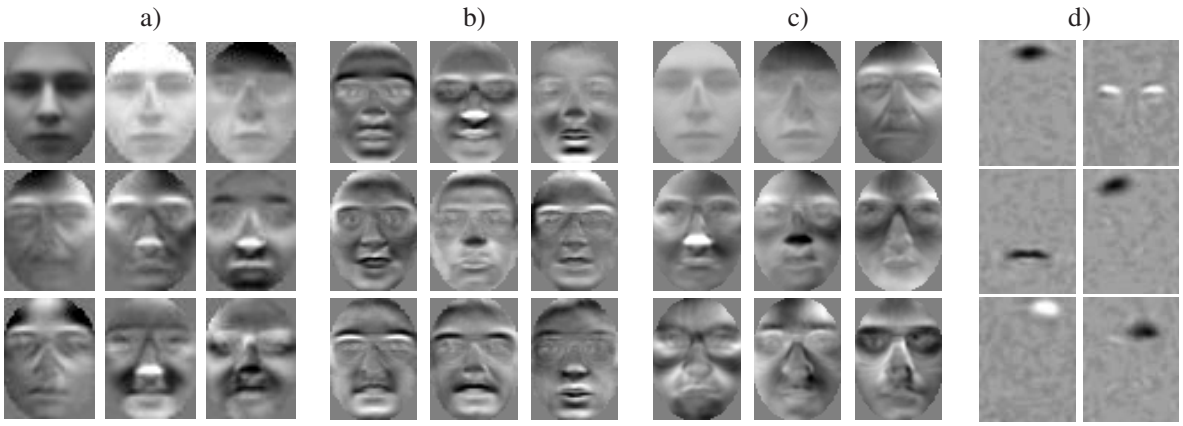


Figura 6.9: *Distintas descomposiciones en autoespacios de la base de caras ESSEX. a) Cara media y autovectores asociados a la matriz de covarianzas (técnica original de autocaras [183]). b) Autovectores de la matriz de dispersión intra-clase. c) Autovectores de la matriz de dispersión entre-clases –véase el gran parecido con a)–. d) Ejemplos de componentes extraídos con ICA (tomado de [6]).*

a las *entre-clases*.

Dada una imagen nueva, se obtiene la diferencia píxel-a-píxel respecto de cada muestra de la galería. Según el *criterio de máxima verosimilitud*, la probabilidad de pertenencia a una clase es mayor cuando esa diferencia sea "explicable" con los modos de variación *intra-clase*. En el llamado *criterio máximo a posteriori*, se tiene en cuenta también la distancia en el espacio *entre-clases*; en particular, la distancia en el espacio *intra-clase* debería ser menor que en el *entre-clases* para las muestras de la clase correcta.

Reconocimiento mediante análisis de discriminantes lineales

Una de las críticas realizadas a la descomposición mediante PCA es que está orientada a la reconstrucción, por lo que puede no ser adecuada en problemas de discriminación entre clases. Para paliar este inconveniente se ha propuesto el uso de *análisis de discriminantes lineales* (LDA) [9], fundamentado en el criterio de *separabilidad* de Fisher. A diferencia de PCA, donde se calcula la matriz de covarianzas total, en estos métodos se usa una matriz de dispersión *intra-clases* (acumulando las covarianzas dentro de cada clase del conjunto) y otra de dispersión *entre-clases* (covarianza obtenida con los centroides de cada clase). El objetivo es calcular un subespacio lineal en el que se minimice la varianza de la primera, maximizando la de la segunda. Básicamente, esto se consigue resolviendo un problema de autovalores generalizado.

En [9], Belhumeur y otros aplican esta estrategia, pero no directamente sobre las imágenes de entrada sino sobre la proyección de las mismas utilizando PCA. Es decir, se utiliza PCA en primer lugar, y el resultado se proyecta después con LDA[10]. Al final del proceso, los vectores asociados a cada imagen son de tan sólo 4 dimensiones. Aun así, el rendimiento conseguido

[10]Obviamente, se puede obtener el mismo resultado con una sola proyección, combinando las bases de PCA y LDA. Los elementos de esta nueva base resultante son llamados las *fisher-caras*.

mejora con creces el de los métodos de autocaras; en la base de caras Yale (160 imágenes de 16 personas) reducen los ratios de error del orden de 20 puntos. De forma colateral, un dato interesante que aportan los autores es que todos los métodos obtienen mejores resultados con un recorte amplio de la cara –"y parte del fondo" [9]–, que con una selección pequeña ajustada a ojos, nariz y boca.

Pero el uso de LDA presenta también algunos problemas y limitaciones. El más evidente es que necesita muchos ejemplos de entrenamiento por clase, lo cual no siempre se puede garantizar. Es más, ya hemos visto que la tendencia actual es a usar una sola imagen por persona. Otra de las limitaciones es el elevado coste de entrenamiento del proceso.

Más recientemente, Shan y otros [164], han planteado un método que reduce estos problemas. Por un lado, los autoespacios no se recalculan para cada nueva galería, sino que demuestran la posibilidad de usar una base estándar y precalculada, tanto para la descomposición PCA como para LDA. Por otro lado, la matriz de dispersión *intra-clases* es *regularizada*, esto es, se le suma la identidad multiplicada por cierta cantidad. De esta forma, si sólo se dispone de un ejemplo por individuo, el resultado es equivalente a usar únicamente PCA. Además, la medida de comparación en los valores proyectados es una medida ponderada. Los resultados obtenidos son excelentes, y demuestran la capacidad de generalizar con individuos no usados en el entrenamiento (la galería usada consta de 46 individuos, usando sólo 25 para la obtención de la base).

Otros métodos de proyección en subespacios

En relación con la mejora de las capacidades de generalización, algunos autores han sugerido estrategias basadas en los principios expuestos por Vapnik [185]: minimización del riesgo estructural, mediante la maximización del margen en las fronteras de decisión. Por ejemplo, en [113] se aplica una estrategia de *persecución evolutiva* (en inglés, *evolutionary pursuit*), para encontrar la proyección que optimiza este criterio. La idea subyacente es un algoritmo *genético*, que parte del autoespacio obtenido con PCA, y va realizando sucesivas rotaciones aleatorias. Existe una función que pondera el criterio a maximizar, y que permite seleccionar la proyección más adecuada entre las generadas en el proceso. Los resultados demuestran una mejor capacidad de generalización que el método de las *fisher-caras* [9].

Otra variante destacable de los métodos de proyección es la utilización de *análisis de componentes independientes* (ICA). ICA se puede entender como una extensión de PCA. Mientras que el segundo trabaja con los momentos estadísticos de segundo orden (las covarianzas), ICA busca una decorrelación de los momentos de mayor orden. Se han propuesto dos arquitecturas distintas [6], ambas partiendo de una base de 200 autocaras obtenidas con PCA. En la primera arquitectura, se usa ICA para extraer características que varían de manera independiente. En la segunda, lo que se busca es la independencia en las salidas. En otras palabras, en la primera arquitectura se aplica ICA a las 200 autocaras, y en la segunda se aplica a las proyecciones de los ejemplos en el espacio 200-dimensional.

Las bases del autoespacio en la primera arquitectura presentan una interesante propiedad de localidad: hay una centrada en una ceja, otra en un ojo, en la mejilla, etc. La figura 6.9d) contiene una muestra de una descomposición obtenida con ICA. Unos resultados muy similares se alcanzaron a través de una técnica distinta conocida como *análisis de características locales* (LFA) [135]. La novedad de estos métodos surge de introducir restricciones de localidad y topografía en la auto-descomposición. De esta manera, los elementos de la base representan características locales, y tienen asociado un índice de posición –a diferencia de PCA, donde los elementos se ordenan por el autovalor–. La técnica LFA es aplicada en el sistema comercial FaceIt, señalado como uno de los más exitosos [100].

Reconocimiento mediante redes neuronales

Las redes neuronales también han sido usadas para el problema de reconocimiento de caras [111]. La arquitectura clásica del perceptrón multicapa parece resultar inadecuada para este problema, por lo que Lin y otros introducen las llamadas *redes neuronales basadas en decisión probabilística* (PDBNN). En esencia, la idea consiste en crear una subred neuronal sencilla para cada una de las clases, que produzca valores altos para las instancias de la misma. El resultado final se obtiene tomando el máximo de las subredes existentes.

El sistema propuesto en [111] resuelve también la detección de la cara y la localización de los componentes. Una peculiaridad de este reconocedor es que sólo usa la parte superior del rostro –al contrario que en los restantes métodos descritos hasta ahora–, descartando la boca. La entrada de la red neuronal está compuesta por una imagen de sólo 14×10 píxeles, con las intensidades normalizadas, y otra imagen de bordes del mismo tamaño.

En los experimentos documentados se usan varias imágenes por persona. Algunos autores [212], argumentan que esta estrategia presentará dificultades en el caso de una sola muestra por individuo. También ponen en duda la viabilidad del método cuando el número de clases supere unos cuantos miles.

6.2.2. Métodos basados en características

Uno de los inconvenientes de los métodos holísticos es que se apoyan en la apariencia global de la cara. Si ésta se modifica –por ejemplo debido a la iluminación, a la aparición de bigote, o a un giro significativo– fracasarán con mucha probabilidad. De hecho, es sorprendente el número de trabajos basados en apariencia donde no se elimina el fondo de la escena[11] al trabajar con las imágenes de las caras.

En el acercamiento basado en características se busca la extracción de información robusta frente a los factores mencionados, para después definir métricas de distancia sobre las propiedades extraídas. El número de publicaciones existentes en este grupo es relativamente inferior

[11] Posiblemente porque, como ya hemos mencionado que ocurre en [9], el rendimiento baja al seleccionar un recorte próximo de cara. La interpretación *optimista* de este hecho es que el contorno de la cabeza tiene un papel muy importante en la identificación de las personas. La *pesimista* es que los experimentos se están llevando a cabo en condiciones poco realistas.

al del enfoque holístico. Vamos a detallar algunas de las más relevantes.

Métodos geométricos

Dentro de esta categoría se sitúan algunos de los trabajos pioneros en el reconocimiento facial de personas, que se remontan a principios de los 1970. Así, en 1973 Kanade propone un sistema completo de localización y reconocimiento de caras [93], basado en propiedades geométricas del rostro. El método desarrollado utilizaba integrales proyectivas para localizar la cara y algunos puntos de la misma, como las esquinas de los ojos y de la boca, los orificios nasales y el borde de la barbilla. Posteriormente, el algoritmo calcula un vector de distancias y ángulos entre los puntos obtenidos; la clasificación se realiza por simple distancia mínima de estos vectores con los ejemplos de la galería.

Algunos trabajos más recientes, como [36], han retomado esta idea, aunque parece un camino con poco futuro: la localización exacta de los puntos característicos es problemática en circunstancias no triviales (como pudimos ver en el capítulo 4); la constancia de las propiedades geométricas no está garantizada con diferentes expresiones faciales; y tampoco su unicidad con tamaños de galería muy grandes.

Más prometedora parece la idea de extraer segmentos de las imágenes. Gao y otros [56], definen los llamados *mapas de líneas de bordes* (LEM), que son obtenidos aplicando filtros de Sobel y agrupación de los bordes en segmentos. La comparación entre imágenes se lleva a cabo mediante una variante de la distancia de Hausdorff.

Métodos basados en integrales proyectivas

El uso de integrales proyectivas en el reconocimiento de caras no ha sido excesivamente explotado hasta la fecha. Posiblemente, el trabajo más destacado es el debido a Wilder [190], que participó en las primeras fases de las evaluaciones del programa FERET [143]. Por su especial relación con el contexto de esta tesis, vamos a hacer una descripción detallada del proceso propuesto por Wilder. Debemos mencionar que el sistema incluía un sencillo método heurístico de localización de caras –basado en detección de bordes y proyecciones–, aunque aquí nos limitaremos a la parte de reconocimiento. El algoritmo consta de tres pasos:

1. En primer lugar, se obtiene la **proyección vertical de la cara**, desde la frente hasta la barbilla. Realmente, para conseguir invarianza frente a rotaciones, se calculan tres proyecciones de cada imagen: la vertical y las proyecciones en ángulos de 7º y -7º.

2. Para reducir aún más la información manejada, se aplica la **transformada del coseno** (DCT) sobre las proyecciones resultantes del paso anterior. Además se descartan los componentes de mayor frecuencia (justificándolo en que aportan poca información y están muy influidos por el ruido).

3. Por último, para clasificar los descriptores frecuenciales obtenidos se utiliza una **red neuronal jerárquica**, que combina árboles de decisión. Se argumenta que esta red tiene,

por lo menos, tan buen rendimiento como una clasificación por distancia mínima. De hecho, en la evaluación FERET [143], no se utiliza la red sino una norma L_1 sobre los vectores obtenidos en el paso 2. Para cada clase, se toma la menor de las distancias con las 3 proyecciones disponibles.

Uno de los aspectos más discutibles de este método es la aplicación de DCT sobre las proyecciones, en lugar de usarlas directamente, como es el caso del algoritmo que proponemos nosotros en la sección 6.3. Aplicando el *teorema de la sección central* (ver la página 46), podemos deducir que calcular la DCT de las proyecciones verticales es equivalente a obtener la DCT de la imagen original y quedarse con una sección vertical de la misma. Sin embargo, esta observación parece no tenerse en cuenta en [190]. En consecuencia, el paso de proyección es realmente irrelevante en el proceso –excepto como una mejora del coste computacional–, y en última instancia el algoritmo descansa en la capacidad expresiva de la DCT.

La evaluación de este método en el programa FERET [143], llevada a cabo en 1994, produjo unos resultados ciertamente pobres[12]. Por ejemplo, en una ejecución con 317 individuos en la galería y 780 en la prueba, el porcentaje de identificación correcta sobrepasa muy ligeramente el 30 %. Como contraste, en un ensayo similar de nuestro método sobre la base FERET pero con 1196 individuos y 1195 pruebas, el ratio de identificación se sitúa en el 81,5 % (los datos concretos se pueden consultar en la página 363, y sucesivas).

Métodos basados en grafos de propiedades

Uno de los acercamientos más exitosos en general, y no sólo entre los métodos basados en características, son los que utilizan *emparejamiento de grafos* (en inglés, *graph matching*). En estos métodos, la cara es representada a través de un grafo cuyos nodos están asociados a puntos característicos –como las esquinas de los ojos, de la boca, o la nariz–. Los grafos almacenan información tanto en los nodos como en las aristas.

En la propuesta de Lades y otros [103], cada nodo contiene un conjunto de coeficientes dados por las respuestas a una serie de filtros wavelet de Gabor, llamados *jets* –ver las figuras 6.10a) y 6.10b)–. En concreto, cada *jet* consta de filtros de Gabor con 5 frecuencias y 8 orientaciones distintas, lo que da lugar a un vector de 40 coeficientes complejos por nodo. Por su parte, las aristas almacenan información de distancia. En la figura 6.10c) se puede ver un ejemplo simplificado de esta arquitectura.

Posteriormente, esta arquitectura se extendió a la denominada *elastic bunch graph matching* (EBGM) [192]. La principal diferencia es que cada nodo del grafo puede contener varios *jets*, en lugar de uno solo; es el denominado *bunch*, o *racimo* –mostrado en la figura 6.10d)–. Al hacer un emparejamiento de grafos, se puede seleccionar cualquier *jet* del racimo, con total independencia entre los distintos nodos.

En el proceso de entrenamiento, se crea un *grafo de racimos* para cada clase, usando las posiciones etiquetadas a mano y una estructura de grafo predeterminada –es decir, el número

[12] Y, posiblemente, fue la causa de su abandono del programa en las fases sucesivas.

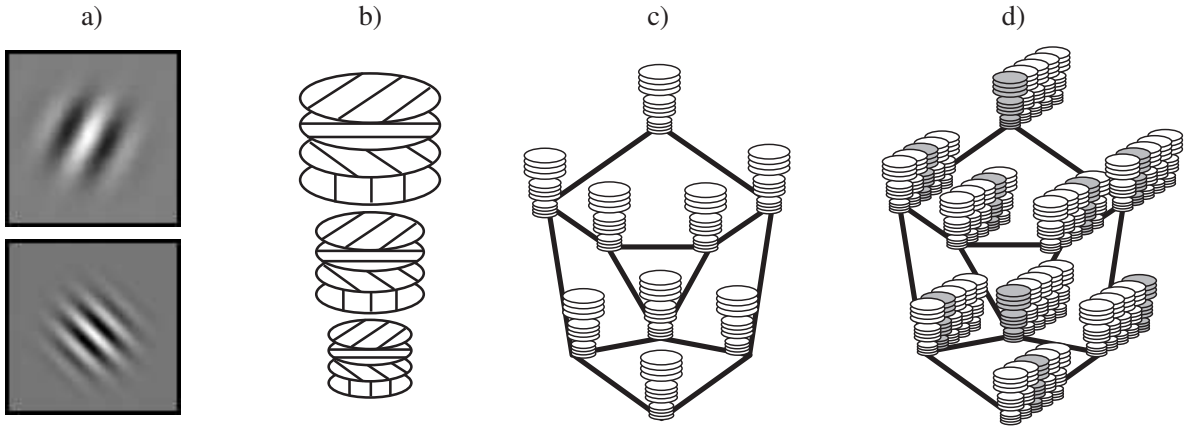


Figura 6.10: *Reconocimiento de caras mediante emparejamiento de grafos. a) Dos filtros wavelet de Gabor, con distinta frecuencia y orientación. b) Un jet consta de varios filtros de Gabor, aplicados sobre el mismo punto. c) El modelo de grafo en la técnica Dynamic Link Architecture (DLA) [103], contiene las respuestas de un jet en cada nodo. d) En Elastic Bunch Graph Matching (EBGM) [192], cada nodo contiene un conjunto, o racimo, de jets entre los que se puede seleccionar uno.*

de nodos y aristas son iguales para todas las clases–. En la clasificación de una cara nueva, se sitúa el grafo en una posición inicial y se realiza un proceso búsqueda, maximizando la similitud con el modelo de grafo correspondiente. La medida de similitud entre un modelo y la instancia actual tiene en cuenta tanto los valores de los *jets* como las distancias de las aristas.

Estas técnicas han demostrado una notable capacidad para afrontar situaciones de cambio de orientación e iluminación. Además, el mecanismo de EBGM ha sido aplicado en detección de caras, localización de componentes, estimación de pose y clasificación del sexo. Como contrapartida, una de sus limitaciones es que necesita imágenes de alta resolución, del orden de 128×128 píxeles [100], lo cual no siempre es posible cuando se usa vídeo.

Métodos basados en modelos de Markov

Otro de los enfoques que ha recibido mucho interés son los *modelos ocultos de Markov* (HMM). Originalmente, esta técnica está orientada al modelado probabilístico de procesos temporales, por lo que su entrada es una secuencia de observaciones en una dimensión. Para aplicarlo a las imágenes se debe llevar a cabo una conversión adecuada. La técnica más elemental es el escaneado por bandas de la imagen de cara. Samaria [159], sugiere un procesamiento de la imagen de arriba abajo, extrayendo bandas solapadas de píxeles. Cada una de ellas se pasa al modelo como una observación en un proceso secuencial 1D.

Posteriormente, Nefian y Hayes [127], extienden la técnica a los llamados *HMM embebidos* o *pseudo-2D*. En el método descrito, la imagen se escanea por bloques de 8×10 píxeles, de arriba abajo y de izquierda a derecha, con un amplio margen de solapamiento entre regiones vecinas. Además, la entrada al proceso no son directamente los niveles de gris, sino unos coeficientes obtenidos con DCT o bien mediante autodescomposición. El modelo consta de

varios estados ocultos (asociados a la frente, ojos, nariz, boca y barbilla), cada uno con varios subestados a su vez. Durante el entrenamiento, se aprenden las probabilidades de transición entre estados para cada clase. En la clasificación, se aplican los diferentes modelos, devolviendo el que produzca mayor probabilidad de haber sido generado.

Es interesante hacer notar que el método de reconocimiento propuesto en [127] estaba disponible públicamente en las primeras versiones de las librerías Intel OpenCV [35] (aunque ha sido eliminado más recientemente). Gracias a ello, hemos podido incluirlo en algunas de las pruebas comparativas que describimos en la sección 6.4.

6.2.3. Métodos híbridos de reconocimiento

Intentar aprovechar las ventajas de los métodos holísticos –que han tenido una presencia predominante en los sistemas de reconocimiento– y los basados en características locales parece una idea, en principio, bastante interesante. Esencialmente, podemos distinguir dos subcategorías de sistemas que siguen el enfoque mixto. La más elemental consiste en incorporar características locales y globales en el clasificador. La otra alternativa son los modelos deformables, de los cuales se han propuesto variantes 2D y 3D.

Reconocimiento con autoespacios modulares

Extendiendo la técnica de las autocaras, Pentland y otros [137], sugieren la utilización de autoespacios asociados a los distintos elementos faciales. De esta manera, surgen los conceptos de *auto-ojos*, *auto-narices* y *auto-bocas*, que son ahora la entrada de los clasificadores. Es más, como en el caso de las caras, la medida del error de reconstrucción se puede utilizar también para el problema de localización (como vimos en el capítulo 4).

En un conjunto de 45 individuos, con 2 imágenes por persona, los *auto-componentes* consiguen alcanzar mejores resultados que las autocaras. También se estudia la aplicación conjunta de ambos en el reconocimiento; aunque, en este caso, las mejoras conseguidas son muy reducidas. No obstante, la gran potencia de esta técnica se encuentra en la capacidad de abordar situaciones complejas, como la oclusión parcial o la aparición de elementos faciales, como barba, bigote o gafas.

Modelos deformables 2D

A lo largo de los capítulos anteriores hemos ido viendo cómo los modelos deformables 2D ayudan en la resolución de los problemas de localización, extracción, y seguimiento de caras. También en el reconocimiento pueden ser de gran utilidad, reduciendo los cambios de apariencia debidos a expresiones faciales y pequeñas variaciones de pose. El esquema subyacente de estos métodos es como el siguiente [114]: (1) construir el modelo manualmente con los ejemplos de entrenamiento; (2) ajustar el modelo a cada imagen nueva; y (3) realizar la clasificación usando los parámetros obtenidos en el ajuste del modelo.

Uno de los ejemplos más destacados es el trabajo de Lanitis y otros [105]. El *modelo de apariencia activa* (AAM) que proponen está compuesto por un modelo de forma (que determina la estructura interna de la cara) y un modelo de apariencia (que determina la textura). Estos modelos pueden variar en un número de modos predefinido, como se ilustra en la figura 6.11. Como resultado del ajuste del modelo a una imagen, se obtiene información de tres tipos: los parámetros que indican el ajuste del modelo de puntos; la información local de niveles de gris asociados a cada punto (en concreto, perfiles de intensidad); y la descomposición PCA de la cara tras una normalización de la forma (en el llamado *modelo libre de forma*). La clasificación se realiza usando LDA sobre estos valores.

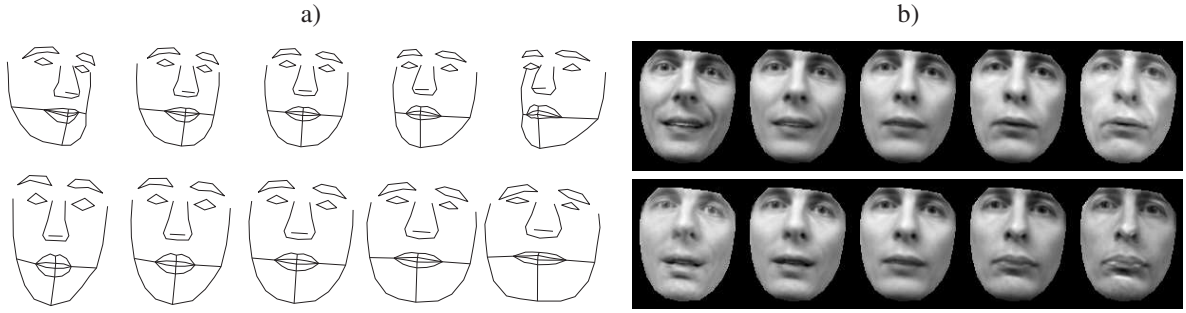


Figura 6.11: *Reconocimiento de caras mediante modelos de apariencia activa [105]. Se trata de un modelo deformable 2D, con dos componentes: a) Modelo de forma (puntos característicos); b) Modelo de textura. Se muestran dos modos de variación de cada uno. Información extraída de:* `http://www.isbe.man.ac.uk/~bim/`

En una galería de 30 individuos, el sistema alcanza un 92 % de reconocimiento para imágenes en condiciones normales, y un 48 % para un subconjunto de casos más complejos. Aunque, en general, la idea parece muy interesante, podemos achacarle algunos de los problemas ya mencionados para otros sistemas: alto coste del proceso de ajuste, necesidad de elevada resolución, y varias muestras por persona. Otro inconveniente es la laboriosidad del proceso de entrenamiento, ya que se deben señalar manualmente los puntos característicos (hasta más de 60) en cada imagen de entrada.

Modelos deformables 3D

En los modelos AAM bidimensionales, el propio mecanismo de construcción permite pequeñas rotaciones de la cara. No obstante, para conseguir una verdadera robustez frente a la pose, se deben utilizar modelos que trabajen explícitamente con información 3D.

El funcionamiento típico de estos sistemas, como el presentado por Blanz y otros [14], resulta parecido al de los modelos 2D: (1) construir un modelo; (2) ajustarlo a los ejemplos; y (3) definir una medida de similitud sobre los parámetros del ajuste, para llevar a cabo el reconocimiento. En concreto, los parámetros del modelo expuesto en [14] son de varios tipos: parámetros de forma, de textura (separando los de la cara, los ojos, y la boca); parámetros de pose (ángulos y posición 3D de la cabeza); intensidad de luz y parámetros de color.

Evidentemente, el proceso de ajuste resulta muy costoso por el elevado número de incógnitas que se deben resolver. El método busca la combinación que genere un resultado lo más parecido posible a la imagen de entrada. La clasificación utiliza una función de similitud, descartando los parámetros de pose e iluminación. Debemos mencionar que los modelos deformables 3D han sido usados también en la generación de vistas virtuales 2D, que después son procesadas con los algoritmos clásicos descritos en los apartados previos.

Como en el caso anterior, las grandes dificultades de este método se encuentran en el costoso proceso de ajuste y la dificultad de crear el modelo. Además, el algoritmo debe ser inicializado en una posición próxima a la correcta, ya que el proceso puede caer con facilidad en mínimos locales. Desafortunadamente, no disponemos de datos comparativos de este método que nos permitan contrastarlo con los anteriores.

6.3. Reconocimiento de personas mediante proyecciones

Observando globalmente el estado del arte en este activo ámbito de investigación, podemos apreciar que existe una cierta tendencia hacia la utilización de técnicas cada vez más complejas. De la extracción de características geométricas [93], se pasó a la reducción a subespacios lineales [183, 125, 9], de ahí a los modelos deformables [105, 14], y las propuestas más recientes apuntan al uso de información 3D de las caras como el futuro de la disciplina [17]. En este sentido, nuestro propósito aquí no es tratar de competir con las técnicas más elaboradas, sino hacer una incursión superficial en el problema, analizar el uso potencial de las proyecciones y compararlo con otros métodos dentro de la misma categoría de técnicas.

Ya hemos visto que el reconocimiento de caras se puede plantear desde dos puntos de vista alternativos: (1) como un problema de clasificación multiclase; o (2) centrándose en la definición de una medida de similitud entre las muestras de la galería y las pruebas. La novedad del método que proponemos reside en que el vector de características asociado a cada muestra biométrica es un conjunto de integrales proyectivas de la cara de entrada. Sobre ellas se aplica el alineamiento y las medidas de distancia definidas en el capítulo 2.

El resto de esta sección está estructurado de la siguiente forma. Empezamos en el apartado 6.3.1 discutiendo y justificando la viabilidad del uso de integrales proyectivas en este problema, mostrando la idea básica con un ejemplo sencillo. Después, los apartados 6.3.2 y 6.3.3 tratan las diversas cuestiones que deben ser abordadas en la definición del proceso de reconocimiento: las medidas de distancia entre proyecciones, el método de clasificación subyacente, y la selección de las regiones más identificativas. Terminamos, en el apartado 6.3.4, analizando cómo combinar el uso de varias proyecciones sobre unas mismas imágenes, con lo cual queda especificada completamente nuestra propuesta.

6.3.1. Justificación del uso de proyecciones

Existen varias evidencias que dan sentido a la utilización de proyecciones en el reconocimiento de personas. Desde un punto de vista biológico, son muchos los parámetros que intervienen en la identificación visual de un individuo: la situación relativa de los elementos faciales –como las distancias entre cejas, ojos y boca–, la forma y tamaño de la nariz y los ojos, el grosor de las cejas, las arrugas, la barba, el pelo y otras características faciales, etc. Las proyecciones conservan implícitamente buena parte de esta información, reduciéndola a un vector unidimensional. Se puede decir que una proyección de cara describe la "estructura global" del rostro a lo largo de cierta dirección; es una estructura de picos máximos y mínimos, asociados a los elementos faciales, que indican su posición, tamaño y brillo relativos.

De hecho, las proyecciones ya han sido aplicadas previamente en el reconocimiento de personas [190], tal y como hemos descrito en el apartado 6.2.2. Sin embargo, existe una diferencia sustancial respecto del método que nosotros proponemos. En la técnica documentada en [190], se aplica la transformada del coseno (DCT) sobre las proyecciones. De esta forma, se pierde la propiedad fundamental –y que ha sido una de las claves de los mecanismos descritos en los anteriores capítulos– de las proyecciones: la conservación de la continuidad espacial, es decir, de la relación de vecindad entre píxeles, que permite aplicar los procesos de alineamiento. Sea por este o por otros motivos, el método de [190] no obtiene una buena clasificación en los resultados de la evaluación FERET [143].

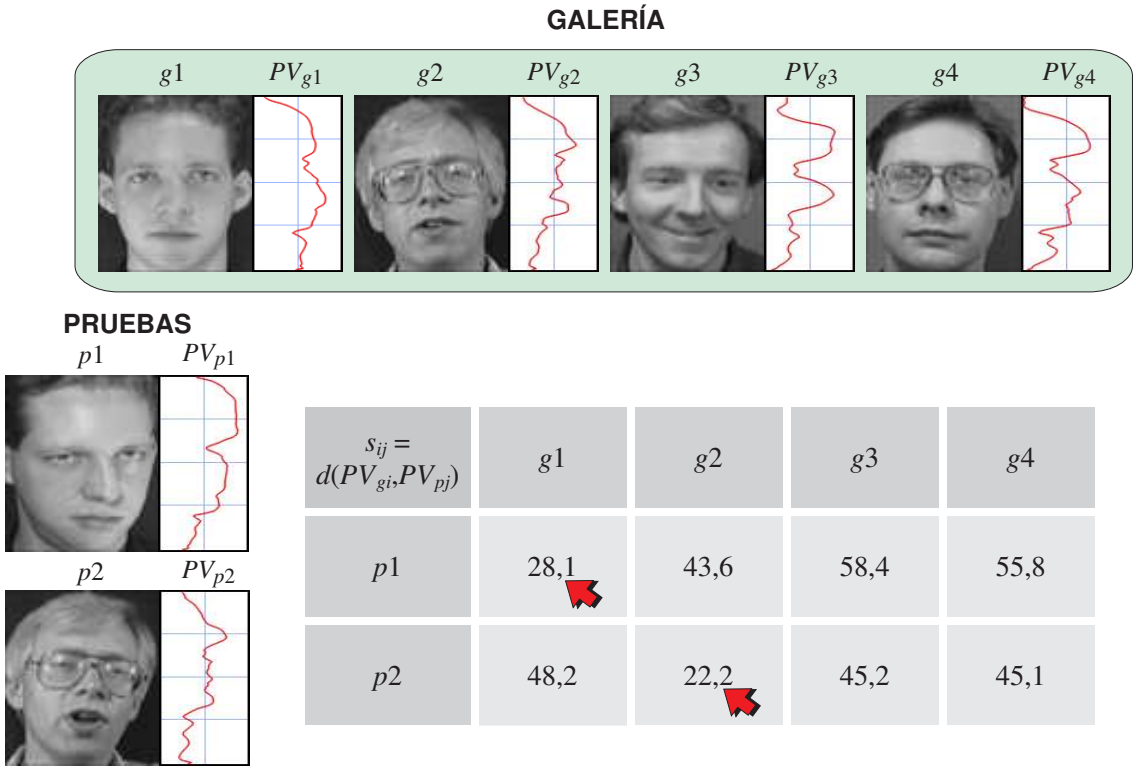
En el capítulo anterior pudimos ver que las proyecciones verticales de la cara –y también las horizontales de los ojos–, mantienen su disposición y forma global para un mismo individuo a lo largo de una secuencia de vídeo. Es decir, la varianza de las proyecciones debida a cambios de expresión, y pequeños giros y variaciones de la iluminación es relativamente reducida, siempre que se trate de la misma persona. Ésta sería la varianza *intra-clase*. Las cuestiones que se plantean ahora son: si la varianza intra-clase es pequeña, ¿es la *inter-clase* suficiente para discriminar a un centenar o un millar de individuos? ¿Qué ocurre con la varianza intra-clase debida al paso del tiempo, o los cambios grandes de iluminación o pose?

Ejemplo sencillo de reconocimiento con proyecciones

En la figura 6.12 se muestra de forma esquemática un primer ejemplo sencillo del proceso de reconocimiento mediante integrales proyectivas. Tenemos una pequeña galería con cuatro imágenes de otras tantas personas, $\mathcal{G} = \{g_1, g_2, g_3, g_4\}$, y dos imágenes de prueba, p_1 y p_2. De cada muestra de entrada –ya sea de la galería o del conjunto de prueba– se extrae la proyección vertical de la cara, PV_{cara}.

Las puntuaciones resultantes, s_{ij}, están basadas en las medidas de distancia entre proyecciones definidas en el capítulo 2. Suponiendo un problema de identificación en conjunto cerrado, se asigna simplemente la clase que produce la menor distancia[13] para cada uno de los ejemplos de prueba.

[13] Como las puntuaciones s_{ij} son distancias, un valor bajo significa aquí una mayor similitud.



$s_{ij} = d(PV_{gi}, PV_{pj})$	$g1$	$g2$	$g3$	$g4$
$p1$	28,1	43,6	58,4	55,8
$p2$	48,2	22,2	45,2	45,1

Figura 6.12: *Ejemplo sencillo de reconocimiento de caras mediante proyecciones. La galería $\mathcal{G}$ consta de 4 caras de 4 personas, g_i, y el conjunto de prueba $\mathcal{P}_\mathcal{G}$ contiene 2 imágenes, p_j. Calculamos para todas las muestras la proyección vertical de la cara, PV_{gi}, PV_{pj}. Las puntuaciones se basan en las medidas de distancia entre proyecciones. La menor puntuación para cada prueba es la identidad reconocida. Imágenes de ejemplo tomadas de la base ORL [159].*

Obsérvese que la prueba p_1 de la figura 6.12 presenta una variación de pose y de tamaño de la cara, respecto de la imagen correspondiente de la galería; mientras que en la prueba p_2 ocurre un cambio de expresión e inclinación del rostro. No obstante, podemos decir que son diferencias no demasiado excesivas. En todos los casos, las regiones proyectadas van desde el pelo hasta la barbilla. No se incluye el fondo –aunque aparezca en las imágenes–, ya que la región proyectada horizontalmente se limita al ancho de cara (aproximadamente, hasta el borde exterior de los ojos).

A pesar de las variaciones mencionadas, ambos ejemplos son identificados correctamente en este sencillo caso. Considerando un problema de verificación, el resultado sería también perfecto. Por ejemplo, estableciendo el umbral τ a cualquier valor entre 29 y 43, se consiguen eliminar todas las falsas alarmas, para un porcentaje de verificación del 100 %. Las pruebas se aceptarían si la puntuación está por debajo de τ, y se rechazarían si están por encima. También para una identificación en conjunto abierto se podría usar ese mismo umbral, llegando al 100 % de detección e identificación.

Obviamente, el reducido ejemplo de la figura 6.12 no tiene más valor que el de describir el fundamento del proceso de reconocimiento mediante integrales proyectivas: calcular proyecciones de las caras y utilizar medidas de distancia para llevar a cabo la clasificación. Aparecen

muchos factores que pueden tener una influencia significativa en la efectividad del método: qué mecanismo de clasificación se utiliza, qué medida de distancia aplicar, qué regiones se proyectan de las caras, qué canales (en el caso de imágenes en color), y cómo usar más de una proyección de cada rostro. En los apartados sucesivos vamos a analizar las diferentes alternativas, haciendo uso de algunas bases de caras disponibles públicamente. Con las conclusiones del estudio, realizamos una propuesta para la resolución del problema.

Bases de caras para el análisis de alternativas

Cuando experimentamos en reconocimiento de personas, resulta siempre conveniente utilizar más de una base de caras, ya que muchas de ellas se centran exclusivamente en determinados aspectos. En nuestro caso, los conjuntos de imágenes con los que vamos a trabajar son los siguientes:

- **Base de caras FERET.** Ya hemos comentado que esta base de caras es una de las mayores entre las disponibles públicamente; en la figura 6.8 presentamos algunos ejemplos. Además de las más de 14.000 imágenes que incluye, existen ficheros donde se agrupan por diferentes criterios: variación de expresión (grupo "fb"), variación de iluminación (grupo "fc"), duplicados ("dup1" y "dup2"), distintos ángulos de giro horizontal, además de la galería estándar. Esta última contiene 1196 individuos diferentes, con sólo una imagen por persona.

  En la distribución original, contenida en dos CD, las imágenes eran de 256×384 píxeles, en escala de grises y compresión TIFF. Actualmente se distribuye en dos DVD, con el doble de resolución, en color y sin compresión. No obstante, usaremos las imágenes de la distribución inicial –denominada a veces *gray FERET*– puesto que la mayoría de las evaluaciones disponibles se refieren a esa versión. También a efectos de permitir una mejor comparación, usaremos las etiquetas manuales de las posiciones de ojos y boca, que aparecen en muchas de las imágenes del conjunto.

- **Base de caras ESSEX.** Esta base de caras fue creada por el Dr. Libor Spacek de la Universidad de Essex, Reino Unido [82], y actualmente es accesible de forma pública. El conjunto consta de 4 grupos: *faces94*, *faces95*, *faces96* y *grimace*. De forma subjetiva, podemos decir que están en orden de complejidad creciente. En la tabla 6.1 se describen estos grupos. Todas las imágenes son en color y existen exactamente 20 imágenes por persona. Pueden verse algunos ejemplos en la figura 6.13.

  Para la realización de las pruebas posteriores, hemos seleccionado de cada individuo la mitad de las muestras para la galería y la otra mitad para pruebas. Esto nos servirá de contraste frente la estrategia usada en la base FERET. La partición se hace de forma aleatoria. En esta base, no existe un etiquetado manual, por lo que se ha hecho de forma automática aplicando detección y localización. La detección facial se realiza con el algoritmo combinado Haar+IP y la localización mediante el método basado en integrales

Figura 6.13: *Imágenes de la base de caras ESSEX [82]. De izquierda a derecha, dos imágenes de la misma persona dentro de los grupos "faces94", "faces95" y "faces96".*

proyectivas. Por eso, el número de imágenes usadas (ver la tabla 6.1) es ligeramente inferior al número de las existentes. El ratio total de detección es aproximadamente del 98,5 %, con cero falsos positivos. Posiblemente, la mayor debilidad de esta base es que todas las imágenes de un mismo individuo están tomadas el mismo día.

Parámetro	faces94	faces95	faces96	grimace
Nº personas usadas /existentes	152/152 100 %	67/72 93,1 %	139/152 91,4 %	17/18 94,4 %
Nº img. entrenam.	1517 (99,8 %)	651 (97,2 %)	1358 (97,7 %)	167 (98,2 %)
Nº img. prueba	1517 (99,8 %)	639 (95,4 %)	1337 (96,2 %)	165 (97,1 %)
Resolución (píxeles)	180 × 200	180 × 200	196 × 196	180 × 200
Expresiones faciales	Sí (hablando)	Escasa o nula	Media	Grandes cambios
Variación de pose	Reducida o nula	Traslación y escala	Muy grande	Pequeña traslación
Variación de luz	No	Media	Media/alta	No

Tabla 6.1: *Descripción de los cuatro grupos de imágenes de la base ESSEX. Las imágenes están disponibles públicamente en la web:* `http://cswww.essex.ac.uk/mv/allfaces/`*. (c) Libor Spacek. Sobre las imágenes existentes se ha aplicado el detector Haar+IP, buscando una sola cara, y después el localizador basado en proyecciones. Las caras no detectadas no se usan en el reconocedor. Las personas con menos del 40 % de caras detectadas se eliminan del conjunto.*

- **Base de caras GATECH.** La particularidad de esta base es que las caras ocupan una pequeña fracción de las imágenes, a diferencia de los demás casos (a excepción del conjunto "faces96" de la base ESSEX, donde ocurre algo parecido). La base fue creada en 1999 por investigadores del Instituto Tecnológico de Georgia (Gatech) [127]. El conjunto incluye 750 imágenes de 50 personas (15 por cada una). Las imágenes son en color, comprimidas con el formato JPEG y de 640 × 480 píxeles. La mayoría de ellas fueron obtenidas en dos sesiones diferentes y existen grandes variaciones en la posición del rostro y la expresión facial, como puede apreciarse en la figura 6.14.

  Las posiciones de los elementos faciales tampoco están etiquetadas, de manera que se ha usado el mismo procedimiento que para la base ESSEX. En este caso, el porcentaje de reconocimiento se reduce al 92,5 % –debido, fundamentalmente, a una inclinación facial por encima de los 20º–. En total usamos 694 imágenes, aproximadamente la mitad para la galería y la otra mitad para las pruebas.

Figura 6.14: *Imágenes de la base de caras GATECH [127]. Se muestran tres imágenes de un mismo individuo. Las dos primeras están tomadas el mismo día. La tercera está tomada 2 meses después.*

6.3.2. Mecanismos de clasificación

La selección del método de clasificación de patrones adquiere relevancia cuando la galería contiene varias muestras por individuo. Lógicamente, si sólo disponemos de una imagen por persona, la única alternativa posible consiste en la búsqueda de la mayor puntuación o menor distancia, de manera que el énfasis se sitúa en diseñar una medida de similitud precisa y robusta.

En este apartado, suponemos que tenemos más de una muestra para cada sujeto de la galería, típicamente entre 4 y 10. Esto nos permite poner a prueba diversos mecanismos de clasificación. En concreto, proponemos utilizar algunos clasificadores sencillos, que parten de una distancia definida sobre el espacio de las proyecciones[14].

En adelante supondremos que las muestras de la galería están organizadas en m clases (una por persona), denotadas por $c_1, c_2, \ldots, c_m$. Cada clase consta de k_i ejemplos, para $i \in \{1, \ldots, m\}$. Denotamos los ejemplos por $g_i^{i'}$, con $i' \in \{1, \ldots, k_i\}$, siendo cada uno de ellos una proyección –un vector– de tamaño n. Es decir, $c_i = \{g_i^1, g_i^2, \ldots, g_i^{ki}\}$. Vamos a introducir varios métodos de clasificación para este problema específico. En última instancia, los resultados de la clasificación serán las puntuaciones, s_{ij} de cada clase i con cada muestra p_j; es decir, virtualmente la galería contiene una sola muestra por persona.

- **Clasificación mediante distancia normalizada.**

  Este es el método de clasificación más elemental. De cada clase i, entre 1 y m, se calcula un modelo de proyección media/varianza, aplicando el algoritmo 2.2 (ver la página 56) sobre los conjuntos $\{g_i^1, g_i^2, \ldots, g_i^{ki}\}$, obteniendo el par (m_i, v_i), con la proyección media y la varianza en cada punto, para todas las clases.

  Dada una nueva proyección de prueba, p_j, el primer paso consiste en alinearla con el modelo correspondiente (m_i, v_i). Para ello aplicamos el algoritmo de alineamiento 2.4; llamamos $alin(p_j, i)$ a la muestra alineada con la clase i. Finalmente, la distancia entre

[14] No debemos olvidar que nuestro propósito es evaluar la potencia expresiva de las proyecciones, más allá del mecanismo de clasificación aplicado.

la prueba p_j y la clase i está dada por:

$$s_{ij} = dist\left((m_i, v_i), alin(p_j, i)\right) \tag{6.8}$$

para la definición de *dist* de la ecuación 2.23, es decir, una suma de diferencias al cuadrado, ponderada inversamente por la varianza en cada punto.

- **Clasificación mediante distancia a la media.**

  El método anterior supone que cada posición de las proyecciones tiene una distribución de probabilidad normal. La varianza tiene el papel de aumentar o reducir el peso relativo de cada punto, de acuerdo con la variabilidad observada en el mismo. No obstante, esta variabilidad puede estar más relacionada con las condiciones de entrenamiento para cada individuo, que con los propios individuos en sí. Es más, una clase i cuyas varianzas, v_i, sean en general más altas que el resto, producirá normalmente distancias bajas para todas las muestras (de la misma o de distintas clases). En otras palabras, surge la necesidad de *normalizar* los valores de *dist* entre las distintas clases, de manera que todas ellas produzcan rangos equiparables.

  No es sencillo encontrar una normalización de valores que funcione bien en la práctica. En nuestros ensayos, el método con mejor comportamiento consiste simplemente en anular v_i, es decir, poner todas las varianzas a valor 1. De esta forma, la puntuación sería una distancia a la proyección media de la clase:

$$s_{ij} = dist\left(m_i, alin(p_j, i)\right) \tag{6.9}$$

  donde ahora *dist* es la distancia entre dos señales de la ecuación 2.19 (página 53).

- **Clasificación mediante vecino más próximo.**

  El problema de los dos métodos anteriores es que consideran que las distintas clases tienen distribuciones unimodales, es decir, que existe un modo principal de variación de las proyecciones. Pero esto puede no ser cierto para las caras, donde los modos de variación están en función de muchos y diversos factores. Alternativamente, la clasificación de vecino más próximo es uno de los métodos más sencillos que permite un modelado multimodal de las distribuciones de probabilidad de las clases.

  Usando vecino más próximo, el valor resultante corresponde a la menor distancia del patrón con cualquiera de los ejemplos de la clase. Igual que antes, se aplica un alineamiento previo a la obtención de la distancia, pero ahora repetido para cada muestra de la galería. Llamaremos a la señal alineada $alin(p_j, i, i')$, consistente en alinear p_j con la proyección $g_i^{i'}$. En definitiva, en este método la puntuación se define como:

$$s_{ij} = \min_{\forall i' \in \{1,\ldots,k_i\}} dist\left(g_i^{i'}, alin(p_j, i, i')\right) \tag{6.10}$$

siendo *dist* la distancia de la ecuación 2.19. Obsérvese que este método requiere comparar cada ejemplo de prueba con todos los patrones, y para cada uno aplicar el algoritmo de alineamiento. Afortunadamente su ejecución es muy rápida, como vimos en el capítulo 2, de manera que se puede conseguir una alta eficiencia computacional.

- **Clasificación mediante *k*-vecinos más próximos.**

  Es evidente que la técnica de vecino más próximo presenta un riesgo de sobreajuste mayor que los métodos basados en la proyección media. Un posible inconveniente es que un ejemplo de la galería muy desviado de la media puede producir clasificaciones erróneas para muchas muestras de entrada.

  Una técnica que evita este problema, dentro de las basadas en distancia, es la de *k*-vecinos más próximos. En la clasificación mediante *k*-vecinos, la clase resultante es aquella para la cual la *k*-ésima menor distancia al ejemplo de entrada es la mínima. Por ejemplo, con $k = 1$ sería un simple método de vecino más próximo. En nuestro caso, la puntuación obtenida sería la *k*-ésima menor de la clase:

$$s_{ij} = kmin_{\forall i' \in \{1,\ldots,k_i\}} dist\left(g_i^{i'}, alin(p_j, i, i')\right) \tag{6.11}$$

  suponiendo que *kmin* es una función que devuelve el *k*-ésimo menor valor de un conjunto de reales.

En relación con otras técnicas de reducción a subespacios lineales, las integrales proyectivas tienen la ventaja de conservar la relación de vecindad entre píxeles. Si p es una proyección, dos valores próximos $p[a]$ y $p[a+1]$ corresponden a regiones cercanas en las imágenes. Podemos decir que los puntos $p[a]$ y $p[a+1]$ son *vecinos* o *adyacentes*. Esto no ocurre en métodos como PCA, ICA o LDA, donde no existe ninguna relación especial entre los valores del vector proyectado. Los cuatro métodos propuestos de clasificación aprovechan de forma explícita esta cualidad, a través de un alineamiento previo de las señales. En condiciones realistas de uso, donde los reconocedores reciben la salida de un detector/localizador de caras (y no posiciones etiquetadas manualmente) esta ventaja puede resultar muy relevante.

Resultados comparativos de los métodos de clasificación

Para validar la bondad de los diferentes mecanismos de clasificación propuestos, hemos llevado a cabo algunas pruebas sobre conjuntos de datos concretos. En particular, manejamos las bases de caras ESSEX y GATECH, que permiten varias muestras por individuo. Usamos los grupos: "'faces94", "faces95" y "faces96" de ESSEX (ver la tabla 6.1) y "gatech". Para cada uno de ellos se aplican 7 clasificadores:

- **dnorm**: distancia normalizada;
- **dmed**: distancia a la media;

- **vmp**: vecino más proximo;
- **2vmp**, **3vmp**, **4vmp** y **5vmp**: k-vecinos más próximos, con $k = 2, 3, 4, 5$, respectivamente.

De cada imagen se obtiene la proyección vertical de la cara, incluyendo desde el pelo hasta la barbilla. Antes de calcular las proyecciones, las caras son rectificadas y extraídas a un tamaño estándar. El tamaño de las proyecciones usadas es de 40 puntos.

En la tabla 6.2 se exponen los resultados de este experimento para un escenario de identificación en conjunto cerrado. Nos centramos aquí en el ratio de identificación correcta, $P_I(1)$.

Conjunto	dnorm	dmed	vmp	2vmp	3vmp	4vmp	5vmp
faces94	90,3	95,3	**99,0**	98,2	97,2	96,1	94,1
faces95	79,4	78,5	**98,7**	96,7	93,2	85,6	76,5
faces96	60,0	72,7	**95,0**	91,4	85,4	79,5	73,6
gatech	49,6	65,3	**84,6**	78,1	68,4	65,0	59,3
Media	69,8 %	78,0 %	**94,3 %**	91,1 %	86,1 %	81,6 %	75,9 %

Tabla 6.2: *Resultados de la identificación con distintos métodos de clasificación. Todos los valores indican porcentajes de identificación correcta para una identificación en conjunto cerrado. Los tamaños de la galería y del conjunto de pruebas están dados en la tabla 6.1 y en el texto. Se señalan en negrita los mejores resultados para cada grupo.*

La figura 6.15a) muestra las curvas CMC para los diferentes métodos en la base "faces96". Las curvas ROC para un problema de verificación, usando esos mismos métodos, se pueden ver en la figura 6.15b).

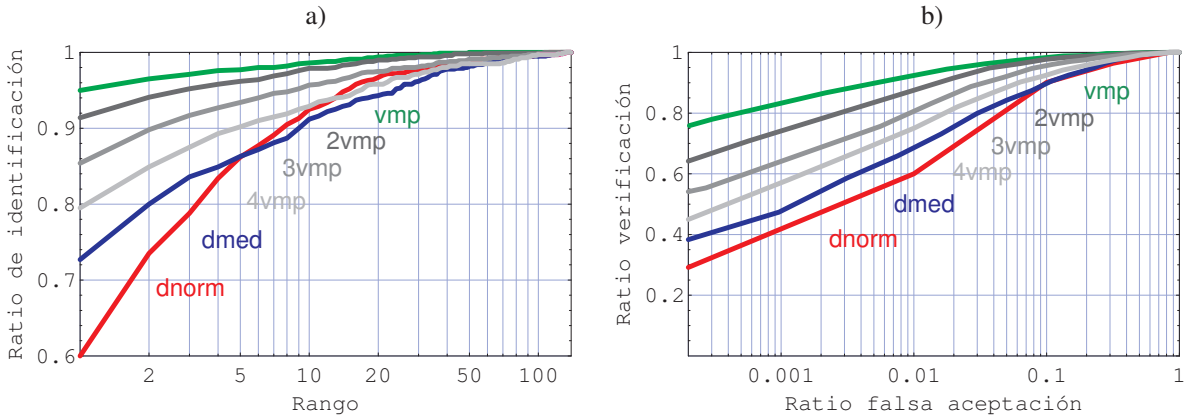


Figura 6.15: *Curvas CMC y ROC usando distintos métodos de clasificación. Se usa la base "faces96", que contiene 139 personas. La galería consta de 1358 imágenes y el conjunto de prueba 1337. a) Curvas CMC para 6 métodos de clasificación. b) Curvas ROC para los mismos métodos.*

Podemos extraer algunas conclusiones de este experimento:

1. La clasificación **mediante vecino más próximo** obtiene claramente los mejores resultados en todos los casos. El método de k-vecinos presenta una degradación progresiva a medida que aumenta el valor de k; de forma aproximada, el error de identificación se multiplica por k.

2. Los clasificadores basados en **proyecciones medias** se encuentran siempre por debajo de los demás. Este resultado es previsible, ya que la obtención de un modelo promedio supone reducir la variabilidad observada en la galería. También era previsible que el método de distancia normalizada (el que usa la varianza) resultara peor que el de distancia a la media. Curiosamente, sin embargo, a partir de rango 5 la situación se invierte, siendo preferible hacer uso de la varianza.

3. Las conclusiones se mantienen para el problema de **verificación** –figura 6.15b)–. La clasificación con vecino más próximo ofrece los mejores resultados, con un porcentaje de verificación del 92 % para un 1 % de falsas aceptaciones.

En definitiva, los resultados no dejan lugar a dudas sobre la técnica de clasificación más adecuada para el problema: la puntuación resultante para una prueba será la menor de las distancias con todas las muestras de la clase dada. Es más, la clasificación mediante vecino más próximo es la única alternativa posible cuando sólo disponemos de un ejemplo para algún sujeto de la galería. Al usar vecino más próximo, se mantiene la uniformidad de criterio en todos los casos, independientemente del número de muestras por persona.

6.3.3. Estudio de las regiones proyectadas

Uno de los aspectos más delicados en el diseño de un reconocedor mediante proyecciones consiste en seleccionar las regiones más *discriminantes* del rostro humano para ser proyectadas. Se debería garantizar que las señales usadas conservan la máxima información relevante de los individuos. En nuestro caso, la selección de las regiones se basará en observaciones experimentales: aplicamos el reconocimiento con proyecciones sobre diferentes regiones, y medimos los resultados obtenidos para cada alternativa. El porcentaje de identificación correcta será el criterio para decidir cuándo una opción es mejor que otra.

En definitiva, vamos a analizar a continuación distintas posibles regiones para la proyección vertical y la horizontal. Estas regiones están definidas en proporción al tamaño y posición de una cara según el modelo estándar que introducimos en el capítulo 2. La figura 6.16 muestra una representación del **sistema de coordenadas basado en la cara**, según las proporciones del modelo y una cara promedio. En adelante, los rangos de las regiones proyectadas se referirán a esta escala, tanto en el eje X como en el Y.

Proyecciones verticales

En los anteriores capítulos hemos analizado cómo las proyecciones verticales del rostro son muy interesantes en una discriminación cara/no cara. También aportan mucha información relevante en el problema de reconocimiento. Así, por ejemplo, en la propuesta de [190] se utiliza exclusivamente la proyección vertical de la cara, y las proyecciones rotadas en $\pm 7^o$, con el fin de compensar pequeñas diferencias de inclinación. Esto último es innecesario en nuestro caso, ya que las caras están rectificadas con los resultados de la localización.

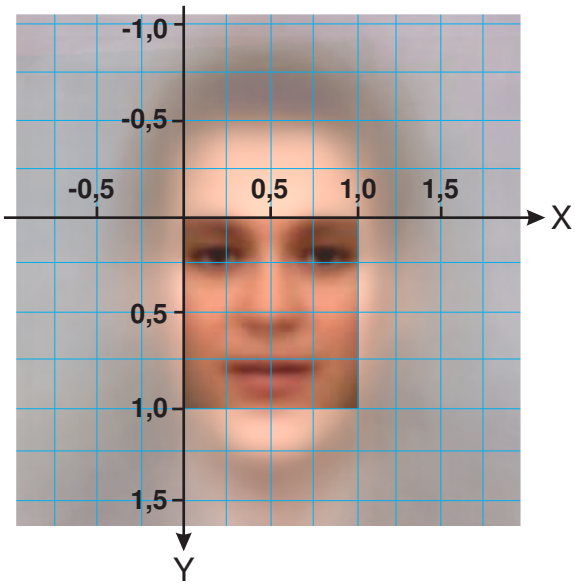


Figura 6.16: *Sistema de coordenadas centrado en la cara. Los valores de los ejes X e Y están en proporción al modelo estándar de la cara (ver el apartado 2.2.5 en la página 61). La cara media del fondo se ha obtenido de la base UMU.*

Pero el rendimiento del reconocedor puede variar sensiblemente en función de las zonas exactas seleccionadas. En el caso concreto de la proyección vertical, la región viene definida por tres parámetros: (1) el límite superior en el eje Y (incluir o no las cejas, la frente o parte del pelo); (2) el límite inferior en Y (llegar hasta la boca o la barbilla); y (3) el ancho en X de la parte proyectada (más o menos grande, suponiendo que está siempre centrado). Si nos fijamos en la figura 6.16, podemos ver que el margen superior puede ser un poco mayor que el inferior. En este sistema de coordenadas, los extremos de la cara en Y se encontrarían aproximadamente en el intervalo (-1; 1,5); más allá no tiene sentido aplicar las proyecciones, ya que se estaría tomando parte del fondo de la escena.

Para encontrar la selección más discriminante hemos realizado el siguiente experimento. Definimos 7 posiciones para el límite superior: (0,2; 0; -0,2; -0,4; -0,6; -0,8; -1); y otras 7 para el límite inferior: (0,9; 1; 1,1; 1,2; 1,3; 1,4; 1,5). Para cada combinación de límite inferior y superior, construimos un reconocedor utilizando proyecciones verticales sobre la región correspondiente y ancho 1 en X. Con el fin de mantener la cantidad de información, en todos los casos las señales usadas son de 40 puntos. Estos reconocedores se aplican a tres conjuntos: el grupo "fb" de la base FERET; la base GATECH; y el grupo "faces 94" de la base ESSEX. En todos los casos se usa clasificación de vecino más próximo, excepto en el último que se aplica distancia a la media[15].

Los resultados de los clasificadores para un problema de identificación en conjunto cerrado se muestran en las tablas 6.3, 6.4 y 6.5.

[15]El único propósito de esta elección es aumentar la diferencia entre la mejor y la peor elección. Recordemos el "problema de los tres osos" (ver la página 305).

Base de caras FERET, "fb"

Límite superior	Límite inferior 0,9	1,0	1,1	1,2	1,3	1,4	1,5
0,2	28,8	31,2	31,7	33,2	33,5	37,4	38,4
0,0	41,0	42,9	42,7	44,5	46,3	48,0	49,2
-0,2	47,6	50,3	52,3	53,8	57,3	58,5	58,8
-0,4	52,6	55,4	56,5	58,4	61,5	61,9	63,4
-0,6	57,9	58,7	60,9	62,5	64,7	65,9	68,5
-0,8	62,9	66,1	66,4	67,5	68,2	68,2	69,0
-1,0	65,6	67,1	67,1	67,8	68,8	68,2	**69,2**

Tabla 6.3: *Resultados de la identificación en conjunto cerrado con varias proyecciones verticales sobre la base FERET. La galería consta de 1196 imágenes y la prueba es el conjunto "fb" con 1195 imágenes. Todos los valores indican porcentajes de identificación correcta.*

Base de caras ESSEX, "faces94"

Límite superior	Límite inferior 0,9	1,0	1,1	1,2	1,3	1,4	1,5
0,2	78,5	81,6	82,6	83,7	83,4	83,1	84,5
0,0	87,4	89,1	90,3	89,1	88,7	89,6	89,7
-0,2	90,1	92,0	92,8	92,6	93,4	92,6	92,8
-0,4	91,3	93,4	93,6	93,7	94,2	93,9	93,8
-0,6	92,9	93,6	93,8	94,0	94,3	94,0	94,1
-0,8	92,5	94,2	94,8	95,0	95,6	95,3	95,5
-1,0	94,7	94,6	94,9	94,9	95,1	**95,7**	95,3

Tabla 6.4: *Resultados de la identificación en conjunto cerrado con varias proyecciones verticales sobre la base ESSEX. La galería consta de 1517 imágenes de 152 personas y la prueba es el conjunto "faces94" con 1517 imágenes. La clasificación es mediante distancia a la media. Todos los valores indican porcentajes de identificación correcta.*

Base de caras GATECH

Límite superior	Límite inferior 0,9	1,0	1,1	1,2	1,3	1,4	1,5
0,2	64,0	67,5	67,8	68,1	67,5	68,4	71,8
0,0	72,1	72,8	73,7	75,0	74,6	75,9	75,6
-0,2	79,0	79,0	80,0	80,6	81,8	80,6	80,9
-0,4	78,4	80,0	79,6	81,8	81,2	81,8	82,4
-0,6	80,3	80,3	81,8	81,2	83,4	84,9	82,8
-0,8	80,6	83,1	84,6	84,0	86,8	**87,1**	85,9
-1,0	84,0	85,6	85,6	82,8	86,2	86,5	84,9

Tabla 6.5: *Resultados de la identificación en conjunto cerrado con varias proyecciones verticales sobre la base GATECH. La galería consta de 347 imágenes de 50 personas y la prueba de 320 imágenes. La clasificación es mediante vecino más próximo. Todos los valores indican porcentajes de identificación correcta.*

Podemos extraer varias consecuencias de los resultados de esta prueba:

1. Tanto en el límite superior como en el inferior, se nota una clara tendencia general a mejorar la identificación conforme **aumenta el tamaño** de las regiones proyectadas. Es decir, cuantos más píxeles intervienen en la proyección, incluyendo la zona del pelo y la barbilla, el resultado es más fiable. Este hecho era previsible: al conservar más información, se puede distinguir mejor a unas personas de otras.

2. El aumento de las regiones proyectadas está limitado por el **tamaño de la cara**; no tiene sentido tomar unos márgenes que se salgan del rostro. Incluso, podría argumentarse la inconveniencia de incluir la zona del pelo, que puede cambiar de forma más fácil y arbitraria. De hecho, se puede apreciar que al acercarse a los extremos, la mejora de los ratios es más pequeña. En algunos casos el tamaño máximo no es el mejor, como en la base GATECH, donde la elección óptima es el intervalo (-0,8; 1,4).

3. Evidentemente, los porcentajes cambian mucho de un conjunto a otro, debido a los **diferentes factores** que intervienen: tamaño de la galería, número de imágenes por persona, variabilidad de las imágenes, etc. Es destacable el hecho de que el mejor ratio en la base GATECH, del 87,1 %, coincide por el presentado por sus creadores en [127], basado en modelos ocultos de Markov (HMM), aun cuando utilizan 10 imágenes de entrenamiento –en lugar de las 6,9 que tenemos nosotros– y sólo 5 de prueba.

En cuanto al ancho de las regiones proyectadas, hemos realizado otro conjunto de pruebas usando los tamaños: 0,4; 0,6; 0,8; 1; 1,2 y 1,4. Las regiones están siempre centradas en X. Por ejemplo, el ancho 0,6 corresponde a un margen en X entre 0,2 y 0,8. Los márgenes verticales para la proyección se fijan en el intervalo (-0,8; 1,4). Se han repetido las mismas pruebas sobre los tres conjuntos de imágenes, obteniendo los resultados de la tabla 6.6.

Conjunto usado	**Ancho proyectado**					
	0,4	**0,6**	**0,8**	**1,0**	**1,2**	**1,4**
FERET	**69,6**	68,7	68,3	68,2	67,1	67,0
ESSEX	94,4	94,7	95,4	95,3	**95,5**	95,1
GATECH	86,8	**88,4**	87,1	87,1	84,6	84,9

Tabla 6.6: *Resultados de la identificación variando el ancho de las proyecciones verticales. El alto de las proyecciones es el intervalo (-0,8; 1,4). Los conjuntos usados son descritos en las tablas 6.3, 6.4 y 6.5. Todos los valores indican porcentajes de identificación correcta.*

La conclusión más relevante es que el ancho utilizado para las proyecciones verticales, PV_{cara}, tiene una influencia menor en los resultados. Por ejemplo, para la base ESSEX la diferencia entre el mejor y el peor caso es de sólo 1 punto porcentual. Hay una ligera degradación para tamaños grandes, que es más evidente en la base FERET. Pero es difícil seleccionar un valor óptimo, teniendo en cuenta la disparidad de resultados. Posiblemente la elección más adecuada es utilizar valores intermedios, entre 0,6 y 1.

Proyecciones horizontales

A priori, las proyecciones horizontales de la cara resultan menos descriptivas que las verticales y, por lo tanto, parecen menos interesantes para el problema de reconocimiento. Sin embargo, estas proyecciones pueden aportar una información adicional suplementaria a la de PV_{cara}, ayudando así a mejorar los resultados del reconocimiento.

Existen diferentes cuestiones por decidir en cuanto a las proyecciones horizontales: (1) qué partes de la cara utilizar (ojos, nariz, boca, o toda la cara); (2) qué margen vertical en con-

creto se aplica; y (3) cuál debe ser la extensión horizontal de la proyección. Igual que antes, vamos a seleccionar unas cuantas alternativas para experimentar con ellas. En concreto, tenemos las siguientes regiones posibles, definidas por las posiciones en el eje Y que se proyectan (véase el sistema de coordenadas de la figura 6.16):

Ojos 1: entre 0 y 0,4. **Ojos 2**: entre 0,1 y 0,3. **Ojos 3**: entre 0,15 y 0,25.
Nariz: entre 0,4 y 0,7. **Boca**: entre 0,7 y 0,9. **Cara**: entre 0 y 1.
Ojos/nariz: entre 0 y 0,6. **Nariz/boca**: entre 0,4 y 1.

Para cada una de estas zonas se definen 7 posibles anchos, esto es, tamaños a lo largo del eje X: 1; 1,2; 1,4; 1,6; 1,8; 2 y 2,2. La extensión de la región en X está siempre centrada en la cara. Cada combinación región/ancho se utiliza para extraer proyecciones horizontales de tamaño 40, que son aplicadas en el reconocedor. Las pruebas son ejecutadas sobre las bases de caras FERET y ESSEX. Los resultados se encuentran en las tablas 6.7 y 6.8.

Base de caras FERET, "fb"

Extensión vertical	Ancho horizontal 1,0	1,2	1,4	1,6	1,8	2,0	2,2	Media
Ojos 1	43,5	45,6	44,5	45,5	48,4	**50,9**	52,2	47,2
Ojos 2	47,6	**47,6**	**48,6**	49,7	**51,4**	50,3	**52,9**	49,7
Ojos 3	**49,6**	**47,6**	48,2	**50,0**	51,2	50,5	52,7	**50,0**
Nariz	21,0	22,9	23,0	25,2	31,3	33,7	37,3	27,8
Boca	21,5	26,3	28,6	33,0	40,1	42,6	44,6	33,8
Cara	26,5	29,8	31,8	35,0	44,0	45,7	49,0	37,4
Ojos/nariz	35,2	36,8	37,4	39,4	42,8	46,7	49,6	41,1
Nariz/boca	24,2	25,8	28,4	31,7	37,5	40,9	45,7	33,5

Tabla 6.7: *Resultados de la identificación en conjunto cerrado con varias proyecciones horizontales sobre la base FERET. La galería consta de 1196 imágenes y la prueba es el conjunto "fb" con 1195 imágenes. Todos los valores indican porcentajes de identificación correcta.*

Base de caras ESSEX, "faces94"

Extensión vertical	Ancho horizontal 1,0	1,2	1,4	1,6	1,8	2,0	2,2	Media
Ojos 1	81,2	82,9	83,6	85,7	88,0	89,7	91,0	86,0
Ojos 2	75,8	79,1	80,5	82,0	87,0	88,2	89,9	83,2
Ojos 3	69,4	72,9	74,2	79,6	83,5	86,7	88,2	79,2
Nariz	78,3	80,9	81,6	82,9	87,0	88,6	90,9	84,3
Boca	59,2	67,7	70,2	76,4	84,5	88,3	91,1	76,8
Cara	81,8	84,3	86,1	**87,2**	88,9	90,9	**93,8**	87,6
Ojos/nariz	**83,5**	**85,4**	**86,2**	86,4	**89,7**	**91,4**	93,2	**88,0**
Nariz/boca	76,5	79,6	80,9	82,5	86,7	88,5	90,9	83,7

Tabla 6.8: *Resultados de la identificación en conjunto cerrado con varias proyecciones horizontales sobre la base ESSEX. La galería consta de 1517 imágenes de 152 personas y la prueba es el conjunto "faces94" con 1517 imágenes. La clasificación es mediante distancia a la media. Todos los valores indican porcentajes de identificación correcta.*

Los resultados, en este caso, son un poco contradictorios en algunos aspectos si comparamos los datos ofrecidos por ambos conjuntos. Pero podemos sacar algunas ideas generales:

1. De forma global, los porcentajes aumentan a medida que crece el **ancho horizontal** uti-

lizado. Esta conclusión es coherente con lo que vimos para las proyecciones verticales: conservar la mayor información posible es siempre preferible. Los mejores resultados se obtienen para ancho 2,2. Paradójicamente, ese tamaño está prácticamente en los límites exteriores de la cabeza (correspondería a los valores en X desde -0,6 hasta 1,6), por lo que no conviene ampliarlo.

2. Globalmente, los ratios obtenidos con las **proyecciones horizontales** son peores que con las verticales. Por ejemplo, para la base FERET el mejor resultado es del 52,9 %, mientras que para las verticales era del 69,2 %.

3. En cuanto a la decisión de cuál es la región más representativa, no existe unanimidad. En la base FERET, las tres **zonas de ojos** ocupan claramente las primeras posiciones. Después, a cierta distancia, se encuentran la región ojos/nariz y la cara. Sin embargo, en la base ESSEX esas dos alternativas están por encima de las proyecciones de los ojos; no obstante, aquí las diferencias son mucho más reducidas, encontrándose en un margen de tan solo 10 puntos.

4. Si nos fijamos en la **altura óptima** de las tres regiones de ojos, también surge una discrepancia entre ambas bases. Mientras que en FERET resulta más adecuado utilizar un región estrecha ("ojos 3"), en la base ESSEX ocurre todo lo contrario. Una posible causa para este fenómeno puede encontrarse en las imprecisiones de la localización. Recordemos que en la base FERET se usa el etiquetado manual, mientras que en ESSEX se parte de las posiciones del algoritmo de localización. Los errores de localización tienen mayores consecuencias si la región proyectada es más estrecha. Por lo tanto, una opción conservadora es utilizar un término intermedio.

La obtención de buenos porcentajes de identificación para las regiones de nariz y boca –como sucede en la tabla 6.8– no era previsible a priori. A pesar de ello, debemos indicar que los resultados de la base FERET son más significativos que los obtenidos con el conjunto ESSEX: primero, porque los márgenes en FERET son mucho mayores; y segundo, porque el volumen de individuos –y, por lo tanto, la fiabilidad de las pruebas– también son claramente superiores en FERET. En consecuencia, en adelante utilizaremos las regiones de los ojos con una altura de tamaño intermedio (la llamada "ojos 2") y anchura 2,2 en X.

Canales de color

En los anteriores capítulos hemos discutido la cuestión del color, en cuanto a la decisión de qué canales usar en las proyecciones. Nuevamente, volvemos a plantearnos la misma elección para los problemas de reconocimiento. En la tabla 6.9 se presentan distintos experimentos de reconocimiento utilizando los tres canales disponibles (R, G, B), y el valor de intensidad de los píxel. Se aplican sólo las proyecciones verticales de la cara, seleccionando los límites en el eje Y (-1; 1,5) y ancho 1. En este experimento se usan los conjuntos disponibles con imágenes

en color: los de la base ESSEX, y el conjunto GATECH. Además, se aplican para todos ellos los clasificadores de distancia a la media y vecino más próximo.

Canal usado	faces94 dme	faces94 vmp	faces95 dme	faces95 vmp	grimace dme	grimace vmp	gatech dme	gatech vmp	Media total
Azul	94,1	**99,4**	83,8	98,9	**57,5**	**85,4**	64,3	88,1	83,9
Verde	93,5	99,2	**86,6**	**99,5**	52,1	84,2	67,8	88,1	83,9
Rojo	**95,1**	**99,4**	85,7	99,2	55,1	**85,4**	**69,3**	87,8	**84,6**
Gris	94,5	99,2	83,4	99,2	50,9	83,6	69,0	**88,4**	83,5

Tabla 6.9: *Resultados de la identificación con proyecciones verticales y distintos canales de color. Se usa el ancho 1 y alturas (-1;1,5). Se proyectan los distintos canales: azul, verde, rojo y gris. Para cada uno, se manejan los cuatro conjuntos de imágenes:* faces94*,* faces95*,* grimace *(descritos en la tabla 6.1) y* GATECH*; con dos métodos de clasificación:* dmed *(distancia a la media),* vmp *(vecino más próximo). Todos los valores indican porcentajes de identificación correcta. Se señalan en negrita los mejores resultados por columna.*

Los porcentajes de acierto con distintos canales resultan **muy similares**. La mayor discrepancia se encuentra en la base GATECH, siendo la diferencia entre usar rojo o azul (para distancia a la media) de 5 puntos a favor del primero. De forma global, el canal rojo produce resultados ligeramente mejores. Los otros se encuentran por debajo, y muy próximos entre sí. Debemos aclarar que, aunque las imágenes de la base ESSEX están en color, la iluminación y la calidad cromática no son excesivamente elevadas. Algunas caras presentan un tono azulado o violeta, causados por el balance de blancos. Precisamente en la base GATECH, donde la definición del color es mucho más alta, se concluye que el canal rojo es el más discriminante, aunque sólo si nos fijamos en los valores de distancia a la media.

Tamaño de las proyecciones

El tamaño, o resolución, de las proyecciones es otro de los parámetros que pueden tener cierta influencia en la efectividad del método. Cuanto menores sean las señales usadas, más rápidos serán los procesos de identificación y verificación; pero al mismo tiempo se puede estar eliminando información relevante. Por contra, si las proyecciones son grandes se conserva más información, aunque con más nivel de detalle se reduce la capacidad de generalización.

Para analizar su influencia en el reconocimiento hemos realizado una serie de experimentos variando este factor para las proyecciones verticales y las horizontales. Los tamaños van entre 8 y 70 (medidos como el número de puntos de la señal 1D) para las PV_{cara}, y entre 8 y 78 para PH_{ojos}. Se utilizan aquí los conjuntos de mayor tamaño: "faces94" y "faces95" de ESSEX, y "fb" de FERET. En los dos primeros se aplica clasificación mediante distancia a la media y en el último vecino más próximo.

Los resultados para los porcentajes de identificación correcta se muestran en la figura 6.17. Obsérvese que los ratios se han expresado aquí en proporción al máximo alcanzado.

Destacamos algunos resultados del análisis de este experimento:

1. Lógicamente, los ratios de identificación no son muy grandes para **tamaños muy pequeños**, entre 8 y 14. Pero es interesante señalar la buena capacidad de reconocimiento que

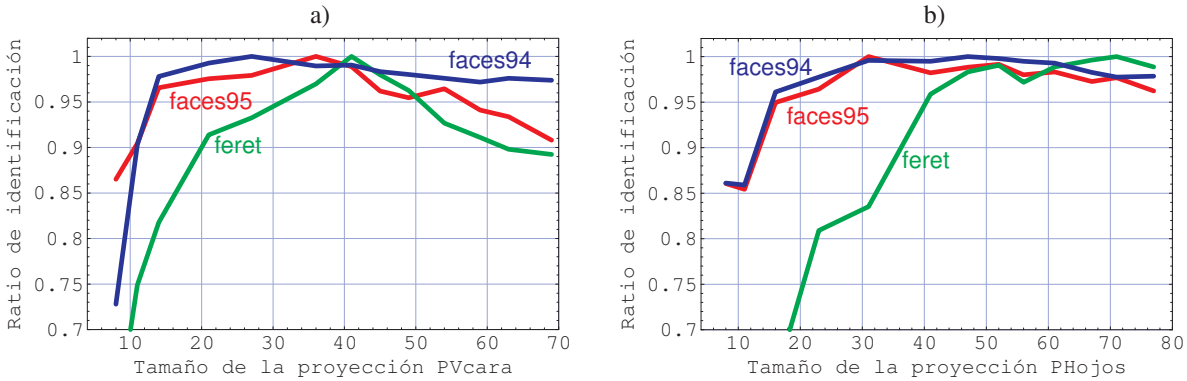


Figura 6.17: *Ratios de identificación relativos en función del tamaño de las proyecciones. Se usan los conjuntos: "faces94" y "faces95" de ESSEX, y "fb" de FERET. Los ratios son en proporción al máximo de cada curva. a) Ratios relativos en función del tamaño de la proyección vertical de la cara. b) En función del tamaño de la proyección horizontal de los ojos.*

muestran; incluso para señales de sólo 8 puntos se consiguen porcentajes muy interesantes. Por ejemplo, en términos absolutos los ratios de identificación para los grupos de ESSEX con tamaño 8 están sobre el 80 %, tanto para PV_{cara} como para PH_{ojos}.

2. Ambas gráficas presentan **picos de rendimiento** para tamaños en torno a 30-50 puntos. A partir de esos máximos, los ratios se mantienen o caen lentamente. El descenso es más evidente para PV_{cara}. Este hecho puede ser un reflejo de lo que ya hemos mencionado: al usar más información se mejora la capacidad de identificación, pero un nivel de detalle excesivo empeora la capacidad de generalización.

3. Los tamaños óptimos son ligeramente mayores para PH_{ojos} que para PV_{cara}. Es lógico pensar que en la proyección horizontal de los ojos se necesita más resolución para encontrar detalles identificativos de cada persona.

Por lo tanto, una elección adecuada podría consistir en fijar los tamaños en la zona "estable" próxima a los picos de máximos rendimiento, sobre los 40 puntos.

6.3.4. Combinación de proyecciones

En el anterior apartado hemos analizado de forma separada el uso de las proyecciones verticales y las horizontales en los problemas de reconocimiento facial. Pero un sistema de identificación o verificación biométrica puede incluir en general dos o más tipos de proyecciones, contribuyendo así a mejorar los resultados conseguidos.

Desde el punto de vista de la función de similitud, la combinación consiste en encontrar una forma de obtener las puntuaciones finales, s_{ij}, a partir de las puntuaciones (en nuestro caso distancias) asociadas a cada proyección, s_{ij}^{p}.

Supongamos que tenemos una proyección vertical de la cara y una horizontal de los ojos, que dan lugar a los valores s_{ij}^{pv} y s_{ij}^{ph}, respectivamente. El par de distancias entre cada prueba

y cada muestra de la galería se puede ver como un punto en un espacio 2D, $(s_{ij}^{pv}, s_{ij}^{ph})$. En la figura 6.18 se presenta un ejemplo de este espacio, para una base de tamaño reducido.

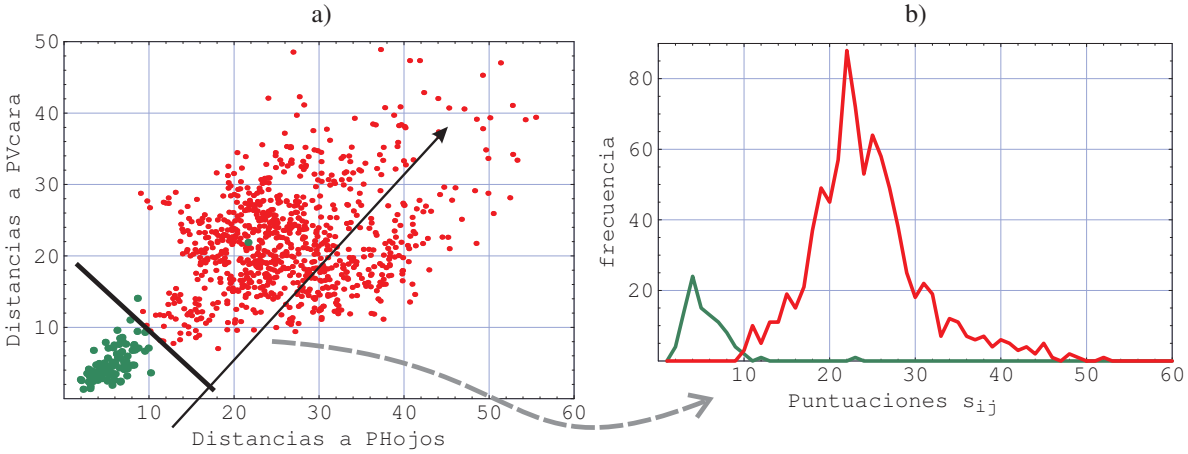


Figura 6.18: *Combinación de proyecciones en el proceso de reconocimiento. a) Distancias de la proyección PV_{cara} frente a PH_{ojos}, entre las pruebas y las caras de la galería. Las distancias para la identidad correcta se representan en verde y las restantes en rojo. b) Histograma de distancias resultantes de la combinación lineal entre ambas distancias. En verde las correctas y en rojo las incorrectas.*

Genéricamente, podemos expresar el mecanismo de combinación de puntuaciones como una función f tal que:

$$s_{ij} = f\left(s_{ij}^{pv}, s_{ij}^{ph}\right) \tag{6.12}$$

La forma más común de definir f es como una **combinación lineal de las puntuaciones** de entrada. De esta manera, si introducimos el parámetro α, entre 0 y 1, para expresar la ponderación de cada término, el resultado sería:

$$s_{ij} = (1-\alpha)\cdot s_{ij}^{pv} + \alpha \cdot s_{ij}^{ph} \tag{6.13}$$

En la figura 6.18b) se puede ver un caso donde se ha utilizado esta función. Podemos interpretar que α representa la pendiente de la recta –más exactamente, el seno del ángulo, entre 0 y 1– sobre la cual se proyectan los pares del espacio 2D.

El objetivo ahora es encontrar el valor del parámetro α que maximice los resultados del reconocimiento, tanto en identificación como en verificación. Normalmente, hacer uso de dos proyecciones supondrá una mejora significativa en el rendimiento alcanzado. Sin embargo, el valor óptimo de α puede variar según los conjuntos usados, el tamaño de la galería, o las proyecciones concretas aplicadas.

Idealmente se debería poder realizar un ajuste específico del α para cada galería. Pero esto parece inviable si la galería contiene una sola muestra por individuo. Por ello, no vamos a plantear un algoritmo para el ajuste automático de la ponderación, sino que nos limitamos a describir una serie de pruebas con el fin de estudiar de forma global el comportamiento de este parámetro.

Combinación en el problema de identificación

Observemos la gráfica de la figura 6.18b). En un problema de verificación los errores cometidos están en función del área de solapamiento entre ambas curvas. Pero en identificación cerrada las partes solapadas no necesariamente dan lugar a error, siempre que para todo j la menor distancia de s_{ij} sea la de la identidad correcta. Por lo tanto, el estudio analítico resulta más complejo. Por ello, vamos a analizar el comportamiento del ratio de identificación de manera experimental.

Más concretamente, se ha realizado una serie de pruebas sobre los grupos: "faces94" y "faces96" de ESSEX, "fb" de FERET, y la base GATECH. Para cada muestra, ya sea de la galería o de prueba, se calcula la proyección vertical de la cara, PV_{cara}, y la horizontal de los ojos, PH_{ojos}. Para la primera se usa el margen vertical (-0,8; 1,4) y ancho 0,8; y para la segunda margen (0,1; 0,3) y ancho 2,2. Con estas proyecciones aplicamos las distancias descritas en el apartado 6.3.2. En los conjuntos de la base ESSEX se usa distancia a la media[16], y en los restantes vecino más próximo. Estas medidas se combinan según se define en la ecuación 6.13 para obtener las distancias finales, s_{ij}; los valores del parámetro α se modifican progresivamente entre 0 y 1. Por último, estas puntuaciones se aplican a un problema de identificación en conjunto cerrado. Los resultados del experimento son las cuatro curvas de la figura 6.19, que expresan los ratios de identificación correcta en función de α.

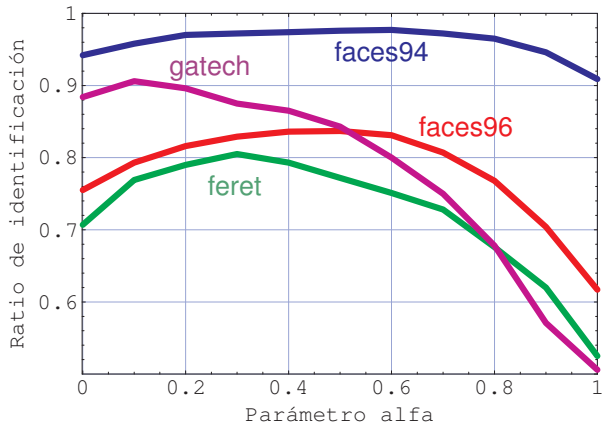


Figura 6.19: *Ratios de identificación combinados con diferentes ponderaciones. El eje horizontal corresponde al parámetro α de la ecuación 6.13. El vertical son los ratios de identificación correcta en cuatro conjuntos de prueba.*

Con el valor $\alpha = 0$ los porcentajes de identificación de la figura 6.19 corresponden a utilizar únicamente PV_{cara}, mientras que $\alpha = 1$ es equivalente a usar sólo PH_{ojos}. Hay varias cuestiones que podemos destacar:

1. En todos los casos, **combinar las distancias** produce mejores resultados que usarlas por separado. Por ejemplo, en FERET y en "faces96" se reduce el error en un 10 % respecto

[16]Como en las pruebas anteriores, el único motivo para usar esta clasificación es aumentar los márgenes de error, mejorando así la separación entre los casos mejores y peores, exclusivamente a efectos comparativos.

al obtenido con PV_{cara}. Esto ocurre incluso aunque una de las proyecciones sea mucho menos discriminante que la otra, como en el caso de GATECH: con PV_{cara} se consigue 88 % y con PH_{ojos} sólo el 50 %; aun así, el método combinado logra en el mejor caso una identificación del 91 %.

2. El **valor óptimo** de α no parece estar claramente definido, encontrándose normalmente entre 0,1 y 0,5. En cierto sentido, se puede decir que depende de los ratios conseguidos en los extremos. Si los errores usando PV_{cara} y PH_{ojos} son muy parecidos, como pasa con "faces94", el valor de α más adecuado estará próximo a 0,5. Sin embargo, si una proyección produce menos error que la otra, es lógico que se deba primar en la combinación. Esta situación sucede con FERET; con GATECH ocurre un caso más extremo, ya que el α óptimo es de sólo 0,1.

Para permitir apreciar mejor el efecto de la combinación, se representan en la figura 6.20 las curvas CMC que corresponden a usar sólo PV_{cara}, PH_{ojos} o realizar la combinación óptima. Podemos ver que la ganancia conseguida es uniforme a lo largo de todo el rango.

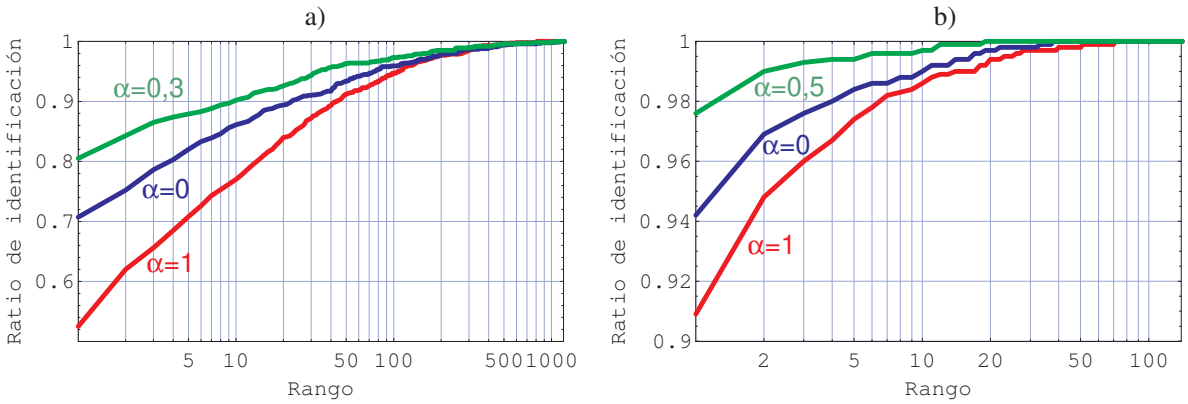


Figura 6.20: *Curvas CMC usando las proyecciones combinadas o por separado. Se muestran los resultados de identificación en función del rango usando sólo PV_{cara} (en azul), PH_{ojos} (en rojo) y combinando ambos (en verde). a) Usando el grupo "fb" de FERET. b) Usando el grupo "faces94" de ESSEX.*

Combinación en el problema de verificación

Las mismas pruebas del punto anterior se han repetido para un problema de verificación. Los conjuntos usados y la forma de obtener las puntuaciones son exactamente iguales. La única diferencia es que los resultados se aplican suponiendo ahora un escenario de verificación: dada una prueba y una identidad aducida, decidir si corresponden o no. Obsérvese que no existen *verdaderos impostores*, sino que el ratio de falsas aceptaciones está asociado a personas de la galería que reclaman una identidad diferente a la suya (ecuación 6.4).

Recordemos que el rendimiento de la verificación se expresa en una curva ROC, que representa el número de falsas aceptaciones frente al ratio de verificación correcta, variando el umbral de aceptación/rechazo. Para reducir la curva a un simple número, utilizamos el

ratio de error igual –es decir, el ajuste con igual número de falsas aceptaciones que de falsos rechazos–, aunque no necesariamente corresponde al modo de funcionamiento más interesante. Los resultados para los cuatro conjuntos de prueba variando la ponderación α se muestran en la figura 6.21. En estas gráficas el rendimiento será mejor cuanto más bajos sean los valores obtenidos.

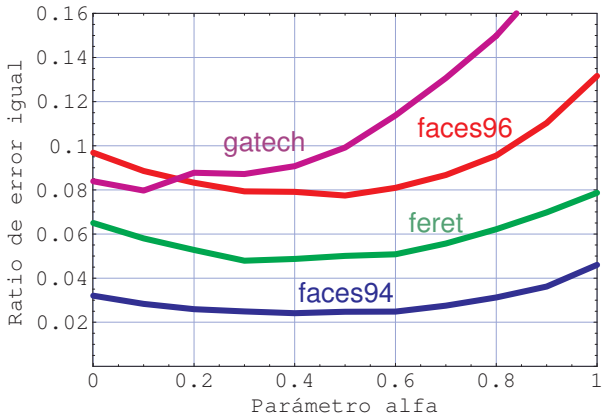


Figura 6.21: *Ratios de verificación combinados con diferentes ponderaciones. El eje horizontal corresponde al parámetro α de la ecuación 6.13. El vertical son los ratios de error igual en cuatro conjuntos de prueba. El caso $\alpha = 0$ es usando sólo PV_{cara}, y $\alpha = 1$ usando PH_{ojos}.*

El beneficio de combinar las distancias de las diferentes proyecciones se puede comprobar mejor en la figura 6.22, donde se representan las curvas ROC en dos conjuntos, usando proyecciones juntas o por separado.

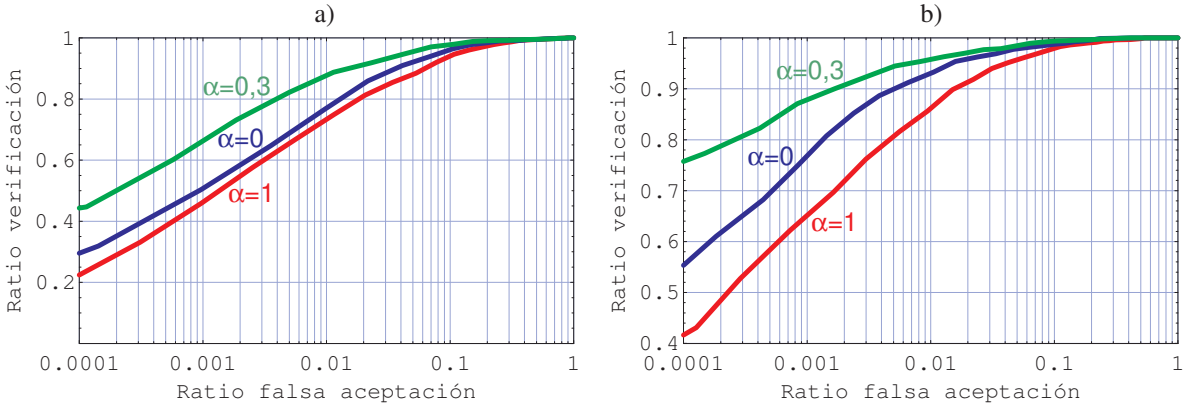


Figura 6.22: *Curvas ROC usando las proyecciones combinadas o por separado. Se muestran los ratios de verificación en función de las falsas aceptaciones usando sólo PV_{cara} (en azul), PH_{ojos} (en rojo) y combinando ambos (en verde). a) Usando el grupo "fb" de FERET. b) Usando el grupo "faces94" de ESSEX.*

Las conclusiones que podemos extraer son parecidas a las que describimos en el caso de la identificación en conjunto cerrado:

1. La **combinación** siempre produce **mejores resultados** que aplicar las proyecciones por separado. Esto se puede apreciar en la figura 6.21, pero es mucho más evidente en las

gráficas de la figura 6.22. Es más, podemos ver que la mejora es mucho mayor para ratios bajos de falsas aceptaciones. Así, por ejemplo, en la base FERET para un 0,1 % de falsos aceptados, se reduce el error de verificación en casi 20 puntos porcentuales.

2. El valor óptimo de α tampoco se puede establecer de antemano para todos los casos. No obstante, la zona de **funcionamiento óptimo** parece encontrarse en un amplio intervalo entre 0,2 y 0,6. La variación dentro del mismo resulta muy pequeña. El caso de la base GATECH es una excepción, ya que el peso óptimo es de 0,1 y el rendimiento decrece rápidamente a partir de ese valor. La razón es que en ese conjunto la proyección horizontal de los ojos es mucho menos discriminante que la vertical, como ya justificamos.

En definitiva, utilizar más de una proyección en el problema de reconocimiento es preferible en todos los casos. Manejar más proyecciones significa conservar más información de las imágenes disponibles. Sin embargo, los parámetros para una combinación óptima deben ser ajustados según las circunstancias de la aplicación. En general, estarán en función de la representatividad de cada proyección usada.

Normalización de las puntuaciones

En el problema de verificación se puede conseguir una mejora adicional introduciendo una adecuada normalización de las puntuaciones. El propósito de la normalización, como vimos en el apartado 6.1.1, es conseguir que las distancias para diferentes individuos de la galería se encuentren en rangos comparables.

Un método básico de normalización consiste en "dividir por el mínimo". Dada una prueba p_j, se obtienen en primer lugar las distancias con todas las muestras de la galería, s_{ij}. Las puntuaciones resultantes serían:

$$s'_{ij} = \frac{s_{ij}}{\text{mín}_{\forall i'}\, s_{i'j}} \tag{6.14}$$

La verificación se realizaría ahora utilizando los valores s'_{ij}. Por ejemplo, un umbral de aceptación/rechazo, τ, de 1 sería equivalente a aceptar sólo las pruebas para las cuales la menor distancia es con la clase aducida (esto es, $rango(p_j) = 1$). De esta forma, la decisión no sólo depende de la prueba y la clase indicada, sino de todas las demás clases de la galería.

Los experimentos del punto previo –cuyos resultados se presentan en las figuras 6.21 y 6.22– han sido repetidos aplicando normalización de las puntuaciones, con la fórmula de la ecuación 6.14. Los nuevos resultados de la verificación aparecen en la tabla 6.10. Se indican los ratios de error igual para los modos de combinación óptimos.

Se pueden ver dos curvas ROC obtenidas tras normalizar las distancias en la figura 6.23, comparadas con las disponibles antes de aplicar el proceso. Las gráficas corresponden a los mismos conjuntos de la figura 6.22.

Claramente, la normalización consigue mejorar los resultados de la verificación para todos los casos, como se puede ver en la tabla 6.10. Pero conviene profundizar más sobre algunos

Modo	faces94			faces96			feret			gatech		
	PV	*PH*	*Comb*	*PV*	*PH*	*Comb*	*PV*	*PH*	*Comb*	*PV*	*PH*	*Comb*
Sin norm.	3,2	4,6	2,4	9,7	13,2	7,7	6,5	7,9	4,8	8,4	19,5	8,0
Normaliz.	0,98	1,3	0,32	5,4	8,6	4,2	5,4	6,8	3,5	3,6	15,9	3,1

Tabla 6.10: *Resultados de la verificación con normalización de puntuaciones. Todas las medidas son ratios de error igual, en porcentaje (de 0 a 100). Para cada conjunto, se muestran los resultados usando sólo PV_{cara}, PH_{ojos} o la combinación óptima de ambas proyecciones. La primera fila son los ratios antes de normalizar (los mismos que los expuestos en la figura 6.21). La segunda fila son los obtenidos después de la normalización.*

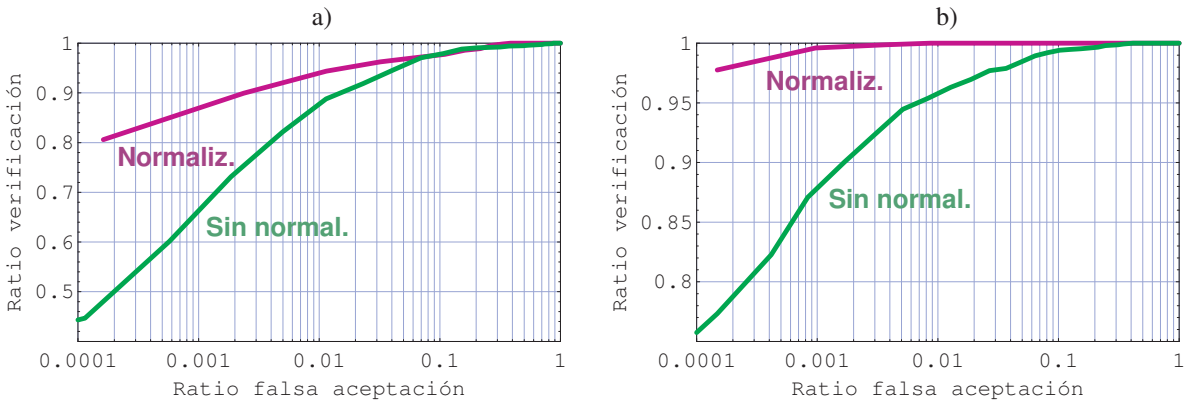


Figura 6.23: *Curvas ROC con o sin normalización de las puntuaciones. Se muestran los ratios de verificación en función de las falsas aceptaciones usando el método óptimo de combinación de proyecciones antes (en verde) y después (en violeta) de aplicar la normalización. a) Usando el grupo "fb" de FERET. b) Usando el grupo "faces94" de ESSEX.*

aspectos de este mecanismo:

1. La **reducción del error** parece ser más destacada para los grupos donde el error de partida es pequeño. Por ejemplo, en "faces94" el error del 2,4 % se reduce 7,5 veces; sin embargo, en los demás casos se divide típicamente el error entre 2 y 1,3 veces.

2. La figura 6.23 demuestra que la mejora dentro de un grupo es mucho más significativa para los valores **bajos del ratio de falsas aceptaciones**. A medida que disminuye este ratio, la diferencia entre normalizar o no es más acusada.

3. Si analizamos más detenidamente las gráficas de la figura 6.23, observamos que las curvas para los métodos normalizados se *detienen* antes de alcanzar cero falsas aceptaciones. En otras palabras, el **mínimo ratio de falsa aceptación** está delimitado en ambos casos. Esto tiene una explicación sencilla. Al estar normalizadas las distancias, todas ellas son mayores o iguales que 1, por lo que el mínimo valor posible de τ es 1. Por tanto, todas las pruebas, p_j, para las que la mínima distancia, s_{*j}, sea con una clase incorrecta, $id(g_*) \neq id(p_j)$ (es decir, $rango(p_j) > 1$), darán lugar siempre a falsas aceptaciones.

De la limitación detectada en el último punto se deriva un **problema de naturaleza operativa**. Supongamos que un verdadero impostor, p_j, quiere ser aceptado por el sistema. Le bastaría con encontrar la persona de la galería, e, que más se le parece –aunque realmente

no se parezca mucho, basta con que sea *la más parecida*–. En principio, la distancia s_{ej} sería la mínima, con lo que al dividir por ella su puntuación normalizada sería $s'_{ej} = 1$, y siempre sería aceptado. Este inconveniente no ocurre antes de la normalización: aunque s_{ej} sea el mínimo para todo j, si es mayor que el umbral τ se rechazaría la prueba.

Además, en el escenario de identificación en conjunto abierto la normalización nunca produciría rechazos, de manera que el ratio de falsa aceptación sería siempre 1.

Todo esto no contradice la bondad de las técnicas de normalización, sino que sugiere que en la práctica es necesario diseñar mecanismos más elaborados que la simple división por el mínimo. Es cierto que con ella se mejoran las curvas ROC obtenidas, ya que añade una información sobre el orden entre las puntuaciones. Pero de alguna manera también se debería tener en cuenta la distancia en términos "absolutos".

6.4. Resultados experimentales

Existen muchos factores y elementos de interés en la evaluación de un sistema de reconocimiento, como la robustez frente a la iluminación, el tamaño de la galería, el número de muestras por individuo, el tiempo transcurrido entre capturas, la resolución de las imágenes, etc. Al mismo tiempo, se deberían tener en cuenta los distintos escenarios o problemas de reconocimiento; y conviene manejar diferentes bases de caras para abarcar un espectro más amplio de condiciones de captura. Todo ello, a su vez, debe realizarse de forma comparativa, contrastando el método propuesto con otras alternativas existentes.

En consecuencia, es necesario delimitar cuidadosamente el número y el tipo de las pruebas a ejecutar. Para los experimentos de esta sección utilizaremos algunas de las bases de caras que ya hemos introducido previamente; en concreto, tenemos las bases ESSEX [82], FERET [52], GATECH [127], y ORL [159]. Para algunas de ellas es posible encontrar resultados publicados, que señalaremos en los puntos correspondientes. No obstante, muchas veces la información ofrecida es incompleta, bien porque no se detallan todas las condiciones de experimentación –por ejemplo, los métodos de detección y localización usados–, o porque sólo se informa de un simple número, el ratio de identificación correcta. Por ello, hemos implementado dos métodos estándar de reconocimiento, basados en comparación de patrones y autocaras. Todos ellos han sido integrados en un programa que permite ejecutar una batería de pruebas cambiando los distintos parámetros de entrada, que se muestra en la figura 6.24. Además disponemos de un reconocedor mediante modelos ocultos de Markov (HMM) accesible públicamente [35].

Lógicamente algunas bases de caras son más adecuadas para ciertos propósitos y otras lo son para fines diferentes. Por esta razón, los experimentos están organizados en función de las bases utilizadas en cada caso. Pero antes de presentar los resultados de los experimentos vamos a detallar algunos aspectos sobre el desarrollo de los mismos.

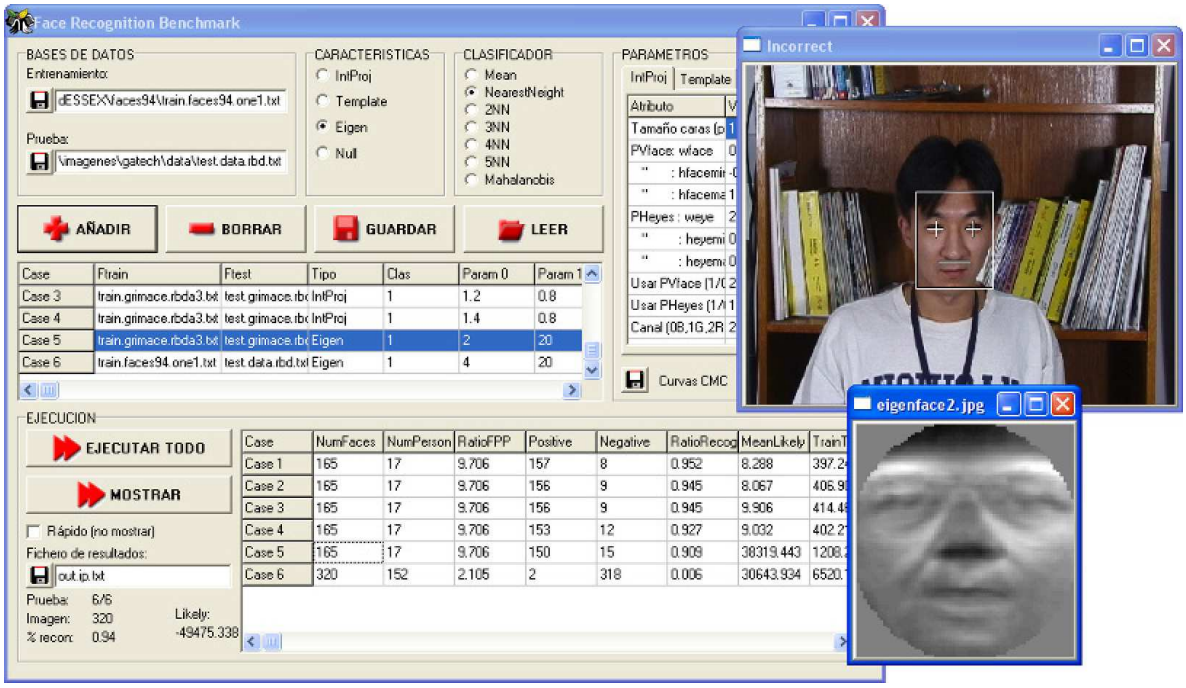


Figura 6.24: *Aplicación creada para la ejecución de las pruebas de reconocimiento. En la parte superior de la ventana, los datos que definen cada caso de prueba (base de datos, método, clasificador, parámetros, etc.). En medio, el conjunto de casos de prueba definidos. Abajo, los resultados de los casos. A la derecha, se pueden ver algunos resultados parciales.*

6.4.1. Descripción de las pruebas y métodos alternativos

Las medidas estándar de rendimiento en los diferentes problemas de reconocimiento han sido descritas extensamente en el apartado 6.1.2. En las pruebas nos centramos fundamentalmente en los escenarios de identificación en conjunto cerrado y verificación. Más específicamente, para cada experimento obtendremos los valores de los siguientes parámetros:

- ***$P_I(n)$***: ratio de identificación para rango n (ver la página 306). Los valores típicos que usaremos para n serán 1, 5 y 10.
- **eer**, **eern**: ratio de error igual en el problema de verificación (ver la página 342) antes y después de normalizar las puntuaciones. Típicamente los valores serán mejores tras la normalización, pero debemos recordar las limitaciones de esta técnica (sobre todo, en cuanto a su aplicación a identificación en conjunto abierto).
- **FA=1 %**, **FAn=1 %**: ratio de verificación, $P_V(\tau)$, cuando el ratio de falsas aceptaciones, $P_{FA}(\tau)$, es del 1 % (ver la página 307) antes y después de normalizar las puntuaciones.
- **Tiempo (ms)**: tiempo medio de ejecución, en milisegundos, en reconocer una imagen, incluyendo la lectura del fichero y la obtención de todas las puntuaciones. En uno de los apartados nos centraremos más específicamente en la cuestión de los tiempos de ejecución, estudiando también el coste de entrenamiento.

Acompañando a cada tabla incluimos una descripción de los datos utilizados para ese caso, con el fin de tenerlos más a mano. La información contiene: nombre de la base y del grupo dentro de la misma, número de personas e imágenes en la galería –se indica el número medio por persona (pp)–, número de imágenes de prueba, resolución de las imágenes (en píxeles), y la variación principal que aparece en el conjunto. A menos que se diga lo contrario, el conjunto de prueba es $\mathcal{P}_\mathcal{G}$, esto es, de personas en la galería.

Otros resultados de interés son las curvas CMC y ROC (ver el apartado 6.1.2) para identificación y verificación, respectivamente. En ambos casos usamos escala logarítmica en el eje horizontal, destacando así el comportamiento para valores bajos.

En cuanto a los métodos aplicados en las pruebas, disponemos de 3 alternativas al reconocimiento basado en proyecciones. Para algunas bases de caras presentaremos también resultados publicados por otros investigadores. Las técnicas disponibles son las siguientes:

IntProy - Reconocimiento mediante integrales proyectivas

Es el método propuesto y desarrollado en la sección 6.3. Como hemos visto, existen muchos parámetros ajustables que pueden tener cierta influencia en la efectividad del proceso. Los valores por omisión de los mismos han sido establecidos de antemano, a partir de las conclusiones descritas en la citada sección; y serán aplicados en todos los experimentos, a menos que se indique lo contrario. Estos valores son los siguientes:

- Proyección vertical de la cara. Extensión en X: (0,1; 0,9), extensión en Y: (-0,8; 1,4), tamaño: 40 puntos.
- Proyección horizontal de los ojos. Extensión en X: (-0,5; 1,5), extensión en Y: (0,1; 0,3), tamaño: 40 puntos.
- Canal usado (en caso de tener imágenes en color): rojo.
- Parámetro de combinación de las proyecciones: $\alpha = 0,4$.

Cuando dispongamos de más de un ejemplo por individuo, el mecanismo de clasificación aplicado será mediante vecino más próximo, al igual que para los restantes reconocedores.

TemMatch - Reconocimiento mediante comparación de patrones

La comparación de patrones es una operación básica en el procesamiento de imágenes, como ya hemos visto en los problemas de detección, localización y seguimiento. También ha sido aplicada en los problemas de reconocimiento [183, 18, 125], aunque más como una técnica básica para el contraste de resultados que como un método con alta fiabilidad. A pesar de ello, bajo ciertas circunstancias un simple algoritmo de este tipo puede conseguir mejores resultados que otras técnicas más avanzadas[17].

[17] Véase, por ejemplo, la comparativa de métodos en la última evaluación de FERET [52], o la comparación de varios métodos basados en subespacios en el capítulo 4 de [108].

El proceso de reconocimiento implementado es el siguiente. En primer lugar, las imágenes de cara se extraen a un tamaño fijo de 27×37 píxeles. Los límites de la cara, en relación al modelo estándar, son iguales que los aplicados en el reconocedor mediante proyecciones: (-0,8; 1,4) en vertical, y (-0,5; 1,5) en horizontal. Después se recorta una forma elíptica en la imagen extraída, para evitar la aparición de píxeles del fondo. En la figura 6.25 se pueden ver algunos ejemplos obtenidos de esta forma.

Figura 6.25: *Ejemplos de caras extraídas de la base FERET [52], para el reconocimiento mediante comparación de patrones. El tamaño de las imágenes es de 27×37 píxeles. Las esquinas se eliminan usando una máscara elíptica.*

Las puntuaciones, s_{ij}, asociadas a cada muestra g_i con cada prueba p_j, se calculan con una suma de diferencias al cuadrado entre ambas imágenes. En el caso de entrada a color, se usa el valor de brillo. El resultado se normaliza dividiendo por la suma de los píxeles de p_j al cuadrado. La clasificación se realiza usando vecino más próximo.

Todos los parámetros y modos de operación del proceso han sido ajustados mediante prueba y error, buscando el funcionamiento más adecuado para el mayor número de casos. Por ejemplo, aunque se podría aplicar una medida de correlación en el paso de comparación de patrones, en la práctica la suma de diferencias al cuadrado suele ofrecer mejores resultados.

EigenFace - Reconocimiento mediante autocaras

La técnica de reconocimiento con autocaras es una de las usadas más habitualmente como método base de comparación. Se trata del mecanismo más intuitivo de reducción a subespacios lineales, donde las imágenes de la base corresponden a los vectores propios del conjunto de entrenamiento. La idea de las autocaras fue propuesta por Turk y Pentland, en [183]. Desde entonces se han desarrollado muchas variaciones y extensiones de la técnica.

En nuestro caso, las imágenes de cara usadas en el proceso son las mismas que las extraídas para el reconocedor basado en comparación de patrones. A partir de las imágenes de la galería, se calculan los valores y vectores singulares mediante SVD, que corresponden a las autocaras principales. Se pueden ver algunas de ellas en la imagen 6.26, para la base ESSEX.

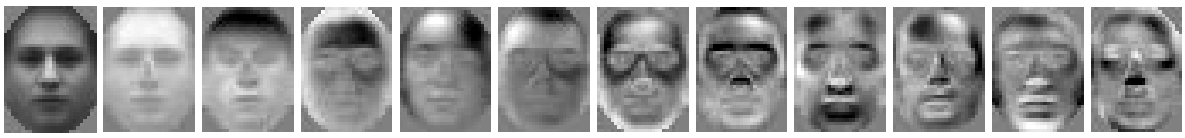


Figura 6.26: *Ejemplo de autocaras para la base ESSEX. El tamaño de las imágenes es de 27×37 píxeles. Las esquinas se eliminan usando una máscara elíptica. De izquierda a derecha, las 11 principales autocaras resultantes para el grupo "faces94" de la base ESSEX. La imagen de la izquierda corresponde a la cara media.*

Tanto las muestras de la galería como las de prueba son proyectadas en el autoespacio, dando lugar a vectores de reducida dimensión. La diferencia entre dos muestras se calcula con una simple distancia euclídea en el espacio de las autocaras.

Normalmente, el rendimiento de este tipo de reconocedores aumenta con el tamaño de la autobase, produciéndose una estabilización a partir de 20 ó 30. Para tamaños mucho mayores, es posible que ocurra un ligero empeoramiento de los resultados. Por ello, hemos establecido a 30 el número de autocaras usadas en los experimentos.

HMM - Reconocimiento mediante modelos ocultos de Markov

Como vimos en el repaso del estado del arte, existen muchos trabajos que aplican HMM en los problemas de reconocimiento de personas. En particular, se trata de una implementación del método propuesto por Nefian y Hayes [126, 127], realizada por sus mismos autores. El programa se incluía como una aplicación de ejemplo en las librerías de Intel OpenCV [35] hasta la beta 3, pero fue eliminado posteriormente (debido a que no cumplía el requisito de ser multiplataforma). La figura 6.27 muestra el programa en funcionamiento.

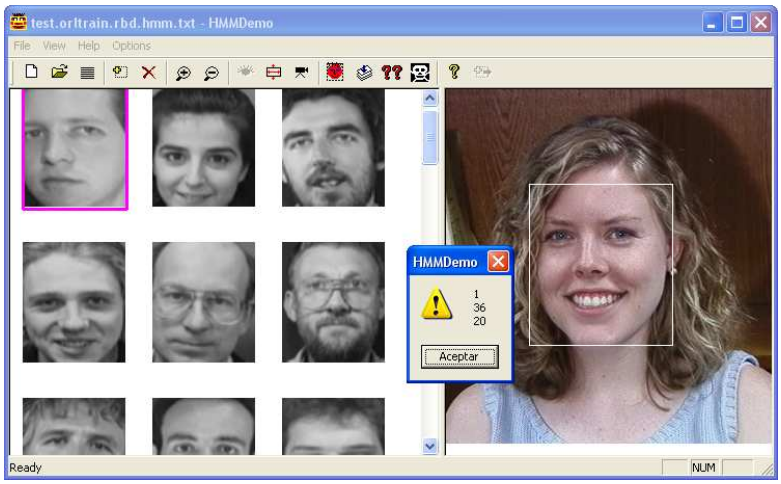


Figura 6.27: *Ejemplo de ejecución del reconocedor mediante HMM, disponible en [35]. El programa implementa un escenario de identificación en conjunto cerrado. A la izquierda, la galería de personas conocidas por el sistema. A la derecha, la nueva imagen a ser reconocida. Al pulsar en "reconocer" se calculan las tres identidades más probables para esa cara.*

La forma de aplicar los modelos HMM en el reconocimiento es tal y como se describe en [126]. De forma resumida, existen 5 estados, asociados a las zonas de frente, ojos, nariz, boca y barbilla; cada uno tiene entre 3 y 6 subestados ocultos; y las observaciones de entrada se calculan a través de la DCT sobre cada zona de las imágenes, que son escaneadas de arriba abajo y de izquierda a derecha.

Además del uso interactivo, el programa permite el procesamiento por lotes de un conjunto de imágenes. Desafortunadamente, aunque el método suele funcionar bien, la aplicación no aporta mucha información sobre los resultados del reconocimiento. Sólo se resuelve el problema de identificación en conjunto cerrado, y el único valor proporcionado es el ratio de

identificación correcta, $P_I(1)$. Además, la forma de la entrada hace difícil su ejecución cuando el número de individuos es muy grande, por lo que no será utilizado para la base FERET.

6.4.2. Resultados sobre la base ESSEX

Ya hemos ido presentando las principales peculiaridades de este conjunto de imágenes durante el desarrollo del método de reconocimiento (se puede consultar, por ejemplo, la tabla 6.1 en la página 326). Recordemos que en esta base las posiciones de las caras se obtienen automáticamente con la aplicación de los algoritmos de detección y localización[18], siendo descartadas algunas imágenes –una mínima parte– en las que no se consigue detectar la cara.

No se encuentran muchos trabajos que ofrezcan resultados comparativos con esta base. No obstante, su utilización sigue siendo interesante por varios motivos. Por un lado, la organización de los grupos permite realizar distintos tipos de experimentos –como veremos a continuación–; por otro lado, las condiciones de captura se aproximan a una aplicación típica, con cámaras de calidad media y resoluciones no muy altas.

Resultados individuales en cada grupo

En primer lugar, mostramos los resultados conseguidos por los cuatro métodos alternativos sobre cada uno de los cuatro grupos de la base ESSEX por separado. En este primer experimento se toman la mitad de las muestras de cada individuo para entrenamiento y la otra mitad para prueba. Los valores obtenidos se pueden consultar en la tabla 6.11.

En la figura 6.28 se muestran las curvas CMC y ROC de los dos últimos grupos, en los cuales existen mayores márgenes entre las diversas técnicas. La figura 6.28b) se centra en la diferencia entre normalizar las puntuaciones o no hacerlo, en el problema de verificación.

Debemos hacer algunas valoraciones en relación a estos resultados:

1. De forma global, los ratios alcanzados para los dos primeros grupos están muy próximos al **reconocimiento perfecto**. Se podría decir que los errores se encuentran dentro de los márgenes de precisión de la medida. Por ejemplo, en "faces94" todos los errores de identificación obtenidos para $P_I(1)$ corresponden en términos absolutos a entre 2 y 5 imágenes, de las 1517 disponibles.

2. Sí que existen más diferencias entre las técnicas si nos fijamos en los **resultados de la verificación**, y más concretamente en "faces95". En la configuración ideal, usando normalización de puntuaciones, el método basado en proyecciones destaca con claridad, siendo capaz de ofrecer un ratio de error igual del 0,1 %, es decir, verifica correctamente al 99,9 % de las personas, aceptando únicamente a 1 de cada 1000 impostores. Esto significa que IntProy produce una mayor separación entre las puntuaciones según se trate del mismo o de distintos individuos.

[18]Para las pruebas de este apartado, en el grupo "faces96" se aplica la detección basada en redes neuronales cuando falla el método combinado Haar+IP, con el objetivo de perder el menor número posible de caras. Esto explica que el tamaño de ese grupo en la tabla 6.11 sea algo mayor que el indicado previamente en la tabla 6.1.

Base caras	Grupo	Nº personas	Img. galería	Img. prueba	Resolución	Variación
ESSEX	faces94	152	1517 (10pp)	1517 (10pp)	180 × 200	Hablando
ESSEX	faces95	67	651 (9,7pp)	639 (9,5pp)	180 × 200	Posición y luz
ESSEX	faces96	150	1488 (9,9pp)	1482 (9,9pp)	196 × 196	Fondo y posic.
ESSEX	grimace	17	167 (9,8pp)	165 (9,7pp)	180 × 200	Expresión

		Identificación			Verificación				Tiempo
Grupo	Reconocedor	P_I(1)	P_I(5)	P_I(10)	eer	FA=1 %	eern	FAn=1 %	(ms)
faces94	**IntProy**	99,6	100	100	1,0	99,0	0,2	**100**	**15,2**
	TemMatch	**99,8**	100	100	1,2	98,8	**0,1**	99,9	44,3
	EigenFace	99,7	100	100	**0,9**	**99,2**	0,3	99,7	17,4
	HMM	99,6	–	–	–	–	–	–	531,6
faces95	**IntProy**	**99,5**	**100**	100	**1,9**	**97,2**	**0,1**	**100**	8,3
	TemMatch	**99,5**	**100**	100	2,1	96,9	0,5	99,6	13,2
	EigenFace	98,7	99,8	100	2,1	96,5	1,2	98,8	**6,1**
	HMM	98,8	–	–	–	–	–	–	197,4
faces96	**IntProy**	**95,6**	**97,8**	**98,7**	**4,6**	**90,9**	**1,0**	**99**	**14,2**
	TemMatch	94,8	97,3	98,5	7,2	87,0	4,3	95,1	34,6
	EigenFace	93,6	96,6	97,9	5,1	89,3	5,8	93,8	15,7
	HMM	95,2	–	–	–	–	–	–	439,3
grimace	**IntProy**	93,3	**98,1**	**99,3**	**7,6**	**84,9**	**2,7**	**94,1**	3,7
	TemMatch	92,1	**98,1**	98,7	12,2	75,7	5,5	92,4	4,9
	EigenFace	**93,9**	97,5	98,7	12,2	74,8	5,2	94,0	**3,0**
	HMM	91,6	–	–	–	–	–	–	67,5

Tabla 6.11: *Resultados del reconocimiento sobre distintos grupos de la base ESSEX. Se muestran ratios de identificación y de verificación para distintos puntos de las curvas CMC y ROC. Todos los valores (a excepción de los tiempos) indican porcentajes.*

3. Los dos primeros grupos podrían corresponder a una aplicación típica de **acceso a un edificio**, con unos pocos cientos de usuarios autorizados. El sistema de adquisición sería una cámara de bajo coste situada en un punto fijo, por ejemplo, un video-portero. Por lo tanto, la primera conclusión es que cuando las condiciones de captura están más o menos controladas, el sistema de reconocimiento es capaz de funcionar de forma óptima, mostrando una gran robustez frente a cambios de expresión facial y pequeñas modificaciones de posición e iluminación. Cualquiera de los métodos estudiados es viable en estas situaciones, pero IntProy es el más adecuado para verificación.

4. El grupo "faces96" es más próximo a un escenario donde el usuario es **capturado de forma *casual***, ocupando el rostro una pequeña fracción de las imágenes. Este caso pone en juego la robustez frente a grandes cambios de pose, del fondo, y reducida calidad de entrada. De hecho, la precisión del localizador de caras tiene un papel importante en los resultados, a diferencia de los grupos anteriores. Aproximadamente 22 de los errores en las 1482 imágenes de prueba (un 1,5 %) se pueden achacar a fallos de localización.

   Precisamente, en estas circunstancias, la superioridad de las integrales proyectivas se hace más patente. Como se ve en la curva CMC de la figura 6.28a), el método IntProy mantiene sobre 1 punto de ventaja respecto a las otras alternativas hasta rango 4. Des-

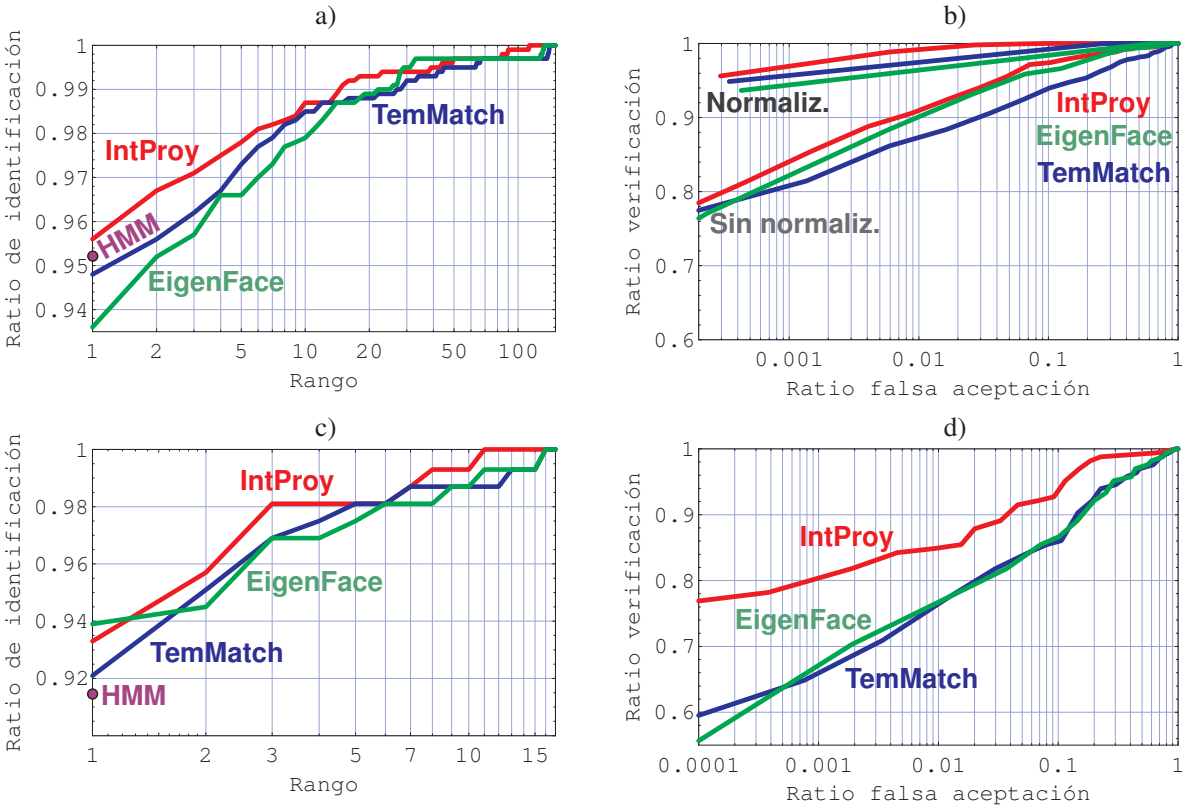


Figura 6.28: *Curvas CMC y ROC sobre los grupos "faces96" y "grimace" de la base de caras ESSEX, para los métodos de reconocimiento basados en proyecciones (en rojo), comparación de patrones (en azul), y autocaras (en verde). Se incluye el ratio de identificación correcta para HMM (en violeta). La descripción de los casos se encuentra en la tabla 6.11. a) Curvas CMC sobre el grupo "faces96". b) Curvas ROC sobre el grupo "faces96". Las de la parte superior corresponden a las puntuaciones normalizadas y las de la parte inferior a las no normalizadas. c) Curvas CMC sobre el grupo "grimace". d) Curvas ROC sobre el grupo "grimace", antes de normalizar las puntuaciones.*

pués, como cabía esperar, las diferencias disminuyen. También en el caso de la verificación las proyecciones obtienen los mejores resultados, y para todos los modos de funcionamiento.

5. En la figura 6.28b) se ve el gran beneficio que supone **normalizar las puntuaciones**, algo que ocurre en todos los métodos. Esto es debido al elevado número de pruebas con rango 1. Para un 0,1 % de falsas aceptaciones, se mejoran los ratios de verificación en unos 10 puntos porcentuales. Sin embargo, esta técnica tiene dificultades para bajar del 0,02 % de falsas aceptaciones, por las limitaciones que ya hemos discutido.

6. El grupo "grimace" hace énfasis en las **variaciones muy exageradas de la expresión**. Además, las condiciones de iluminación son ciertamente pobres, y todas las caras aparecen muy oscuras. Esto conduce a peores ratios, en general. En identificación, las autocaras logran mejores resultados. Pero en la gráfica de la figura 6.28c) se puede comprobar que para rango mayor que 1, IntProy consigue superar a los demás. La diferencia es mucho más clara en la curva ROC de la figura 6.28d). Mientras que EigenFace y Tem-

Match van a la par, el reconocedor basado en integrales proyectivas consigue hasta 20 puntos de mejora en los ratios de verificación.

Pruebas de identificación en conjunto abierto

Este escenario presenta una exigencia adicional a los anteriores, ya que los impostores son reconocidos en función de la menor distancia con todas las muestras de la galería. Es decir, se toma el mínimo entre un conjunto grande de distancias, pero ese mínimo debe ser lo suficientemente grande como para detectar que el individuo no está en $\mathcal{G}$. De esta forma, cuanto mayor sea el tamaño de la galería más probabilidad habrá de que el impostor se parezca a alguna persona de la misma, haciendo difícil establecer un buen umbral de aceptación/rechazo. Además, como es evidentemente, no tiene sentido aplicar el método de normalización que hemos usado para verificación.

Para una correcta evaluación de la identificación en conjunto abierto se necesitan tanto muestras de personas en la galería, $\mathcal{P}_{\mathcal{G}}$, como de verdaderos impostores, $\mathcal{P}_{\mathcal{N}}$. En consecuencia, en este segundo experimento tomamos las imágenes de "faces96" para la galería $\mathcal{G}$ y para $\mathcal{P}_{\mathcal{G}}$, y los 152 individuos de "faces94" como verdaderos impostores de $\mathcal{P}_{\mathcal{N}}$.

Las medidas de rendimiento en este escenario (página 308 y sucesivas) son: el ratio de detección e identificación para cierto umbral y rango, $P_{DI}(\tau, n)$, y el ratio de falsas aceptaciones, $P_{FA}(\tau)$. Si tomamos el mismo umbral τ para ambos, y centrándonos en el caso de rango $n = 1$, podemos expresar P_{DI} en función de P_{FA}. En la figura 6.29 se muestran las gráficas obtenidas para los tres métodos de reconocimiento con los datos indicados, y que se detallan en la tabla 6.12. La técnica basada en HMM no se ha podido incluir en este experimento.

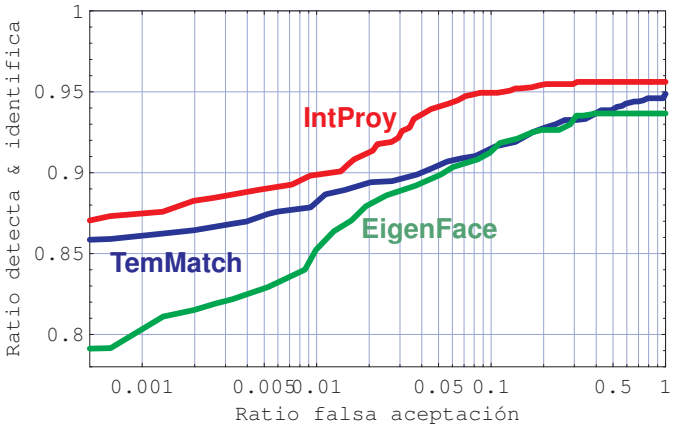


Figura 6.29: *Curvas ROC para identificación en conjunto abierto sobre la base ESSEX, para los métodos de reconocimiento basados en proyecciones (en rojo), comparación de patrones (en azul), y autocaras (en verde). La descripción de los casos se encuentra en la tabla 6.12. La galería es el grupo "faces96" y los impostores son de "faces94".*

Comentamos brevemente los resultados de esta prueba:

1. Lógicamente, **todos los ratios disminuyen** en relación al escenario de verificación, donde el usuario aporta su identidad al sistema. Así, por ejemplo, para un 1 % de falsas

Conjunto	Grupo	Nº personas	Nº imágenes	Resolución	Variación
$\mathcal{G}$: galería	faces96	150	1488 (9,9pp)	196×196	Fondo y posic.
$\mathcal{P}_{\mathcal{G}}$: pruebas	faces96	150	1482 (9,9pp)	196×196	Fondo y posic.
$\mathcal{P}_{\mathcal{N}}$: impostores	faces94	152	1517 (10pp)	180×200	Hablando

Reconocedor	Ratios de detección e identificación, $P_{DI}(\tau,1)$ *FA*=0,1 %	*FA*=0,5 %	*FA*=1 %	*FA*=5 %	eer	Tiempo (ms)
IntProy	**87,5**	**89,0**	**89,9**	**94,1**	**5,7**	16,4
TemMatch	86,1	87,4	88,2	90,4	8,8	34,5
EigenFace	80,2	82,8	85,3	89,8	9,0	**15,3**

Tabla 6.12: *Resultados de la identificación en conjunto abierto sobre la base ESSEX, usando los tres métodos de reconocimiento. Arriba se indican las imágenes usadas para la galería y para las pruebas. Abajo se muestran diversos ratios de detección e identificación para algunos puntos de la curva ROC, según el ratio de falsas aceptaciones (FA). También aparece el ratio de error igual (eer) y el tiempo de ejecución por imagen.*

aceptaciones, pasamos en IntProy del 99 % de verificación al 90 % de detección e identificación. Aun así, los valores alcanzados demuestran la viabilidad de utilizar integrales proyectivas, al menos para un tamaño de galería en torno al centenar de individuos.

2. Las **proyecciones superan** ampliamente a los otros dos métodos **basados en apariencia**. La mayor o menor ventaja depende del punto concreto en la curva ROC. Esto es debido a la evolución distinta de las tres curvas. La de TemMatch crece de forma regular –recordemos que se usa una escala logarítmica– mientras que las otras se estabilizan sobre 0,2 en el eje horizontal. Esto significa que la zona de solapamiento entre distancias correctas e incorrectas está más restringida en esos dos métodos, mientras que en TemMatch se extiende en un intervalo más amplio. Es decir, no sólo es importante el porcentaje de solapamiento, sino también la extensión de la parte solapada. Los resultados de IntProy indican que esa región se limita a los rangos de entre 1 % y 7 % falsas aceptaciones, ya que el crecimiento fuera de la misma es muy lento.

3. Aunque las **autocaras** parten de los peores resultados, la curva crece más rápidamente y llega a superar a TemMatch. El inconveniente es que el punto donde ambas se igualan –aproximadamente en el 10 % de falsas aceptaciones– es inadecuado para muchas aplicaciones, por lo excesivo del error.

Se puede decir que el problema de identificación en conjunto abierto aporta una funcionalidad extra a los otros dos escenarios. Respecto a la verificación, elimina la necesidad de pedir la identidad al usuario. Respecto a la identificación cerrada, detecta cuándo la persona no pertenece a la galería. Por ello, resulta más difícil obtener altos porcentajes, como ha quedado comprobado en las pruebas. No obstante, en comparación con los otros métodos, las proyecciones siguen siendo más descriptivas de la información que caracteriza a los individuos.

Variación del número de imágenes por persona en la galería

Una de las grandes cuestiones de interés en el reconocimiento biométrico de personas es analizar cómo influye en el rendimiento el número de muestras por individuo. Evidentemente, cuantas más imágenes tengamos de un sujeto podemos esperar mejores resultados. En la actualidad, se tiende a reducir al mínimo este valor. Esto se justifica, fundamentalmente, en la dificultad de conseguir más muestras. Sin embargo, si el sistema maneja una entrada de vídeo, esta limitación se puede superar con relativa facilidad.

Para este ensayo manejamos el grupo "faces96", por ser el que ofrece mayores ratios de error. Por ejemplo, en "faces94" los porcentajes de identificación usando entre 1 y 19 imágenes por persona están en el intervalo 94-100 %, lo cual deja poco margen para el análisis.

De las 20 imágenes disponibles para cada una de las 150 personas –realmente, algunas imágenes menos, debido a las caras no detectadas– se toman k para la galería y $20 - k$ para la prueba, seleccionadas de forma aleatoria. El valor de k varía desde 1 hasta 19. Para cada elección de k, se entrenan los reconocedores de los tres métodos y se aplican sobre los conjuntos de prueba correspondientes. De cada ejecución se mide el ratio de identificación correcta ($P_I(1)$) y el ratio de verificación cuando el número de falsas aceptaciones es del 1 % ($P_V(\tau)$ con $P_{FA}(\tau) = 0,01$). La figura 6.30 contiene las gráficas de ambos valores en función de k.

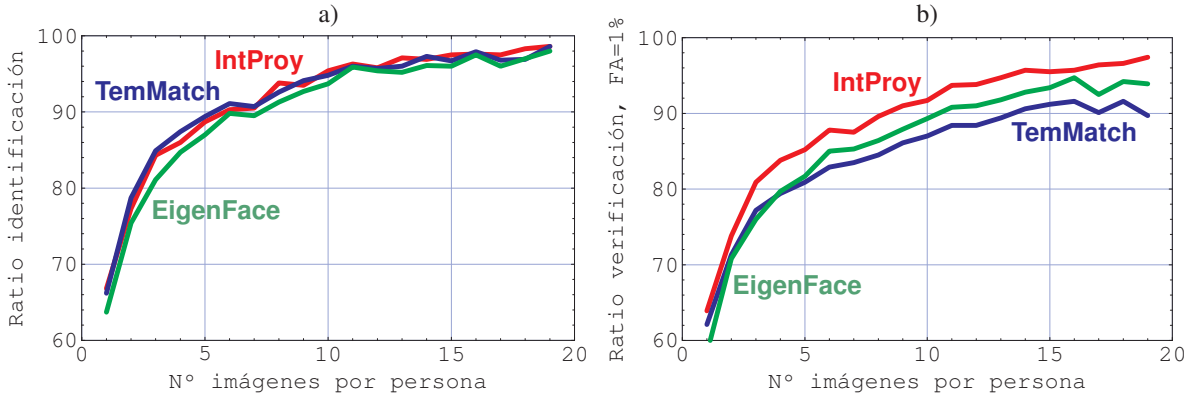


Figura 6.30: *Resultados del reconocimiento en función del número de imágenes por persona sobre el grupo "faces96" de la base ESSEX, para los métodos de reconocimiento basados en proyecciones (en rojo), comparación de patrones (en azul), y autocaras (en verde). a) Ratios de identificación en función de las muestras por persona. b) Ratios de verificación en función de las muestras por persona.*

Nuevamente, discutimos las conclusiones más relevantes punto por punto:

1. El comportamiento de las curvas de la figura 6.30 se acerca bastante a un **crecimiento logarítmico**. Para valores bajos, los porcentajes aumentan rápidamente, pero después se estabilizan o crecen lentamente. Por ejemplo, en el caso de identificación con IntProy, el segundo ejemplo mejora el ratio en un 10 %, el tercero un 7 %, el cuarto un 2 %, y así sucesivamente. El tamaño óptimo parece estar entre 14 y 16 muestras por persona. Más allá, la mejora es muy poco significativa. Está claro que el valor concreto depende de otros factores. Pero podemos extraer como conclusión que existe un tope teórico, a

partir del cual es innecesario añadir más muestras a la galería.

2. En el caso de la **identificación** –figura 6.30a)–, los tres métodos están igualados muy estrechamente. Las diferencias son de unos pocos puntos, y se produce una alternancia en el ganador para cada valor de k. Es destacable la mejora general que se consigue hasta $k = 6$, de unos 25 puntos porcentuales, pasando del 65 % a más del 90 %.

3. En **verificación** –figura 6.30b)–, las diferencias entre técnicas son más notables. Las tres curvas crecen también de forma logarítmica. Sin embargo, IntProy se encuentra en torno a 5 puntos por encima de los otros reconocedores. Es más, la distancia aumenta al usar más imágenes por persona. El método EigenFace está en segundo lugar, por delante de TemMatch. Estos dos últimos parecen tener dificultades para alcanzar el 100 % de verificación correcta, algo que no ocurre con IntProy.

En definitiva, siempre resulta conveniente incluir más de una muestra por individuo en la galería. El beneficio se notará más, por ejemplo, al pasar de 1 a 3 muestras, que de 3 a 5. Si la aplicación maneja entrada de vídeo, se podrían tomar diferentes capturas de cada persona con pequeñas variaciones de expresión, posición y orientación. Empero, hay que tener en cuenta que este factor está relacionado con otro aspecto fundamental de los sistemas de reconocimiento: la eficiencia computacional. Tanto los tiempos de ejecución como el consumo de memoria crecerán con el número de muestras usadas. Para tamaños medio-grandes (unas decenas de miles de personas), ambos recursos pueden volverse críticos. Seguidamente vamos a pasar a analizarlos en profundidad.

Eficiencia computacional de los reconocedores

La complejidad computacional ha sido una preocupación constante en el reconocimiento de personas. El aumento de velocidad de los ordenadores no ha aliviado la necesidad de conseguir mecanismos más rápidos, ya que paralelamente han crecido los tamaños de los conjuntos usados. Por ejemplo, las evaluaciones del programa FRVT [13, 139], manejan casos de hasta 37.400 individuos. Esto significa que para reconocer una imagen nueva en 1 segundo, la comparación entre dos muestras no debería tardar más de 0,03 milisegundos.

Pero el **tiempo de reconocimiento** –ya sea identificación o verificación– no es el único factor de interés. También debemos considerar el **coste del proceso de entrenamiento** y el **consumo de memoria**. En las técnicas analizas en nuestras pruebas, los tres recursos dependen del número total de imágenes de la galería, siendo prácticamente nula la influencia de otros parámetros como el número de personas o el tamaño de las imágenes.

En consecuencia, para este experimento tomamos distintos fragmentos del grupo "faces94" de la base ESSEX; el número de muestras por individuo se mantiene a 10, y modificamos la cantidad total de imágenes, tomando los valores: 200, 400, 600, 800, 1000 y 1200. El conjunto de prueba consta siempre de 200 imágenes. Para cada tamaño de entrada, se mide:

(1) el tiempo de realizar el entrenamiento;

(2) el tiempo en reconocer todo el conjunto de prueba –que después se divide por 200–;

(3) la memoria usada por el proceso –medida de forma global y aproximada–.

La medición se repite 5 veces, calculando después los promedios. Todos los tiempos incluyen la lectura de los ficheros de imagen. Para evitar la influencia de la caché de disco, antes de realizar las pruebas se leen las imágenes de la base. Como los archivos ocupan en el disco menos de 16 Mbytes, todos ellos caben en la caché, de manera que se garantiza que las imágenes se obtienen siempre de memoria. Los resultados aparecen en la tabla 6.13. En la figura 6.31 se representan los tiempos en función del tamaño del problema.

Procesador	Intel (R) Pentium IV
Velocidad del procesador	2,60 GHz
Memoria caché	8 Kb (1^{er} nivel) + 512 Kb (2^{o} nivel)
Memoria RAM	512 Mb

Tamaño galería (# img.)	**IntProy**			**TemMatch**			**EigenFace**			**HMM**		
	Entr. (s)	**Test (ms)**	**Mem (Mb)**	**Entr. (s)**	**Test (ms)**	**Mem (Mb)**	**Entr. (s)**	**Test (ms)**	**Mem (Mb)**	**Entr. (s)**	**Test (ms)**	**Mem (Mb)**
200	**0,54**	**4,08**	**0,72**	0,79	8,03	0,77	2,52	5,46	1,36	3,14	64,6	2,62
400	**1,14**	**5,97**	1,25	1,5	12,67	**1,07**	19,35	6,82	1,33	5,6	124,3	3,67
600	**1,69**	**7,5**	1,75	2,34	17,01	**1,32**	77,22	8,0	1,64	8,7	196,1	5,59
800	**2,26**	**9,22**	2,25	3,14	21,42	**1,58**	220,36	9,98	1,92	11,75	261,3	7,06
1000	**3,08**	12,33	2,81	3,97	26,33	**1,83**	455,86	**11,96**	2,21	14,47	362,3	8,46
1200	**3,98**	14,73	3,32	4,88	32,53	**2,13**	889,28	**13,77**	2,51	17,71	467,4	9,73

Tabla 6.13: *Eficiencia computacional de los distintos métodos de reconocimiento, sobre fragmentos del grupo "faces94" la base ESSEX. Arriba se muestran los datos del ordenador utilizado en las pruebas. Abajo aparecen los resultados. Para cada método y tamaño de la galería, se indica el tiempo del proceso de entrenamiento (en segundos), el tiempo medio de prueba por imagen, desde la lectura hasta la obtención de las puntuaciones (en milisegundos), y la memoria consumida por el proceso (en Mbytes) medida de forma aproximada. Se señalan en negrita los mínimos para cada caso.*

Los valores obtenidos son bastante concluyentes en muchos aspectos. Vamos a señalar algunos de los principales:

1. En relación a los **tiempos de entrenamiento**, existe una clara diferencia entre los que muestran un crecimiento lineal y los que aumentan más rápidamente. Dentro de los primeros, IntProy y TemMatch se encuentran en órdenes de magnitud parecidos, aunque el método de proyecciones es alrededor de un 30 % más rápido. También HMM crece linealmente con el número de imágenes; sin embargo, es más de 4 veces más lento que los otros. Un caso muy diferente es el de EigenFace, con un orden de complejidad muy superior al lineal. En concreto, su tiempo de entrenamiento es un $O(n^3)$, siendo n el número de imágenes de la galería. Este hecho hace que sea excesivamente costoso aplicar la técnica para tamaños muy grandes. Por ejemplo, suponiendo el mismo crecimiento de la tabla 6.13, para tamaño 37.000 el proceso tardaría unos 290 días. El punto crítico es el cálculo de la descomposición en valores singulares (SVD).

   Existen muchas formas de reducir estos altos requerimientos. Por ejemplo, la SVD de una matriz M y de M^{T} están estrechamente relacionadas, pero no son igual de costosas.

Sea M de $n \times m$, con n el número de imágenes y m el de píxeles por imagen. Si $n < m$ será más rápido calcular la SVD de M, y en otro caso la de M^{T}. Otra posibilidad es utilizar algún mecanismo, por ejemplo *clustering*, para reducir el número de imágenes usadas. Es más, algunos autores proponen que debería usarse una autobase calculada a priori [164], y no necesariamente deducida del conjunto de entrenamiento.

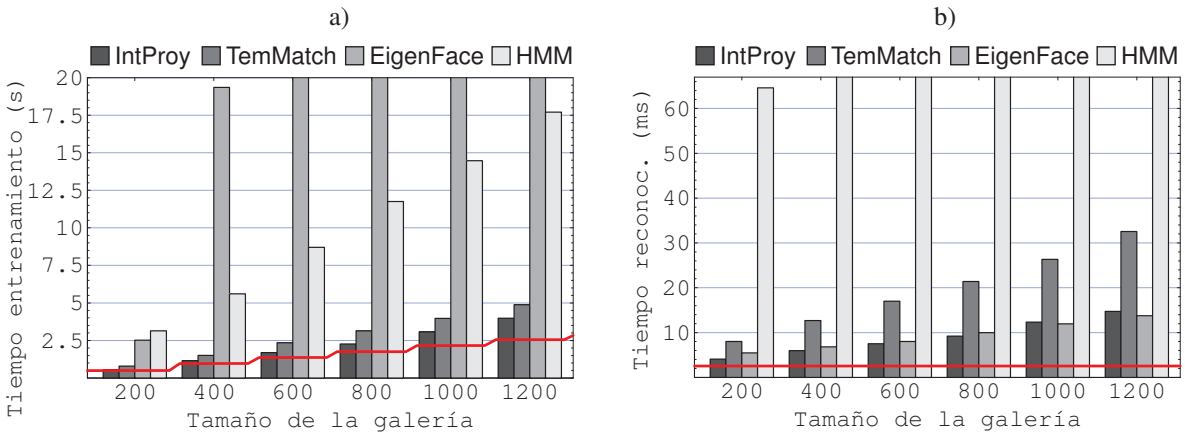


Figura 6.31: *Tiempos de ejecución de los distintos métodos de reconocimiento, en función del número total de imágenes de la galería. Los datos del experimento se encuentran en la tabla 6.13. a) Tiempos del proceso de entrenamiento. b) Tiempos medios de reconocimiento. La línea roja indica el tiempo requerido para la lectura de los ficheros.*

2. En muchas aplicaciones el **tiempo de reconocimiento** es más crítico que el de entrenamiento. Aquí, todos los métodos exhiben una relación lineal con el número de imágenes de la galería. Las técnicas IntProy y EigenFace resultan las más rápidas, necesitando poco más de 10 milisegundos para comparar una prueba con 1000 muestras de la galería (unos 0,01 ms por muestra). El método basado en comparación de patrones –que usa las funciones optimizadas de las librerías OpenCV– tarda algo más del doble. Este resultado es bastante lógico: los dos primeros trabajan en espacios de reducida dimensionalidad, mientras que el segundo compara directamente imágenes 2D.

   Bastante alejado es el crecimiento de HMM, que para tamaños grandes es unas 30 veces más costoso que los métodos más rápidos. Hay que tener en cuenta las distintas condiciones de cada implementación (la de HMM ya viene dada, mientras que las otras han sido creadas por nosotros); no obstante, está claro que la aplicación de los modelos HMM resulta de por sí menos eficiente que la comparación directa realizada por los otros métodos.

3. En cuanto al **consumo de memoria**, también existe en todos los métodos una dependencia lineal con el número de imágenes. Puesto que las caras son de tamaño pequeño, la memoria utilizada no resulta especialmente elevada en ningún caso. Curiosamente, la técnica TemMatch, que almacena directamente las imágenes extraídas de la galería, es la que presenta menos requisitos de memoria. Algunos factores secundarios, como el uso

de variables auxiliares y la forma de reservar memoria, pueden influir en este resultado. En cualquier caso, la optimización del tiempo ha primado sobre la del uso de memoria. Destacan nuevamente los altos requerimientos de HMM.

En conjunto, el reconocimiento basado en integrales proyectivas es el único método que consigue mantener bajo el consumo de todos los recursos: el entrenamiento añade poca carga a la propia lectura de las imágenes de disco; el reconocimiento es tan eficiente como las técnicas más rápidas; y el uso de memoria permitiría manejar algunos cientos de miles de individuos sin sobrepasar las capacidades de almacenamiento de un ordenador medio.

6.4.3. Resultados sobre la base ORL

Esta base es una de las clásicas en el reconocimiento facial de personas, pero a su vez una de las más antiguas. Las imágenes incluyen variaciones en expresión, tamaño, orientación y uso de gafas. Algunos autores argumentan que no es adecuada para la evaluación a gran escala [142], por el bajo número de individuos que contiene (sólo 40), y porque muchos sistemas consiguen ya unos rendimientos próximos al 100 %. Por este motivo, no profundizaremos excesivamente en la experimentación con este conjunto.

Todas las imágenes de la base ORL están más o menos centradas en las caras, abarcando verticalmente desde el pelo hasta la barbilla, y horizontalmente los bordes exteriores de la cabeza. Esto hace innecesarios los pasos de detección y localización. Simplemente, las proyecciones de cara se aplican tomando posiciones fijas de ojos y boca. De las 10 imágenes existentes por persona, tomamos aleatoriamente 5 para entrenamiento y otras 5 para prueba.

En la tabla 6.14 se muestran los principales resultados obtenidos sobre esta base para los métodos de reconocimiento disponibles. Las curvas CMC y ROC correspondientes aparecen en la figura 6.32. En todos los casos la clasificación es realizada mediante vecino más próximo.

Base caras	Nº personas	Img. galería	Img. prueba	Resolución	Variación
ORL	40	200 (5pp)	200 (5pp)	70×80	Expresión, pose, gafas

	Identificación			Verificación				Tiempo
Reconocedor	$P_I(1)$	$P_I(5)$	$P_I(10)$	eer	FA=1 %	eern	FAn=1 %	(ms)
IntProy	**95,5**	**99,4**	**100**	**3,7**	**93,5**	**1,3**	**98,3**	**12,6**
TemMatch	92,5	96,9	**100**	8,4	81,7	3,1	93,7	27,6
EigenFace	91	96,9	99,4	5,2	87,4	5,2	91,6	22,0
HMM	92,5	–	–	–	–	–	–	67,0

Tabla 6.14: *Resultados del reconocimiento sobre la base ORL. Se muestran ratios de identificación y de verificación para distintos puntos de las curvas CMC y ROC. Los valores indican porcentajes.*

Vamos a valorar algunos aspectos de los resultados presentados:

1. En el caso de la **identificación en conjunto cerrado**, como habíamos previsto, todos los resultados se encuentran muy próximos entre sí. Por ejemplo, en la gráfica de la figura 6.32a) el eje vertical sólo representa valores entre 0,9 y 1. En términos absolutos, este

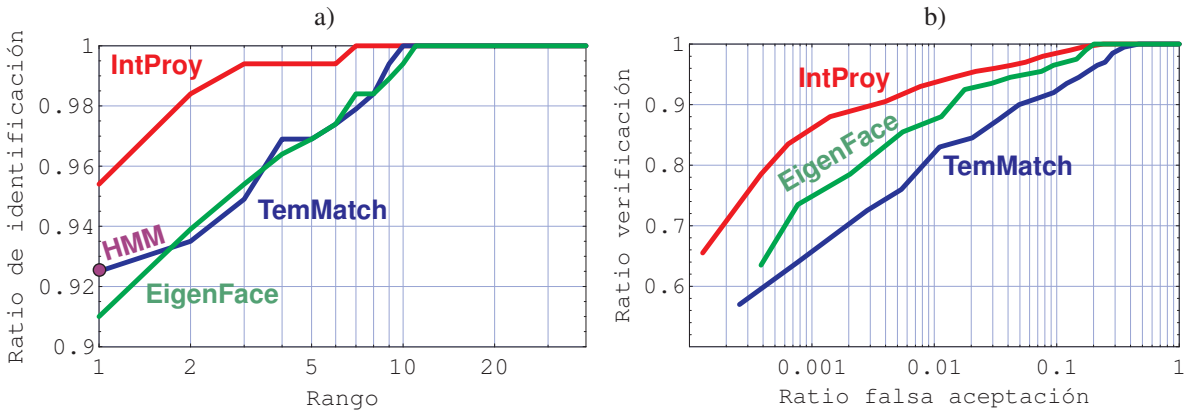


Figura 6.32: *Curvas CMC y ROC obtenidas sobre la base de caras ORL, para los métodos de reconocimiento basados en proyecciones (en rojo), comparación de patrones (en azul), y autocaras (en verde). a) Curvas CMC para identificación. b) Curvas ROC para verificación.*

intervalo corresponde únicamente a 20 imágenes. Además, a partir de rango 11 todos los métodos alcanzan el 100 % de identificación. A pesar de ello, podemos decir que el método basado en proyecciones está claramente por encima de los otros, con diferencias entre 3 y 5 puntos para rangos intermedios. El número absoluto de ejemplos mal identificados en IntProy es de 9; en la figura 6.33 se pueden ver los 6 casos incorrectos que producen menores distancias.

2. Las diferencias son más amplias en el **escenario de verificación**. Por ejemplo, para un 0,1 % de falsas aceptaciones, el método IntProy supera en unos 10 puntos a EigenFace y en 20 a TemMatch. En todos los casos, el reconocimiento mediante proyecciones está por encima. Lógicamente, las diferencias se reducen de forma progresiva hasta el punto donde convergen las tres curvas. Se han mostrado aquí las curvas ROC antes de normalizar las puntuaciones. En la tabla 6.14 se puede analizar la gran mejora que ocurre después de realizar este paso. Por ejemplo, IntProy consigue reducir 3 veces el ratio de error igual, situándolo en un 1,3 %.

3. Aunque el método de autocaras se encuentra por debajo de TemMatch para rango 1, a medida que aumenta el rango ambos se igualan y crecen a la par. Es más, si nos fijamos en la curva ROC de la figura 6.32b), EigenFace es mejor que TemMatch para todas las falsas aceptaciones. Esta **"inversión" de los resultados** es infrecuente, pero posible. La bondad de la verificación está relacionada con la capacidad de producir puntuaciones muy separadas, según el individuo sea el mismo o diferente. Sin embargo, en identificación esa separación es irrelevante, y lo importante es cuál es la máxima puntuación –o la mínima distancia– aunque sea por un estrecho margen.

4. Los resultados obtenidos para IntProy son comparables con los que se pueden encontrar en [127], donde se hace una **recopilación de ratios de identificación correcta** alcanzados con diferentes acercamientos. No se ofrecen más datos ni están claras las condiciones de

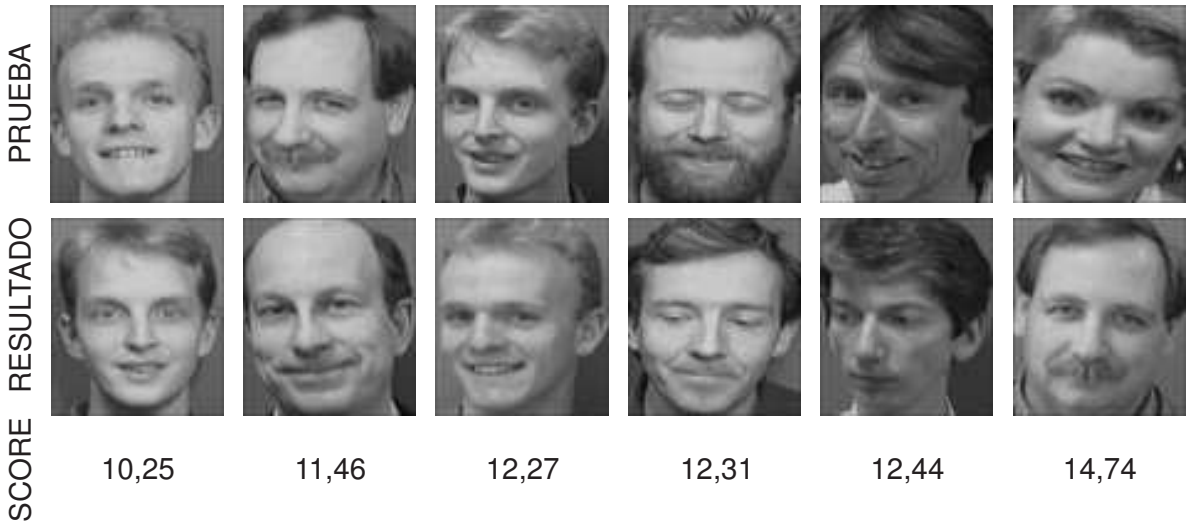


Figura 6.33: *Ejemplos de identificaciones incorrectas de IntProy en la base de caras ORL. Arriba, las imágenes de entrada. En medio, muestras de la identidad resultante. Abajo, la puntuación para la identidad resultante.*

experimentación en todos los casos. De forma orientativa, los porcentajes para diversos tipos de métodos basados en HMM están entre 85 % y 98 %; se detallan también algunos trabajos con redes neuronales, siendo el mejor caso el de las redes convolucionales, con un 96,2 %. Otros acercamientos logran porcentajes algo inferiores, entre el 80 % y el 92 %. Debemos recordar, no obstante, que no es adecuado realizar una comparación directa; en especial, el número de imágenes usadas para prueba o para la galería puede tener una influencia muy grande, como ya analizamos en el apartado anterior.

6.4.4. Resultados sobre la base FERET

Hoy por hoy, podemos afirmar sin temor a equivocarnos que la base FERET es un estándar *de facto* en la evaluación de los sistemas de reconocimiento de personas. Además del volumen de imágenes e individuos que contiene, posee algunas características muy útiles en relación a los otros conjuntos disponibles. La más importante es la definición de las llamadas "particiones", ficheros de texto que listan las imágenes de la galería, además de varios subconjuntos para el estudio de diferentes aspectos: variación de expresión, de iluminación, de tiempo entre capturas, etc. Esto posibilita la comparación en igualdad de condiciones entre técnicas implementadas por diferentes autores.

Además de la propia base [52], se encuentran a disposición pública los resultados de las diversas evaluaciones del programa FERET [143, 144, 141, 142]. Por ello, no aplicaremos aquí los métodos alternativos usados para las otras bases de caras, sino que haremos la comparación con los datos de las técnicas mencionadas. Algunas de ellas son métodos base para el análisis y contraste de resultados; hay también un sistema comercial; y el resto son las propuestas participantes en la evaluación (todas ellas han sido ya descritas en la sección 6.2). En concreto,

tenemos las siguientes:

- **arl_cor**, **arl_ef**. Métodos básicos provistos por el *Army Research Laboratory (ARL)*, basados en correlación normalizada y en autocaras, respectivamente.
- **ef_hd_ang**, **ef_hd_anm**, **ef_hd_l1**, **ef_hd_l2**, **ef_hd_md**, **ef_hd_ml1**, **ef_hd_ml2**. Siete variantes de la técnica de autocaras. En todas ellas se usa la misma representación. La diferencia se encuentra en las métricas de distancia definidas sobre el autoespacio. Por ejemplo, "ang" es ángulo, "l1" es distancia de Manhattan, "l2" es distancia euclídea, "m1" y "m2" son similares a "l1" y "l2" pero aplicando distancia de Mahalanobis. No se documenta claramente el significado de las otras.
- **excalibur**. Sistema comercial presentado por la compañía *Excalibur Technologies*.
- **mit_m95**. Corresponde al método de reconocimiento con autocaras propuesto por Turk y Pentland en [183], basado en una simple distancia euclídea en el autoespacio obtenido con PCA.
- **mit_s96**. Método mejorado de autocaras, desarrollado por Moghaddam y Pentland [125], consistente en el modelado de probabilidades en los *autoespacios duales de cara* (uno para las variaciones intra-clase y otro para las inter-clases).
- **umd_m97**. Reconocedor basado en *análisis de discriminantes lineales* (LDA), propuesto originalmente por Belhumeur y otros [9], y mejorado posteriormente por otros autores.
- **usc_m97**. Algoritmo *elastic bunch graph matching* (EBGM), creado por Wiskott y otros [192], basado en la comparación de grafos con información de filtros wavelet.

Para mantener una uniformidad en la comparación, usamos las posiciones etiquetadas de ojos y boca, incluidas junto con la base FERET. Todos los experimentos comparten la misma galería –denominada grupo "fa"–, que contiene un total de 1196 individuos con una sola imagen por persona. El conjunto de prueba se va modificando de un caso a otro.

Variación de la expresión facial

El primer experimento que vamos a presentar trata de evaluar la robustez de los sistemas de reconocimiento frente a la variación de expresiones faciales. La partición asociada a esta prueba es el grupo "fb" que incluye 1195 personas (una menos que la galería), con igual posición y condiciones de iluminación que "fa", pero cambiando de forma moderada la expresión de los sujetos. Los resultados obtenidos con el método basado en integrales proyectivas y con las técnicas disponibles se resumen en la tabla 6.15.

En la figura 6.34 se representan algunas curvas ROC y CMC de los métodos contenidos en la tabla 6.15. No se han añadido todos ellos por motivos de claridad.

Destacamos algunos de los aspectos más interesantes de estos resultados:

Conjunto	Grupo	Nº personas	Nº imágenes	Resolución	Variación
$\mathcal{G}$: galería	fa	1196	1196 (1pp)	256×384	–
$\mathcal{P}_\mathcal{G}$: pruebas	fb	1195	1195 (1pp)	256×384	Expresión

	Identificación			Verificación				Tiempo
Reconocedor	$P_I(1)$	$P_I(5)$	$P_I(10)$	eer	FA=1 %	eern	FAn=1 %	(ms)
IntProy	81,5	90,7	92,9	3,4	90,8	2,7	95,9	16,8
arl_cor	82,7	92,6	95	4,1	88,4	–	–	–
arl_ef	79,7	89,2	92,1	6,7	82,4	–	–	–
ef_hd_ang	70,1	83,1	88,2	5,4	83,6	–	–	–
ef_hd_anm	77,4	89,5	93,1	2,9	93,4	–	–	–
ef_hd_l1	77,2	89,1	92,2	6,6	81,5	–	–	–
ef_hd_l2	71,6	85,6	89,6	5,1	82,9	–	–	–
ef_hd_md	74,1	86,9	90,7	3,5	89,5	–	–	–
ef_hd_ml1	73,3	81,6	83,8	12,1	68,8	–	–	–
ef_hd_ml2	77,2	88,1	91,6	7	80	–	–	–
excalibur	79,4	89,5	93	4,8	84,7	–	–	–
mit_m95	83,4	91,9	94,1	5,6	86,7	–	–	–
mit_s96	94,8	97,3	97,9	4,8	92,4	–	–	–
umd_m97	**96,2**	**98,5**	**99,1**	**1,2**	**98,5**	–	–	–
usc_m97	95	98,4	98,6	2,5	96	–	–	–

Tabla 6.15: *Resultados del reconocimiento sobre el grupo "fb" de la base FERET. Arriba se indican las imágenes usadas para la galería y para las pruebas. Abajo se muestran diversos ratios de identificación y verificación para algunos puntos de las curvas CMC y ROC.*

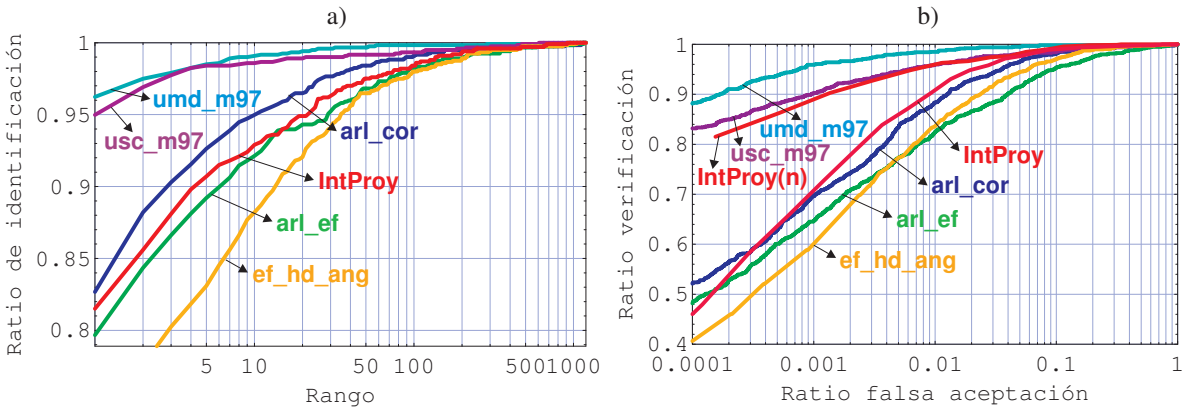


Figura 6.34: *Curvas CMC y ROC de distintos métodos de reconocimiento sobre el grupo "fb" de la base FERET. Los datos de la prueba están descritos en la tabla 6.15. Se muestran algunos de los métodos más relevantes. a) Curvas CMC para identificación. b) Curvas ROC para verificación.*

1. El **reconocimiento mediante proyecciones** se muestra claramente superior a la mayoría de los métodos basados en autocaras. No obstante, se encuentra bastante próximo a las técnicas básicas "arl_cor" y "arl_ef". Los tres consiguen ratios de identificación correcta sobre el 80 %. Por su parte, las propuestas más avanzadas consiguen subir por encima del 96 %. No obstante, quedan atrás los casos del sistema comercial "excalibur" y "mit_m95", que están en los mismos intervalos que IntProy. Esto es una prueba de la complejidad implícita de este experimento, aun siendo menos difícil que los que veremos a continuación.

2. Evidentemente, **los porcentajes** de reconocimiento **disminuyen** respecto a las otras bases de caras, debido a dos hechos: el número de personas es mucho mayor, y sólo manejamos una muestra por individuo. Teniéndolo en cuenta, los resultados conseguidos se pueden considerar como bastante positivos. Por ejemplo, para rango 20 el reconocimiento mediante proyecciones alcanza el 95 %, y sobre 200 (una sexta parte de la galería) prácticamente llega al 100 %.

3. En relación al **problema de verificación**, IntProy mejora los datos de "arl_cor" y "arl_ef", que antes estaban tan próximos. Esto denota un mejor comportamiento de las puntuaciones producidas por el método, que presentan una mayor separación entre las distancias para la misma o diferentes personas.

4. Si consideramos las **puntuaciones normalizadas**, el método basado en proyecciones sería el tercero de los evaluados[19]. Para un 1 % de falsas aceptaciones el ratio de identificación es de casi el 96 %, lo cual mejora los resultados de algunas de las bases de caras más reducidas, documentadas en los apartados previos.

Como indicamos en el apartado 6.3.1, la evaluación inicial del programa FERET incluía un reconocedor basado en proyecciones, debido a Wilder [190], aunque usando un acercamiento diferente al nuestro. Los resultados de este método quedan bastante por detrás de los obtenidos con IntProy. Por ejemplo, en [141] se documenta un experimento idéntico al de la tabla 6.15[20]; el ratio de identificación correcta de [190] es unos 8 puntos inferior al de IntProy, y el valor $P_I(10)$ está por debajo del 88 %.

Debemos puntualizar, sin embargo, las diferentes condiciones en el desarrollo de los métodos alternativos. Las imágenes de la base FERET no estaban disponibles para los diseñadores de esos sistemas; mientras que en nuestro caso la base está accesible, y ha sido usada para el ajuste de algunos parámetros del algoritmo, como hemos descrito en la sección 6.3.

Variación de la iluminación

El grupo "fc" de la base FERET está orientado al análisis de la robustez frente a cambios de iluminación. El efecto sobre la apariencia de las caras no se reduce a un simple aumento o reducción de los niveles de gris en las imágenes, sino que es mucho más complejo: aparición de sombras, iluminación no uniforme, variación del contraste, etc. Por ello, el impacto en los sistemas de reconocimiento puede ser bastante notable. En cualquier caso, el estudio de este factor es fundamental en muchas aplicaciones donde no se puede garantizar que se mantengan siempre las mismas condiciones de luz.

Los datos y los resultados de este experimento se encuentran en la tabla 6.16. El conjunto de prueba contiene únicamente 194 imágenes de otros tantos individuos. Sin embargo, esto

[19] Aunque, obviamente, la comparación aquí no es adecuada, ya que el mismo proceso de normalización se podría aplicar sobre los otros métodos, mejorando igualmente sus resultados.

[20] Lamentablemente, sólo se muestra la gráfica para el caso de identificación, y no se aportan datos numéricos.

no reduce la dificultad del problema, ya que la galería sigue constando de 1196 personas. Es decir, hay más de 1000 sujetos en la galería que no aparecen en la prueba.

Conjunto	Grupo	Nº personas	Nº imágenes	Resolución	Variación
$\mathcal{G}$: galería	fa	1196	1196 (1pp)	256×384	–
$\mathcal{P}_\mathcal{G}$: pruebas	fc	194	194 (1pp)	256×384	Iluminación

	Identificación			Verificación				Tiempo
Reconocedor	$P_I(1)$	$P_I(5)$	$P_I(10)$	eer	FA=1 %	eern	FAn=1 %	(ms)
IntProy	31,4	54,1	64,4	15,3	39,5	11,1	65,4	21,2
arl_cor	5,2	11,3	17,5	24,7	12,4	–	–	–
arl_ef	18,6	42,8	52,1	16	42,8	–	–	–
ef_hd_ang	7,2	11,9	21,1	18,4	18,4	–	–	–
ef_hd_anm	23,7	45,4	53,1	10,3	55,2	–	–	–
ef_hd_l1	25,8	47,9	53,1	14,4	47,5	–	–	–
ef_hd_l2	4,1	12,4	17	22,5	11,9	–	–	–
ef_hd_md	23,2	44,3	52,1	10,6	52,6	–	–	–
ef_hd_ml1	39,2	53,1	64,9	16	53,1	–	–	–
ef_hd_ml2	30,9	48,5	57,2	12,9	50,5	–	–	–
excalibur	21,6	43,8	54,1	14,4	41,2	–	–	–
mit_m95	15,5	28,9	37,1	18,6	30,4	–	–	–
mit_s96	32	59,8	67	16	49,5	–	–	–
umd_m97	58,8	79,9	86,6	10	66,5	–	–	–
usc_m97	**82**	**90,2**	**91,8**	**5,1**	**90,2**	–	–	–

Tabla 6.16: *Resultados del reconocimiento sobre el grupo "fc" de la base FERET. Arriba se indican las imágenes usadas para la galería y para las pruebas. Abajo se muestran diversos ratios de identificación y verificación para algunos puntos de las curvas CMC y ROC.*

Igual que antes, la figura 6.35 contiene las curvas CMC y ROC de algunos de los métodos más representativos de la tabla 6.16.

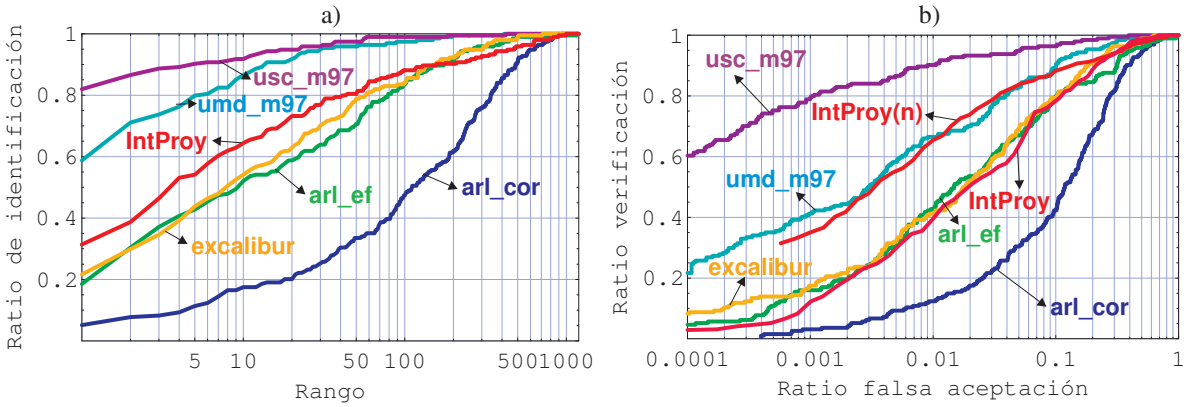


Figura 6.35: *Curvas CMC y ROC de distintos métodos de reconocimiento sobre el grupo "fc" de la base FERET. Los datos de la prueba están descritos en la tabla 6.16. Se muestran algunos de los métodos más relevantes. a) Curvas CMC para identificación. b) Curvas ROC para verificación.*

Vamos a interpretar y valorar estos resultados:

1. De forma global, la **degradación del rendimiento** es muy elevada. En identificación, los métodos básicos caen del 80 % a poco más del 20 %. Además, las diferencias entre los

diversos sistemas se acentúan, aunque algunos de los más avanzados ofrecen también porcentajes bastante pobres. El único reconocedor que muestra un comportamiento más o menos robusto es "usc_m97", con un ratio $P_I(1)$ del 82 %.

Lógicamente, el descenso afecta también a IntProy, con un ratio de identificación del 31 %. A pesar de ello, si lo comparamos con el resto de técnicas, consigue ganar un puesto, pasando al 5º lugar en una hipotética clasificación por el valor de $P_I(1)$. Podemos concluir que la iluminación afecta a las proyecciones en grado parecido, o ligeramente menor, que el resto de técnicas. Merece la pena mencionar también los resultados del método de proyecciones de Wilder [190], que en esta prueba no pasa del 15 % de identificación correcta [141]. Aunque el ratio es ciertamente bajo, téngase en cuenta que el valor esperado de $P_I(1)$ para un método completamente aleatorio sería del 0,08 %, y para $P_I(10)$ del 0,8 %.

2. La mayor degradación del rendimiento la protagoniza "arl_cor". Este hecho es bastante significativo. Se demuestra que los buenos resultados de la correlación sobre el conjunto "fb" son meramente indicativos de un alto parecido entre las imágenes de "fa" y las de "fb". Pero al verse sometido a condiciones más complejas, el algoritmo falla de manera rotunda. Así, se puede decir que esa técnica no captura realmente la información relevante del individuo, sino que se basa en una **similitud de apariencia a bajo nivel**.

3. En cuanto a los resultados en el escenario de verificación, IntProy sigue ocupando una **posición intermedia** entre las técnicas básicas y algunas de las más avanzadas. Otro dato interesante es que la normalización de puntuaciones no consigue mejorar significativamente los ratios de verificación. Esto es debido al relativamente bajo número de pruebas con rango 1.

En conclusión, está claro que un tratamiento adecuado de la iluminación requiere diseñar un mecanismo de modelado y compensación de los efectos no triviales de la variación de fuentes de luz. Los acercamientos usados en esta prueba están basados en la apariencia plana de las caras, y sólo son robustos frente a cambios globales de los niveles de gris.

Variación del tamaño de la galería

El gran volumen de imágenes de la base FERET permite diseñar pruebas para medir el efecto del tamaño de la galería en los resultados del reconocimiento. Lógicamente, se espera que los ratios de identificación y verificación disminuyan cuanto mayor sea el número de personas en la galería. El objetivo es cuantificar qué forma toma ese decrecimiento.

Este experimento se desarrolla de la siguiente manera. Se usa el grupo "fa" para la galería y "fb" para la prueba, seleccionando aleatoriamente las k mismas personas en uno y en otro. El tamaño, k, toma los valores: 1, 5, 10, 20, 30, 50, 70, 100, 150, 200, 250, 300, 400, 500, . . ., 1100 y 1196. Para cada prueba se mide el ratio de identificación correcta. Los 12 primeros tamaños

se repiten 20 veces, y los restantes 10 veces. Finalmente, se calculan los valores medios para cada tamaño.

Las evaluaciones de FERET incluyen un análisis similar [52], en el que todos los métodos se aplican sobre diferentes fragmentos de los grupos "fa" y "fb". A diferencia de nuestro caso, se estudian de forma exhaustiva todos los tamaños desde 1 hasta 1196. Desafortunadamente, no se ofrecen resultados numéricos sino únicamente las gráficas de los ratios en función del número de personas de la galería. En la figura 6.36 se muestran algunas de ellas, junto con los resultados del experimento para el método de integrales proyectivas. Obsérvese que ambas gráficas representan los mismos datos, usando en un caso escala lineal para el tamaño y en el otro logarítmica.

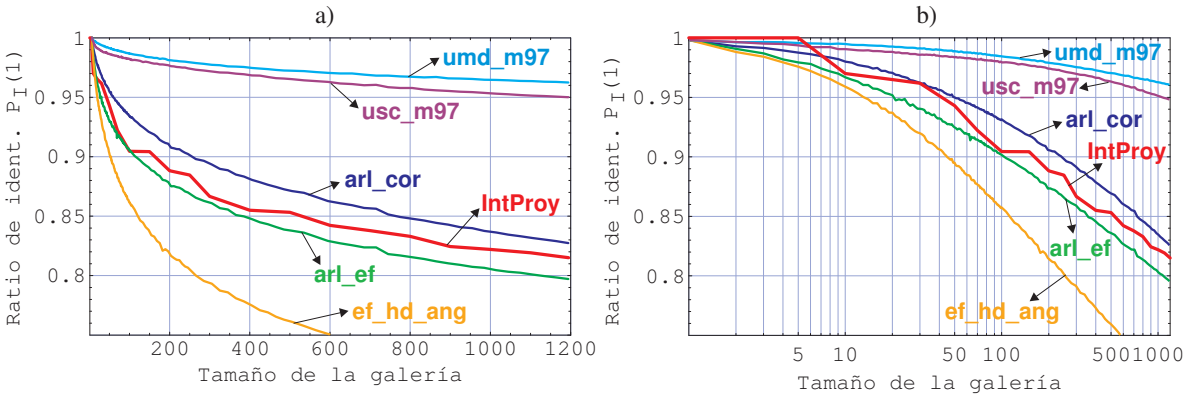


Figura 6.36: *Variación del ratio de identificación correcta en función del tamaño de la galería. La galería está formada por fragmentos del grupo "fa" de FERET, y la prueba se toma del grupo "fb". Se muestran los resultados del método basado en proyecciones (en rojo) y algunos de los métodos más relevantes de la evaluación FERET [52]. a) Con escala lineal. b) Con escala logarítmica en el tamaño de la galería.*

Podemos establecer varias conclusiones relevantes:

1. El **decrecimiento** del ratio de identificación parece adoptar una forma **logarítmica** para la mayoría de los métodos –muy acusado para tamaños pequeños y más suave para los grandes–. Se puede constatar más claramente este hecho en la gráfica de escala logarítmica de la figura 6.36b). Este resultado es coherente con las conclusiones de la evaluación FRVT 2002 [139], donde se estableció que cada vez que se duplica el tamaño de la galería los porcentajes de identificación descienden entre 2 y 3 puntos.

2. De manera colateral, este experimento nos ha permitido valorar el efecto del **factor individual** en el reconocimiento, ya que la prueba se repite 10 ó 20 veces para cada tamaño (con diferentes subconjuntos escogidos aleatoriamente). La desviación típica de los resultados de IntProy para un mismo tamaño se encuentra normalmente sobre los 2 puntos. Por ejemplo, para 150 personas, el mejor ratio obtenido es del 97 % mientras que el peor caso apenas llega al 84 %. Fijándonos de nuevo en el informe de FRVT 2002 [139], encontramos resultados similares. Por ejemplo, se comprobó que los hombres se reconocen entre 6 y 9 puntos porcentuales mejor que las mujeres; es más, el reconoci-

miento mejora con la edad de los sujetos, aproximadamente unos 5 puntos por cada 10 años.

Si bien los resultados globales indican que los sistemas de reconocimiento funcionarán mejor con tamaños muy pequeños de la galería, el segundo punto implica la imposibilidad de garantizar a priori unos ratios medios esperados. Estos dependerán en gran medida de los ejemplos concretos que se usen.

Variaciones en la fecha de captura

Una de las grandes dificultades de los sistemas de reconocimiento es conseguir invarianza frente a los cambios de apariencia facial debidos al paso del tiempo. La base FERET es una de las primeras en abordar esta cuestión. Los llamados "duplicados" contienen imágenes de personas de la galería tomadas con hasta tres años de diferencia. En concreto, existen dos particiones: "dup1" y "dup2". Aunque, a grandes rasgos, la primera contiene menores diferencias en fechas de captura que la segunda, no hay un criterio claro de separación. Así, en "dup1" las fechas van entre 1 mes y 3 años, y en "dup2" entre 18 meses y 3 años.

Debemos señalar, también, que el cambio de fecha implica normalmente una variación en las condiciones de iluminación, ya que las diferentes sesiones de captura se realizaron en escenarios distintos. Otro inconveniente es que no se dispone de duplicados para todos los individuos de "fa". En "dup1" hay 243 personas distintas y en "dup2" sólo 75. Recordemos, no obstante, que la galería sigue constando de 1196 sujetos.

La ejecución de IntProy sobre los duplicados ha requerido la modificación de algunos parámetros. Esto afecta principalmente a dos aspectos: las regiones proyectadas están más ajustadas a la zona de la cara; y aumenta ligeramente la resolución de las proyecciones. En definitiva, los resultados conseguidos por el método propuesto, en comparación con los presentados en la evaluación FERET, se detallan en la tabla 6.17.

En este caso, la figura 6.37 no se refiere a los diferentes métodos disponibles, sino que analiza de forma comparada los resultados del reconocimiento basado en proyecciones sobre los duplicados y sobre el grupo "fb".

Hagamos algunos comentarios en relación a estos nuevos datos:

1. La **degradación** de los resultados **afecta de manera desigual** a las técnicas existentes. El orden de la reducción es similar al del grupo "fc". Sin embargo, en este caso se estrechan las distancias entre los métodos básicos y los avanzados. En identificación, la diferencia entre el mejor y el peor algoritmo no supera los 26 puntos en "dup1" y los 40 en "dup2", cuando en "fc" llegaba a los 80 puntos porcentuales. El problema es más claro todavía en verificación, donde algunos métodos básicos logran los mejores ratios. Esto es una muestra de la complejidad implícita de la prueba y de los drásticos cambios de apariencia provocados por el transcurso del tiempo.

   La evaluación FRVT 2000 [13], repitió el mismo experimento para medir la mejora conseguida con los sistemas comerciales disponibles tres años más tarde. En "dup1" los

Conjunto	Grupo	Nº personas	Nº imágenes	Resolución	Variación
$\mathcal{G}$: galería	fa	1196	1196 (1pp)	256×384	–
$\mathcal{P}_{\mathcal{G}1}$: pruebas	dup1	243	722 (3,0pp)	256×384	Fecha captura 1-36 meses
$\mathcal{P}_{\mathcal{G}2}$: pruebas	dup2	75	234 (3,1pp)	256×384	Fecha captura 18-36 meses

Conjunto "dup1"

	Identificación			Verificación				Tiempo
Reconocedor	P_I(1)	P_I(5)	P_I(10)	eer	FA=1 %	eern	FAn=1 %	(ms)
IntProy	36,7	48,2	55,8	17,7	43,9	16,6	56,0	17,3
arl_cor	36,3	48,6	54	18,7	44,6	–	–	–
arl_ef	41,0	53,5	59,7	17,3	44,7	–	–	–
ef_hd_ang	34,1	45,4	50,8	17,6	45,4	–	–	–
ef_hd_anm	44,6	56,8	62,9	10,7	63,2	–	–	–
ef_hd_l1	35,0	46,0	51,7	21,9	41,4	–	–	–
ef_hd_l2	33,1	44,5	50,8	19,4	42,5	–	–	–
ef_hd_md	42,2	55,3	62,2	**10,4**	57,2	–	–	–
ef_hd_ml2	34,6	47,0	52,2	22,3	39,8	–	–	–
excalibur	41,4	55,3	60,5	15,4	51,0	–	–	–
mit_m95	33,8	46,8	54,0	18,4	41,7	–	–	–
mit_s96	57,6	68,4	**72,6**	17,6	55,4	–	–	–
umd_m97	47,2	60,4	67	12,6	63,4	–	–	–
usc_m97	**59,1**	**69,4**	72,3	13,2	**63,9**	–	–	–

Conjunto "dup2"

	Identificación			Verificación				Tiempo
Reconocedor	P_I(1)	P_I(5)	P_I(10)	eer	FA=1 %	eern	FAn=1 %	(ms)
IntProy	30,3	46,2	51,7	17,6	48,6	14,5	53,5	22,7
arl_cor	17,1	33,3	39,3	19,2	34,2	–	–	–
arl_ef	22,2	36,8	46,2	19,0	38,5	–	–	–
ef_hd_ang	12,4	23,5	30,8	22,6	26,1	–	–	–
ef_hd_anm	20,9	31,6	36,3	13,9	34,2	–	–	–
ef_hd_l1	13,2	25,2	32,1	25,6	28,6	–	–	–
ef_hd_l2	13,7	24,4	31,2	21,8	27,8	–	–	–
ef_hd_md	16,7	29,5	33,8	13,7	30,3	–	–	–
ef_hd_ml1	12,8	20,5	26,1	34,9	21,8	–	–	–
ef_hd_ml2	12,8	26,1	32,1	25,9	27,8	–	–	–
excalibur	19,7	33,3	41,0	18,4	37,6	–	–	–
mit_m95	17,1	30,8	41,0	21,6	30,3	–	–	–
mit_s96	34,2	50,0	59,8	20,5	36,3	–	–	–
umd_m97	20,9	42,7	53,4	**13,2**	45,7	–	–	–
usc_m97	**52,1**	**66,7**	**70,5**	14,2	**57,3**	–	–	–

Tabla 6.17: *Resultados del reconocimiento sobre los grupos "dup1" y "dup2" de la base FERET. Arriba se indican las imágenes usadas para la galería y para las pruebas. Abajo se muestran diversos ratios de identificación y verificación para algunos puntos de las curvas CMC y ROC. También aparece el ratio de error igual (eer) y el tiempo de ejecución por imagen.*

mejores algoritmos consiguieron un 63 % y en "dup2" un 64 %. El progreso es, por lo tanto, muy lento. El efecto del tiempo sigue siendo aún una cuestión abierta.

2. Los **resultados de IntProy** demuestran un descenso muy similar para ambos grupos, como se puede ver en la figura 6.37. Este hecho lo sitúa en un lugar intermedio para

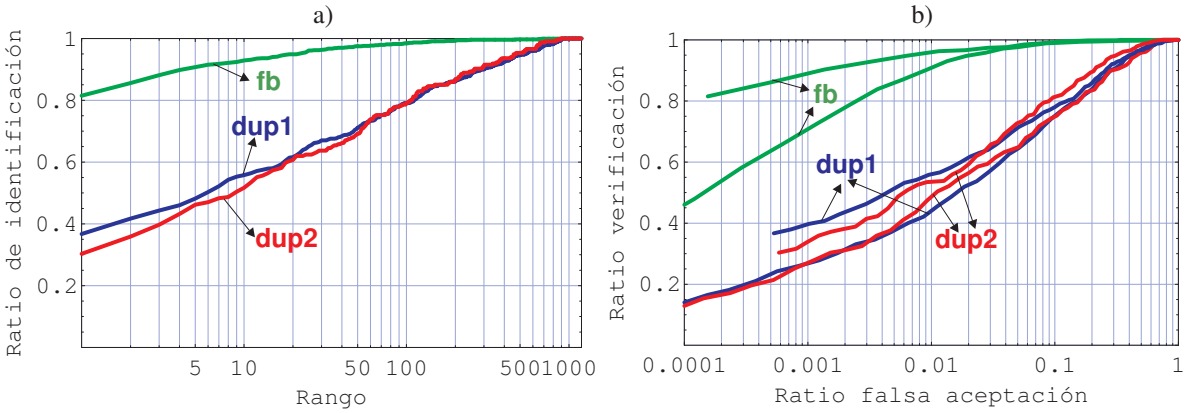


Figura 6.37: *Curvas CMC y ROC del reconocedor basado en proyecciones sobre los conjuntos de duplicados de FERET, "dup1" (en azul), "dup2" (en rojo) y "fb" (en verde). a) Curvas CMC de los tres conjuntos. b) Curvas ROC antes y después de normalizar las puntuaciones.*

el grupo "dup1" (el 8º de los 14 evaluados), mientras que para "dup2" consigue una interesante tercera posición. Las conclusiones son parecidas en la prueba de verificación. Es más, si nos fijamos en el ratio de verificación para un 1 % de falsas aceptaciones, se encontraría en segundo lugar para "dup2", con un 48,6 %.

Aunque los ratios de identificación para IntProy no son muy altos, siguen siendo mejores que para el método basado en proyecciones debido a Wilder [190]; así, por ejemplo, $P_I(1)$ está sobre el 30 % para "dup1" y cerca del 15 % para "dup2".

Los relativamente buenos resultados del reconocedor de integrales proyectivas para el conjunto más difícil, "dup2", son una muestra de cómo las proyecciones conservan la información que distingue a los individuos, a un nivel bastante superior al de otras técnicas, como las autocaras o la correlación. Sin embargo, es evidente que para conseguir buenos resultados hay que mejorar mucho los métodos existentes. Mientras tanto, los sistemas prácticos deberían incorporar mecanismos de actualización de la información que almacenan.

6.4.5. Resultados sobre la base GATECH

Acabamos la experimentación en reconocimiento de personas estudiando el rendimiento obtenido por los diferentes métodos sobre la base de caras del *Instituto Tecnológico de Georgia* [127], de Nefian y Hayes. El número de individuos aquí no es excesivamente grande, pero hay una interesante variación en expresiones faciales, posiciones, giros, fondo complejo y capturas en días diferentes. El contexto de trabajo de esta base podría corresponder a una aplicación donde varios usuarios –unas cuantas decenas, por ejemplo, los alumnos de un curso– comparten una serie de ordenadores y utilizan el sistema biométrico para identificarse y dar fe de su presencia.

Como en el caso de la base ESSEX, las posiciones de los rostros son obtenidas mediante la aplicación de los algoritmos de detección y localización. Muchas caras presentan una in-

clinación elevada, lo que hace que el porcentaje de detección esté próximo al 93 %. Tomamos aleatoriamente cerca de la mitad de los ejemplos para la galería y el resto para prueba.

Los resultados alcanzados por los cuatro métodos alternativos de reconocimiento se muestran en la tabla 6.18 y en la figura 6.38. Obsérvese que en IntProy se ha usado el valor óptimo del parámetro de ponderación de las proyecciones, α.

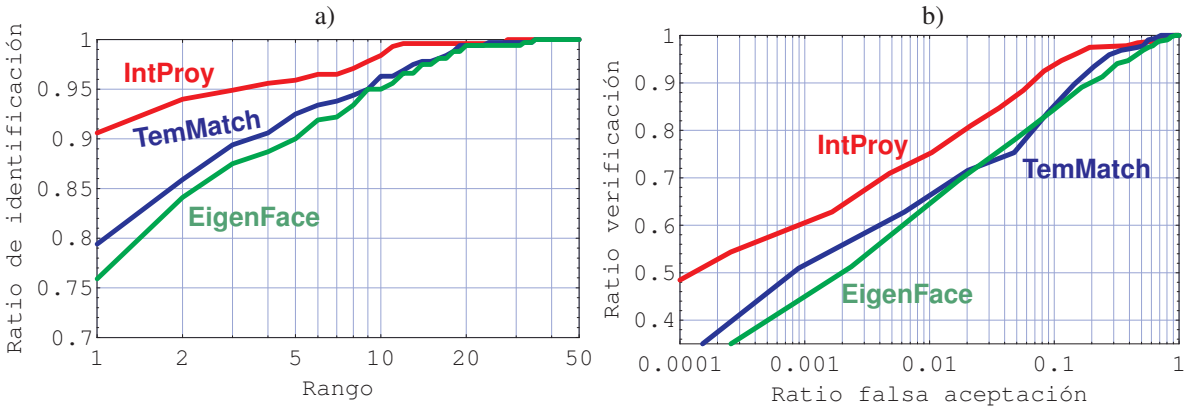


Figura 6.38: *Curvas CMC y ROC resultantes sobre la base de caras GATECH, para los métodos de reconocimiento basados en proyecciones (en rojo), comparación de patrones (en azul), y autocaras (en verde). La descripción de los casos se encuentra en la tabla 6.18. a) Curvas CMC para identificación. b) Curvas ROC antes de normalizar las puntuaciones.*

Base caras	Nº personas	Img. galería	Img. prueba	Resolución	Variación
GATECH	50	347 (6,9pp)	320 (6,4pp)	640 × 480	Expresión, pose, fecha

	Identificación			Verificación				Tiempo
Reconocedor	$P_I(1)$	$P_I(5)$	$P_I(10)$	**eer**	**FA=1 %**	**eern**	**FAn=1 %**	**(ms)**
IntProy	**90,6**	**95,9**	**98,4**	**8,0**	**75,0**	**3,1**	**93,8**	32,2
TemMatch	79,4	92,5	96,3	12,4	65,1	6,5	75,7	36,9
EigenFace	75,9	90,0	95,0	13,3	60,7	15,7	76,2	**30,5**
HMM	65,7	–	–	–	–	–	–	528,3

Tabla 6.18: *Resultados del reconocimiento sobre la base GATECH. Se muestran ratios de identificación y de verificación para distintos puntos de las curvas CMC y ROC. Los valores indican porcentajes.*

La gran disparidad de resultados entre los distintos métodos demuestra la dificultad implícita de esta base, debida especialmente a la diferencia entre fechas de captura y a los elevados ángulos de inclinación que presentan muchas de las caras. Hagamos una valoración de estos datos:

1. El **reconocedor de integrales proyectivas** se muestra manifiestamente superior al resto de algoritmos, tanto en identificación como en verificación. En el primer caso supera en 10 puntos al siguiente método, TemMatch, y en el segundo caso mejora casi 20 puntos el ratio de verificación para un 1 % de falsas alarmas, con normalización de puntuaciones. Esto indica que las proyecciones son mucho más robustas frente a giros y rotaciones que las autocaras o la comparación de patrones.

2. Todos los **tiempos de ejecución** se ven afectados por la mayor resolución de las imágenes de entrada. Esto conduce a una gran igualdad en los costes medios de reconocer una imagen. Las autocaras son ligeramente más rápidas que IntProy, posiblemente debido a optimizaciones de código a bajo nivel (que existen para EigenFace pero no para IntProy). Por su parte, HMM tarda más de medio segundo en procesar cada ejemplo.

3. Es bastante significativo el **mal dato del método HMM**, que no llega al 66 % de identificación correcta; y más si lo comparamos con los resultados expuestos en [127], donde se informa de un ratio del 87 % para ese mismo mecanismo. Incluso, los autores llegan a afirmar que "usando el mismo acercamiento descrito en el artículo pero con diferentes vectores de observación, se logra un ratio de reconocimiento del 92,5 %". Parte de la discrepancia se explica por las distintas condiciones de experimentación, básicamente en dos aspectos: (1) los autores utilizan posiciones etiquetadas manualmente; y (2) la galería contiene 10 imágenes por persona, en lugar de las 6,9 que tomamos nosotros.

Influencia de los canales de color

La base GATECH es especialmente interesante para analizar la expresividad de los distintos canales de color. A diferencia del conjunto ESSEX, las imágenes tienen una resolución alta, iluminación uniforme y buena calidad cromática. Por ello, hemos repetido el experimento descrito previamente pero manejando por separado los canales R, G y B. La prueba es realizada únicamente con el método basado en integrales proyectivas. Los resultados se muestran gráficamente en la figura 6.39.

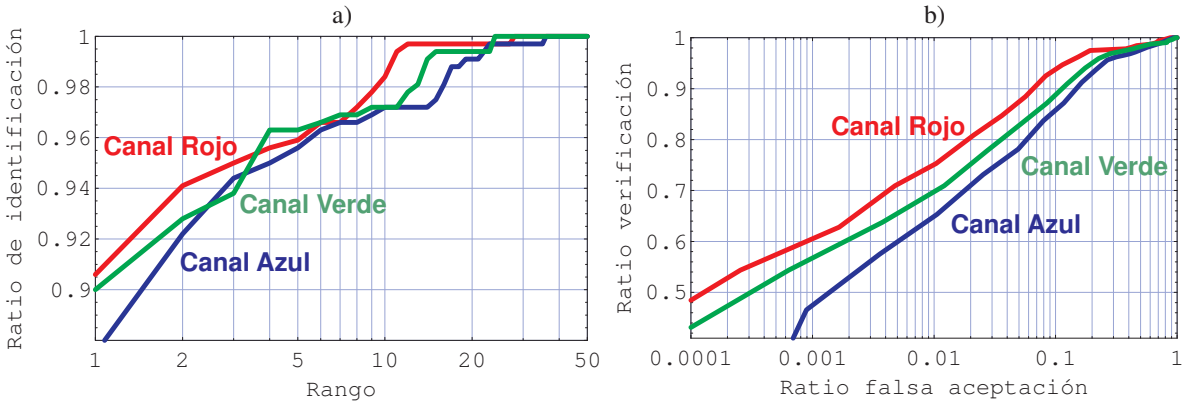


Figura 6.39: *Curvas CMC y ROC usando distintos canales de color de RGB en la base GATECH. Se usa el método de reconocimiento basado en proyecciones y los datos descritos en la tabla 6.18. Las gráficas corresponden a los canales R (en rojo), G (en verde) y B (en azul). a) Curvas CMC para identificación. b) Curvas ROC antes de normalizar las puntuaciones.*

Ya estudiamos la influencia del color en los problemas de reconocimiento facial dentro del apartado 6.3.3, donde vimos la conveniencia de usar el canal R. Las curvas de la figura 6.39 vienen a confirmar esta conclusión.

En el escenario de identificación en conjunto cerrado las diferencias entre canales son muy

pequeñas. Sin embargo, en verificación resulta más evidente la **superioridad del rojo** sobre los otros canales de color. En segundo lugar se sitúa el verde, que incluso llega a conseguir mejores resultados para algunos rangos en identificación. A mayor distancia, el canal azul produce los peores ratios en ambos escenarios. Por su parte, aunque no se haya representado en las gráficas de la figura 6.39, debemos decir que el valor de intensidad ofrece unos resultados intermedios, muy próximos a los del canal G.

6.5. Resumen y conclusiones

El panorama actual en reconocimiento facial de personas presenta una doble perspectiva. En un escenario más o menos controlado, manejando unos cuantos cientos de individuos interesados en ser reconocidos, los sistemas actuales son capaces de alcanzar rendimientos muy próximos al óptimo. Sin embargo, en condiciones muy variables de iluminación, pose, elementos faciales, fecha de captura, y sin la colaboración explícita de los usuarios, el reconocimiento de caras sigue siendo aún uno de los grandes problemas abiertos de la visión artificial. Y, posiblemente, lo seguirá siendo a corto y medio plazo.

En este capítulo sólo hemos pretendido realizar una incursión superficial en el problema, centrándonos en la potencia expresiva de las integrales proyectivas y su capacidad para extraer la información que permite distinguir a unas personas de otras. Como resultado de nuestros experimentos, podemos establecer varias conclusiones generales:

- Con una selección adecuada de las regiones proyectadas, las integrales proyectivas son capaces de ofrecer una **representación compacta** y reducida de las caras, que conserva la información que caracteriza a los individuos a un nivel superior al conseguido con otras técnicas como las autocaras. Es más, las proyecciones mejoran de forma significativa la capacidad de **generalización** en relación al uso de las propias imágenes faciales.

- En comparación con otras técnicas de reducción a subespacios, las proyecciones conservan la **continuidad espacial** en los valores de las señales obtenidas. Esta propiedad permite el alineamiento de las señales, lo que ofrece robustez frente a ligeros cambios de escala, posición y valores de gris de las imágenes.

- En la construcción de un sistema de reconocimiento basado en integrales proyectivas, es posible y aconsejable utilizar **varias proyecciones** de las caras. La combinación de las mismas ofrece mejores resultados que su uso por separado. Sería muy interesante disponer de mecanismos automatizados para la selección de las regiones más representativas y la optimización de los parámetros de combinación.

- A lo largo de los experimentos, ha quedado comprobado que las proyecciones pueden aprovecharse para construir sistemas de reconocimiento **fiables**, **eficientes**, **invariantes** frente a expresiones faciales, con tolerancia a ligeras variaciones de iluminación y

pose, y todo ello para tamaños de galería relativamente grandes. El rendimiento mejorará de manera muy significativa si disponemos de más de una muestra por individuo. En el escenario "controlado", el rendimiento de las proyecciones se aproxima bastante al óptimo, pero con un **coste de entrenamiento** y de **clasificación** muy inferior al de otras técnicas que constituyen el estado del arte.

- Algunos otros aspectos quedan pendientes de un estudio más profundo para el caso del escenario "no controlado", como los **cambios de iluminación**, las grandes diferencias en las **fechas de captura**, y el reconocimiento **no frontal** de caras. Estos inconvenientes no son exclusivos de las proyecciones, sino que constituyen las grandes cuestiones abiertas en el ámbito del reconocimiento facial de personas. Su resolución se presenta muy compleja, y especialmente cuando se dispone de una sola muestra de entrenamiento por individuo. Una vía prometedora es la aplicación de modelos deformables, que podría ser un paso previo a la obtención de las proyecciones.

CAPÍTULO 7

"El grito", E. Munch, 1893

Extracción de Información Facial

"Nunca he visto un rostro sonriente que no fuera bello."
H. JACKSON BROWN

La cara es el espejo del alma.
PROVERBIO POPULAR

Hasta este punto hemos podido comprobar extensamente cómo las proyecciones permiten resolver de forma robusta y eficiente muchos de los problemas básicos en el dominio del procesamiento de caras humanas. Pero el uso de esta técnica no se limita a tales problemas, sino que puede resultar de gran ayuda en aplicaciones como el análisis de expresiones faciales, la estimación de pose, la clasificación del sexo, los interfaces perceptuales, los sistemas de monitorización y video-vigilancia, y muchas otras.

La utilización *metódica* de las integrales proyectivas –es decir, basada en la descripción mediante modelos, la aplicación de los algoritmos de alineamiento y las funciones de distancia–, permite extraer más información de las caras que la que se deriva de un simple análisis heurístico de picos máximos y mínimos, o zonas de máxima variación de las señales. Además, seleccionando proyecciones asociadas a regiones concretas del rostro, se puede realizar un análisis pormenorizado de las diferentes partes de la cara.

En este capítulo vamos a describir dos aplicaciones de las técnicas de detección, localización y seguimiento de caras desarrolladas en los capítulos anteriores. De esta manera, nos apoyamos en los mecanismos ya diseñados, que nos permiten encontrar las caras que existen en una imagen, conocer la posición de ojos y boca, y seguir su movimiento a lo largo de una secuencia de vídeo. El interés ahora se centra en extraer información relevante del rostro, con el fin de crear sistemas capaces de responder a las interacciones faciales del usuario.

Más específicamente, hemos trabajado en dos aplicaciones de distinta índole: un sistema de generación de avatares, descrito en la sección 7.1; y un interface perceptual para la navegación en un entorno virtual, presentado en la sección 7.2. La primera aborda el análisis de

expresiones faciales, partiendo de un sistema simplificado de codificación de expresiones. La segunda está relaciona con la estimación de pose 3D de la cara, aunque más orientada a la "manejabilidad" del entorno que a la obtención precisa de ángulos y posiciones. En consecuencia, el propósito final de ambos programas es la experimentación y la ejemplificación del uso de proyecciones, más que el desarrollo de herramientas destinadas al usuario final. No obstante, el interface perceptual sí que está implementado para el uso público, y ha sido presentado en varias publicaciones [61, 64].

7.1. Análisis de expresiones faciales mediante proyecciones

En los capítulos previos hemos utilizado las integrales proyectivas para extraer características globales de la cara. Sin embargo, las proyecciones también pueden ser restringidas a componentes concretos del rostro, con el fin de analizarlos de forma separada. Partiendo de una cara correctamente localizada, es posible definir regiones específicas asociadas a cada elemento facial, principalmente ojos y boca. Las proyecciones obtenidas de tales regiones aportan información de enorme utilidad en la localización y análisis de los componentes.

Aplicando las bases establecidas en el capítulo 2, en esta sección detallamos el diseño de un método sencillo de análisis de expresiones faciales con proyecciones. En primer lugar, en el apartado 7.1.1, definimos un sistema simplificado de codificación de las expresiones faciales, introduciendo un conjunto reducido de unidades de activación asociadas a los ojos y la boca. A continuación, exponemos la forma de modelar y detectar las diversas unidades mediante proyecciones, en el apartado 7.1.2. El apartado 7.1.3 describe la aplicación del proceso en la construcción de un sistema de generación de avatares. Para acabar, en el apartado 7.1.4 se presentan los resultados experimentales, y en el 7.1.5 se extraen las conclusiones finales.

7.1.1. Sistema de codificación de las expresiones faciales

Gran parte de las técnicas existentes para el análisis de expresiones faciales se basan en la distinción de un conjunto discreto y reducido de posibles estados. El sistema clásico de codificación de expresiones es FACS (*Facial Action Coding System*) propuesto por Ekman y Friesen en 1978 [48]. Este estándar define 44 *unidades de activación* (AU), que pueden aparecer solas o combinadas. Cada una de ellas representa un posible estado de una parte de la cara. Por ejemplo, AU1 es "interior de las cejas levantado", AU2 es "exterior de las cejas levantado", AU19 es "enseñando la lengua", etc. Otras propuestas más recientes, como el modelo Candide [1], o el estándar de codificación definido en MPEG-4 [133], permiten diferentes niveles de graduación de las expresiones y los gestos del individuo.

En esta sección no pretendemos hacer un análisis tan exhaustivo de las expresiones, sino que nuestro objetivo final es construir una aplicación sencilla de generación automática de avatares. Por lo tanto, vamos a introducir un sistema de codificación mucho más reducido, compuesto exclusivamente por cuatro unidades de activación por cada componente.

El sistema que proponemos tiene las siguientes características:

1. El análisis de la expresión facial se descompone en el estudio del **ojo izquierdo**, el **ojo derecho** y la **boca** que, en principio, se tratan de forma independiente.

2. Para cada uno de los componentes de interés, se definen **cuatro unidades de activación**, que se muestran en la figura 7.1.

Figura 7.1: *Definición de las unidades de activación en el sistema simplificado. Para cada una se muestra el código de la unidad, un ejemplo y una breve descripción.*

3. Las unidades de activación asociadas a un mismo componente son **incompatibles** entre sí. En otras palabras, cada componente facial estará asociado en todo momento a una, y sólo una, de las unidades correspondientes.

4. También son incompatibles los pares: (OI1, OD3), (OI1, OD4), (OI2, OD3), (OI2, OD4), (OI3, OD1), (OI3, OD2), (OI3, OD4), (OI4, OD1), (OI4, OD2), y (OI4, OD3). Estas incompatibilidades surgen de la propia dificultad o **limitación anatómica** para producir tales combinaciones.

La restricción número 1 de nuestro sistema de expresiones faciales garantiza que los componentes se deben analizar por separado. De esta forma, cada uno tendrá definida una región específica en el modelo estándar de cara. Por su parte, de las restricciones 2 y 3 se deduce que el objetivo consiste en asignar a cada una de esas regiones uno de los cuatro estados posibles. Finalmente, la número 4 implica que debe haber una comprobación adicional de consistencia entre los estados de los ojos. Básicamente, se permite que ambos ojos tengan el mismo estado o que uno esté abierto y el otro cerrado. Las demás combinaciones son eliminadas.

Los dos últimos puntos introducen una simplificación adicional del sistema de codificación. Algunas de las incompatibilidades propuestas pueden ser más o menos discutibles.

Por ejemplo, es posible que un ojo esté a la vez cerrado y con las cejas levantadas; o pueden aparecer los dientes con la boca entreabierta. Sin embargo, debemos reiterar que nuestro propósito es desarrollar una técnica básica; ésta podrá ser aumentada posteriormente añadiendo nuevas unidades de activación o modificando las incompatibilidades definidas.

7.1.2. Modelado y detección de las unidades de activación

En un sistema de codificación como el definido, las proyecciones pueden suponer una gran ayuda para el análisis de la expresión facial. Varias características apoyan esta decisión: en primer lugar, existen regiones claramente delimitadas que son el objetivo del análisis; en segundo lugar, el número de estados de cada componente es finito; y en tercer lugar, por simple inspección visual vemos que la distinción de estados por niveles de gris es factible.

En definitiva, la idea básica del método propuesto consiste en aplicar proyecciones sobre partes predefinidas de las caras. Para cada una, dispondremos de un conjunto de ejemplos de entrenamiento asociados a las unidades de activación. En consecuencia, el análisis de expresiones se reduce a un problema de clasificación de patrones con señales unidimensionales.

Regiones asociadas a los componentes

Sobre un modelo estándar de cara es posible acotar las zonas en las que, a priori, se espera encontrar cada componente facial. La figura 7.2 muestra las regiones que proponemos utilizar para el ojo izquierdo, el derecho y la boca, en proporción a nuestro modelo de cara, descrito en el apartado 2.2.5 del capítulo 2 (ver la página 61).

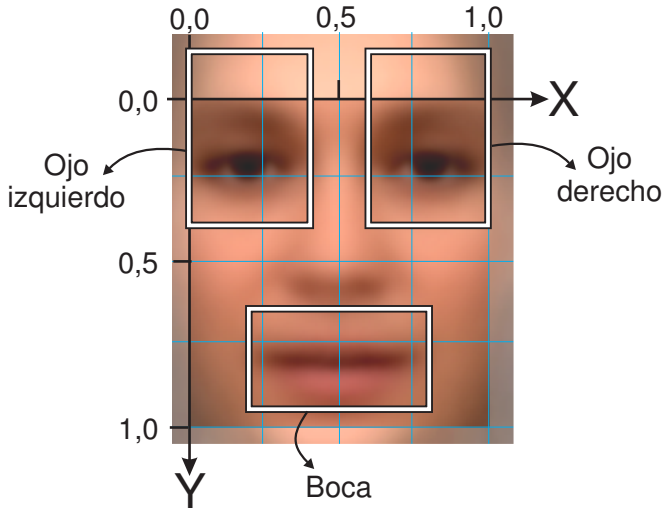


Figura 7.2: *Regiones utilizadas para cada componente facial. El sistema de coordenadas está en proporción al modelo de cara. Se señalan en blanco las regiones de ojo izquierdo, ojo derecho y boca.*

Los límites concretos de las regiones definidas son los siguientes:

- **Ojo izquierdo (ojo1)**. Extensión en X: (0; 0,4). Extensión en Y: (-0,15; 0,4).

- **Ojo derecho (ojo2)**. Extensión en X: (0,6; 1). Extensión en Y: (-0,15; 0,4).
- **Boca**. Extensión en X: (0,2; 0,8). Extensión en Y: (0,65; 0,95).

El algoritmo de reconocimiento de expresiones parte de una cara correctamente localizada o seguida a lo largo de una secuencia de vídeo. Utilizando una transformación afín, el rostro es extraído a un tamaño y posición estándar, como se describe en el apartado 4.3.2 (página 191). Esta imagen rectificada y escalada es la entrada del analizador de expresiones. Para cada una de las regiones definidas, se calculan las proyecciones verticales, que llamaremos PV_{ojo1}, PV_{ojo2} y PV_{boca}, asociadas al ojo izquierdo, el derecho y la boca, respectivamente. En la figura 7.3 se representa gráficamente esta etapa inicial del analizador.

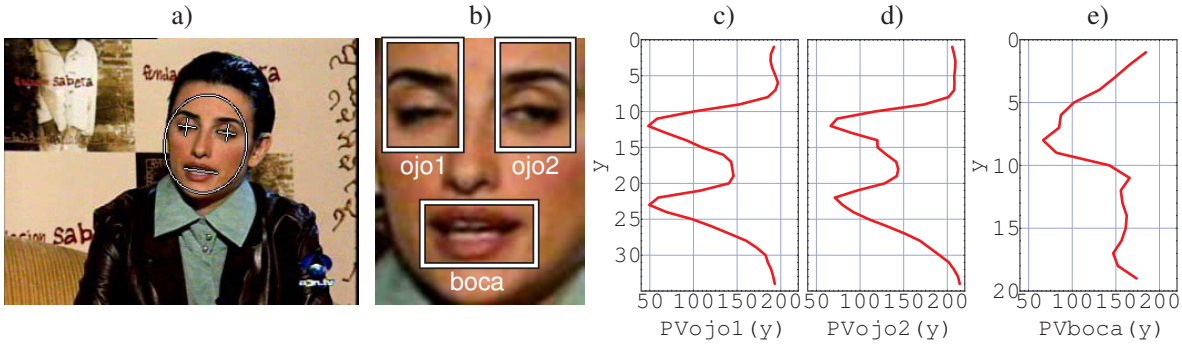


Figura 7.3: *Proceso de análisis de expresiones faciales. a) Usando una técnica de seguimiento, se localiza la cara a lo largo de la secuencia. b) La cara localizada se extrae a un tamaño y posición predefinidos mediante una transformación afín. c-e) Sobre las regiones de ojo izquierdo, derecho y boca se aplican proyecciones verticales para obtener PV_{ojo1}, PV_{ojo2} y PV_{boca}, respectivamente.*

Por omisión, las proyecciones de los ojos tienen tamaño 34 y las de la boca 19 puntos.

Modelado y clasificación de las unidades de activación

Ya en el capítulo 6 planteamos el reconocimiento de personas como un problema de clasificación de proyecciones. También el análisis de expresiones faciales puede ser resuelto de forma similar. Pero hay una diferencia fundamental entre ambos: en el primero existen muchas clases y muy pocos ejemplos por clase; en el segundo se invierte la situación, pocas clases y muchos ejemplos de cada una. Esto permite aprovechar diferentes métodos de clasificación, como vecino más próximo, *k*-medias, máquinas de vectores de soporte, redes neuronales, modelos de Markov, redes bayesianas, etc.

En una fase previa de aprendizaje, el mecanismo es entrenado con un conjunto de ejemplos etiquetados adecuadamente. Después, dada una nueva instancia, se usa el clasificador para reconocer la unidad de activación más probable. En todos los casos, la entrada está constituida por las proyecciones verticales de los componentes, como las mostradas en la figura 7.3, donde cada proyección puede ser vista como un vector en $\mathbb{R}^k$.

Sin ánimo de ser exhaustivos, proponemos los siguientes métodos de clasificación (algunos de ellos ya los vimos en el apartado 6.3.2 del anterior capítulo):

- **Clasificación mediante distancia a la media**. Para cada clase se calcula un modelo de proyección media/varianza, como los que se muestran en la figura 7.4. La distancia de un ejemplo a una clase se obtiene tras el alineamiento señal/modelo del algoritmo 2.4.

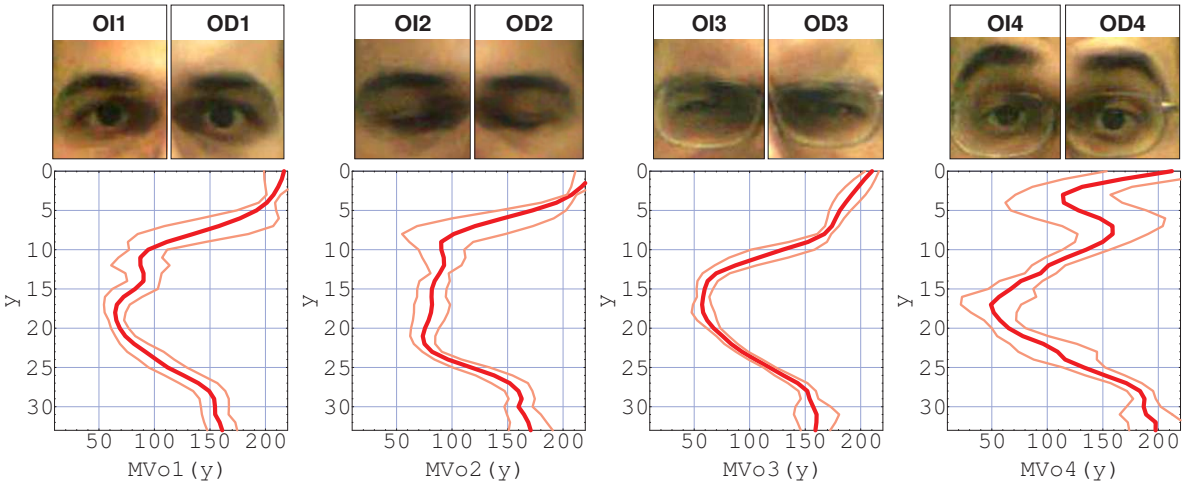


Figura 7.4: *Modelos media/varianza asociados a diferentes unidades de activación de los ojos. Los modelos son iguales para el ojo izquierdo y para el derecho. Arriba se muestran dos ejemplos y abajo los modelos de proyección vertical asociados. Las líneas finas denotan la varianza en cada punto.*

Este método tiene más sentido cuando los ejemplos de entrenamiento proceden del mismo individuo. Así, aunque las proyecciones para distintas personas tienen formas parecidas, en general el uso de un modelo medio parece poco viable con múltiples usuarios. Por ejemplo, el contraste entre una persona con cejas espesas y otra con cejas finas es difícil de modelar con una simple proyección media.

Por otro lado, como explicamos en el problema de reconocimiento, puede suceder que la varianza no modele adecuadamente la variabilidad relativa entre las diferentes clases, al estar muy influida por las proyecciones de entrenamiento. Por ejemplo, el modelo MV_{O4} de la figura 7.4 presenta varianzas mucho mayores que MV_{O3}, de manera que tenderá a producir siempre menores diferencias. Podemos prescindir de la varianza para conseguir puntuaciones comparables en todas las clases.

- **Clasificación mediante vecino más próximo**. El entrenamiento en este caso es inexistente. Simplemente se trataría de recolectar un conjunto de proyecciones asociadas a cada clase. El resultado de la clasificación es la clase del ejemplo con menor distancia (tras el alineamiento) respecto de la señal de entrada. En la figura 7.5 se representa el espacio de las integrales proyectivas, con algunos ejemplos de entrenamiento asociados a los estados de ojo. En adelante utilizaremos OX*i* para denotar los lugares donde pueden aparecer indistintamente las unidades de activación de ojo izquierdo y derecho, es decir, los pares OI*i* y OD*i*. Puesto que ambos ojos son simétricos, la proyección vertical es igual para los dos, y sólo se requiere un clasificador de ojos.

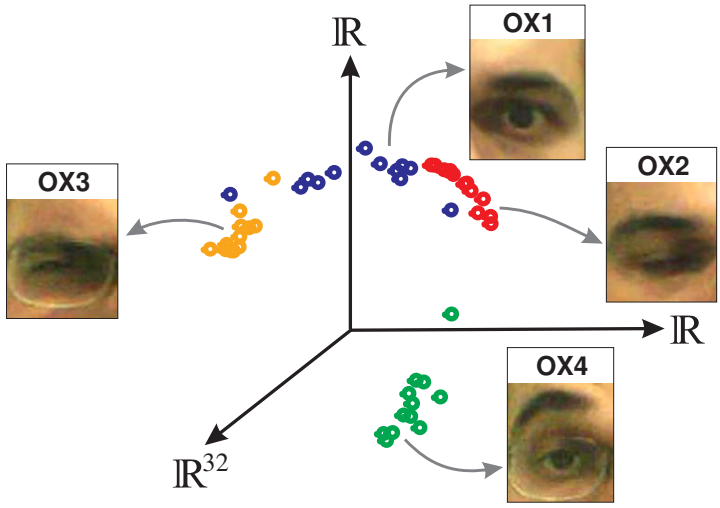


Figura 7.5: *Ejemplos de entrenamiento de los estados de activación de los ojos (tanto izquierdo como derecho). Cada ejemplo es un vector en $\mathbb{R}^{34}$. La representación ha sido obtenida mediante proyección en las dos direcciones principales (las de mayor autovalor asociado).*

El método de vecino más próximo resulta más adecuado cuando aparecen diferentes usuarios y condiciones de iluminación. Sin embargo, tiene el inconveniente de que el conjunto de entrada debe ser suficientemente representativo de las distintas situaciones posibles. La dependencia con los ejemplos de entrenamiento es mucho mayor, lo cual implica un posible riesgo de sobreajuste y reducida capacidad de generalización.

- **Clasificación mediante *k* medias.** La clasificación mediante k medias intenta disminuir la dependencia en relación con los ejemplos disponibles, a través de una agrupación y simplificación del conjunto de entrenamiento. Para ello se aplica un algoritmo de k medias [68], seleccionando un número adecuado de grupos (en inglés, *clusters*).

  El proceso de clasificación sería parecido al método de vecino más próximo, pero usando ahora las k proyecciones resultantes por clase. Es más, podríamos crear k modelos de proyección media/varianza para describir de manera más refinada la distribución de cada clase. El resultado es que la función de densidad de probabilidad condicionada de las clases se modela mediante modelos de mezcla de gaussianas.

  La figura 7.6a) muestra un ejemplo de esta idea, donde se ha utilizado $k = 2$. De esta forma, la distribución de cada clase es descrita con dos gaussianas.

- **Clasificación mediante máquinas de vectores de soporte (SVM).** Respecto a los otros mecanismos de clasificación, las máquinas de vectores de soporte suponen normalmente una mejora de las capacidades de generalización. En nuestro caso, los vectores son las proyecciones asociadas a cada clase.

  La idea del método consiste en seleccionar un conjunto reducido de ejemplos que mejor modelen las fronteras entre las diferentes clases, como se ilustra en la figura 7.6b). Para ello se usa un criterio de *minimización del riesgo de error estructural* [185]. En concreto, se

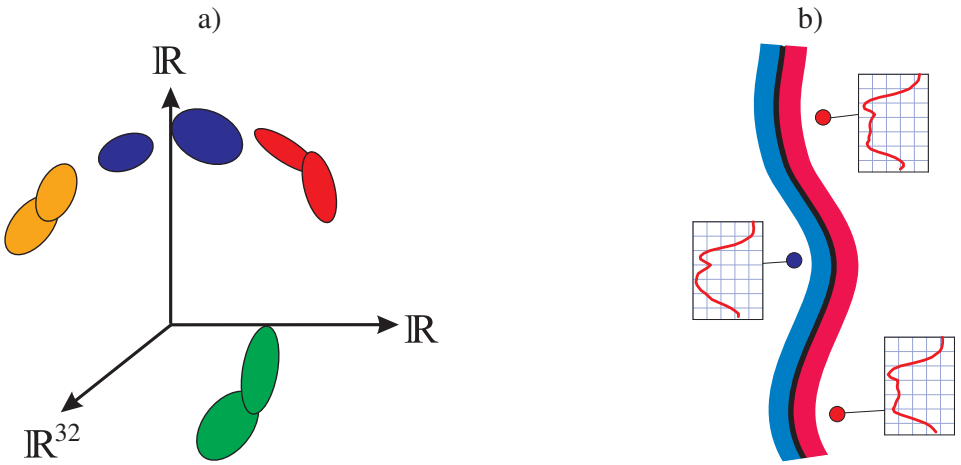


Figura 7.6: *Clasificación de proyecciones mediante k medias y SVM. a) En el algoritmo de k medias, se realiza una partición de los ejemplos de cada clase en k grupos. Para cada uno se crea un modelo gaussiano de media/varianza. b) En SVM se seleccionan los ejemplos que mejor delimitan la frontera entre clases, produciendo una máxima separación. Se usan los mismos colores para las clases que en la figura 7.5.*

busca el hiperplano que produzca una mayor separación entre clases, trabajando implícitamente en un espacio de mayor dimensionalidad que el de entrada.

Las funciones de *kernel* clásicas (polinomial, RBF, etc.) no aprovechan el alineamiento entre proyecciones. Sería interesante explorar la definición de un *kernel* en el que interviniera este algoritmo, haciendo así uso de las ventajas de ambos mecanismos: la gran capacidad de generalización de SVM, y la potencia de los métodos de alineamiento. Por limitaciones de espacio, no se ha llevado a cabo en el contexto de esta tesis.

Sea cual sea el método elegido, se deben construir dos clasificadores, uno para los ojos y otro para la boca. Ambos constan de cuatro clases, asociadas a las unidades de activación: OX1, OX2, OX3 y OX4 para el clasificador de ojos; y B1, B2, B3 y B4 para el de la boca.

Detección de las unidades de activación

Una vez seleccionados y entrenados los mecanismos de clasificación, las proyecciones verticales calculadas para la nueva cara son sometidas al clasificador correspondiente, dando lugar a la *verosimilitud* de cada unidad de activación; en nuestro caso, el resultado son *distancias*. Pero ambas medidas son, en el fondo, análogas, ya que es posible derivar probabilidades de pertenencia a partir de las distancias a las clases.

Sea d la función que expresa las distancias de las unidades de activación con las proyecciones dadas; por ejemplo, tenemos $d(\mathrm{OI1}, PV_{ojo1})$, $d(\mathrm{OD2}, PV_{ojo2})$, etc. Puesto que cada unidad está asociada a un tipo de proyección de forma unívoca, podemos simplificar la notación de la función d, escribiendo sencillamente $d'(\mathrm{OI1})$, $d'(\mathrm{OD2})$, etc.

Las restricciones introducidas en el sistema de expresiones imponen el hecho de que las unidades (OI3, OD3) y (OI4, OD4) deben aparecer siempre juntas, es decir, ambas cejas deben

estar subidas o bajadas al mismo tiempo. Una manera de garantizar este resultado es tomar la distancia media para ambos ojos, es decir:

$$d'(\text{OI3}) = d'(\text{OD3}) = \frac{d(\text{OI3}, PV_{ojo1}) + d(\text{OD3}, PV_{ojo2})}{2} \tag{7.1}$$

$$d'(\text{OI4}) = d'(\text{OD4}) = \frac{d(\text{OI4}, PV_{ojo1}) + d(\text{OD4}, PV_{ojo2})}{2}$$

Finalmente, a partir de las distancias podemos conseguir probabilidades de aparición de cada unidad de activación, U, que denotamos por $p(U)$, utilizando un mecanismo similar a las funciones *softmax* aplicadas típicamente en redes neuronales:

$$p(U) = \frac{\exp(-d'(U))}{\sum_{\forall V \in Grupo(U)} \exp(-d'(V))} \tag{7.2}$$

donde $Grupo(U)$ son todas las unidades definidas para el mismo componente facial que U. La función p puede interpretarse como una probabilidad, puesto que los valores están entre 0 y 1, y está normalizada. La decisión se obtiene aplicando el criterio de máxima verosimilitud:

$$\widehat{Clase(U)} = \text{argmáx}_{\forall V \in Grupo(U)} p(V) \tag{7.3}$$

Adicionalmente, las probabilidades pueden servir para conocer el grado de fiabilidad del resultado o la existencia de una expresión intermedia entre dos de las predefinidas. Vamos a ver un ejemplo de aplicación de esta técnica en un sistema de creación de avatares.

7.1.3. Experimentación y aplicación en generación de avatares

La generación automática de avatares animados es una de las posibles aplicaciones del análisis de expresiones faciales. Un usuario de un sistema de videoconferencia puede desear que sus interlocutores observen sus gestos y expresiones faciales, pero sin dar a conocer su identidad. El uso de avatares es una versión simplificada de la *síntesis de expresiones faciales*, donde se pretende trasladar todos los detalles de un gesto a otro rostro base [108, capítulo 12], o a un modelo 3D [20]. En nuestro caso, la generación del resultado consiste simplemente en componer los trozos de imagen asociados a las expresiones con mayor verosimilitud.

Estructura global de la aplicación

En la figura 7.7 se representa a grandes rasgos la estructura de la aplicación desarrollada. El sistema trabaja con una cara localizada, dos clasificadores entrenados (de ojos y boca), y dos arrays de imágenes, i_{ojos} e i_{boca}, para producir el icono resultante.

Los pasos del proceso de análisis y generación del avatar son los siguientes:

1. Aplicar los algoritmos de detección, localización y seguimiento de caras a lo largo de la secuencia de vídeo, como describimos en los capítulos 3, 4 y 5, respectivamente. La

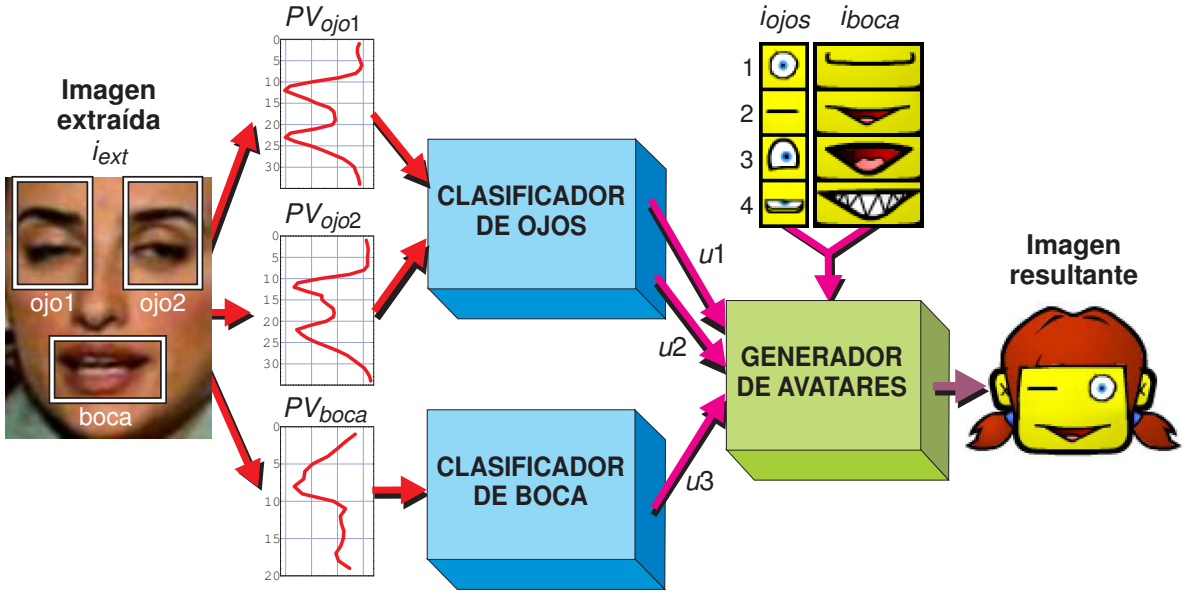


Figura 7.7: *Estructura global del generador de avatares. A partir de la cara extraída del seguimiento, se calculan las proyecciones verticales PV_{ojo1}, PV_{ojo2} y PV_{boca}. Estas señales se pasan a los clasificadores entrenados de ojos y boca. Las clases resultantes, $u1$, $u2$ y $u3$, y las imágenes asociadas a cada posible unidad, i_{ojos} e i_{boca}, son la entrada para el generador propiamente dicho. El resultado es la imagen iconográfica que representa al usuario, el avatar.*

ejecución del detector se ajusta para encontrar una sola cara en las imágenes.

2. Para cada nueva imagen, i, de la secuencia, extraer la cara a una posición estándar mediante una transformación afín, i_{ext}, como se explicó en el apartado 4.3.2.

3. Calcular las proyecciones verticales de las zonas de ojo izquierdo, derecho y boca (especificadas en el apartado 7.1.2) sobre la imagen extraída i_{ext}, obteniendo PV_{ojo1}, PV_{ojo2} y PV_{boca}.

4. Clasificar las proyecciones PV_{ojo1}, PV_{ojo2} y PV_{boca}. Existirá un clasificador entrenado para los ojos y otro para la boca. Si el resultado son distancias, calcular a partir de ellas las probabilidades de cada unidad de activación $p(\text{OI1})$, $p(\text{OI2})$, $p(\text{OD1})$, etc.

5. Seleccionar las unidades más probables:

   $u1 = \arg\max_{i=1,..,4} p(\text{OI}i)$; $u2 = \arg\max_{i=1,..,4} p(\text{OD}i)$; $u3 = \arg\max_{i=1,..,4} p(\text{B}i)$

6. Crear la imagen del avatar usando los fragmentos $i_{ojos}[u1]$ para el ojo izquierdo, $i_{ojos}[u2]$ para el derecho, e $i_{boca}[u3]$ para la boca.

La decisión del mecanismo de clasificación a aplicar se encierra en el paso 4. En principio, se podría usar cualquiera de las técnicas introducidas en el apartado 7.1.2 u otras más avanzadas. En concreto, hemos implementado los métodos de distancia a la media, vecino más próximo y k medias, cuyos resultados analizamos a continuación.

Aspectos de implementación y descripción de las pruebas

La figura 7.8 muestra el aspecto gráfico del prototipo desarrollado. El programa maneja las librerías de detección, localización y seguimiento documentadas en los capítulos anteriores, de manera que se puede aprovechar cualquiera de las técnicas disponibles. Igualmente, se utiliza el entorno de programación Borland C++ Builder 6 y las librerías Intel OpenCV [35].

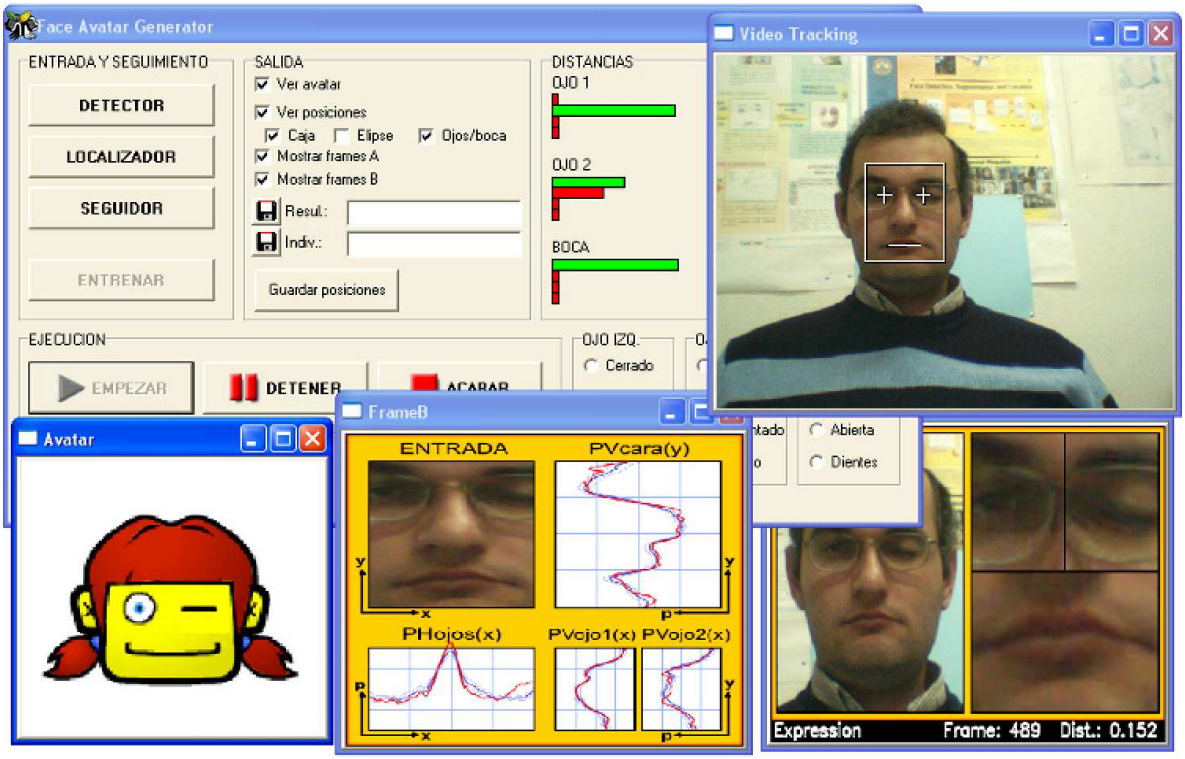


Figura 7.8: *Vista del prototipo de analizador de expresiones y generador de avatares. El programa incorpora todos los métodos de detección, localización y seguimiento de caras descritos en los capítulos previos. Además de la imagen resultante (abajo a la izquierda) se puede ver información adicional del proceso de seguimiento.*

Además de la generación de avatares, la aplicación permite entrenar los tres tipos de clasificadores mencionados a través de un etiquetado realizado de manera interactiva.

En un escenario típico, existirá un único usuario que maneja el programa y usa como entrada una cámara de videoconferencia de bajo coste, en interior o parcialmente iluminada con luz del exterior. En consecuencia, las pruebas se centran en estas condiciones de trabajo.

Más específicamente, se han grabado cuatro secuencias del mismo individuo. En las dos primeras ("exp.ggm0.avi" y "exp.ggm1.avi") se mantiene la iluminación interior; en la tercera ("exp.ggm2.avi") se cambia parcialmente a iluminación mixta; y en la cuarta ("exp.ggm3.avi") se usa luz solar indirecta, lo que supone una variación de apariencia mucho más drástica. En la figura 7.9 se puede ver un caso de cada una de estas condiciones.

En las pruebas documentadas más adelante se usa el método combinado Haar+IP para la detección del rostro; la localización aplica el algoritmo basado en proyecciones; y en el

a) b) c)

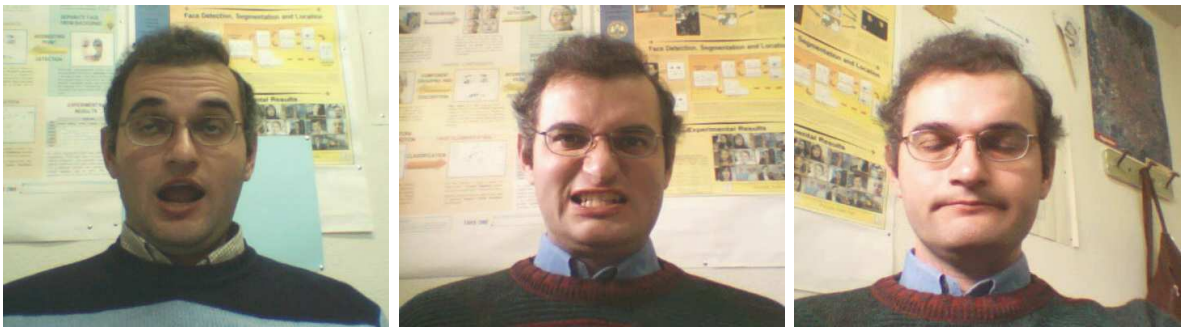


Figura 7.9: *Extractos de las secuencias de prueba del generador de avatares. El sistema de adquisición es una cámara de videoconferencia Creative Webcam NX Pro. La resolución es de 640 × 480 píxeles a 15 fps, con compresión DivX media/alta. a) Ejemplo del vídeo "exp.ggm1.avi", con iluminación interior, capturada el mismo día y con iguales condiciones que la secuencia de entrenamiento. Este vídeo consta de 1380 frames. b) Ejemplo de "exp.ggm2.avi", con luz mixta y tomada un día diferente, con 1872 frames. c) Ejemplo de "exp.ggm3.avi", con predominio de luz exterior, capturada el mismo día que la anterior. Contiene 935 frames.*

seguimiento de la secuencia se ejecuta el seguidor de caras mediante integrales proyectivas, en este caso con predictor nulo.

El entrenamiento ha sido realizado con algunas imágenes de la primera secuencia ("exp. ggm0.avi"). El resto de secuencias se aprovechan como casos de prueba. El número de ejemplos utilizado para el entrenamiento de cada unidad de activación es el siguiente:

OX1: 42	**OX2**: 28	**OX3**: 28	**OX4**: 32
B1: 37	**B2**: 27	**B3**: 29	**B4**: 24

A partir de estas proyecciones, se entrenan los clasificadores de distancia a la media, vecino más próximo y k medias, con $k = 4$. En total existen 6 clasificadores distintos, 3 para los ojos y otros 3 para la boca.

7.1.4. Resultados experimentales

Sobre cada *frame* analizado de las tres secuencias de prueba, se han aplicado los 6 clasificadores, contrastando los resultados con los de un etiquetado manual. El proceso se repite de 6 en 6 *frames*, con el fin de reducir la laboriosa tarea de etiquetado. El resultado son las denominadas **matrices de confusión**, que expresan las unidades de activación obtenidas en función de las etiquetas indicadas por el operador humano.

Las tablas 7.1, 7.2 y 7.3 contienen las matrices de confusión para el primer vídeo de prueba con los clasificadores de distancia a la media, vecino más próximo y k medias, respectivamente. Cada una contiene a su vez dos tablas, una para los ojos –los resultados de ambos están agrupados– y otra para la boca.

Veamos punto por punto los hechos más destacados que podemos extraer de estos datos:

1. **Valoración global.** De forma global, los resultados obtenidos son bastante positivos, encontrándose alrededor del 90 % de clasificación correcta. Estos porcentajes son acep-

Clasificación mediante distancia a la media, vídeo "exp.ggm1.avi"

Unidad etiquetada	Unidad resultante del clasificador OX1	OX2	OX3	OX4	Correctas/ Total
OX1	241 (90,6 %)	25 (9,4 %)	0 (0 %)	0 (0 %)	241/266
OX2	11 (12,2 %)	79 (87,8 %)	0 (0 %)	0 (0 %)	79/90
OX3	0 (0 %)	12 (20,7 %)	46 (79,3 %)	0 (0 %)	46/58
OX4	0 (0 %)	8 (16,7 %)	0 (0 %)	40 (83,3 %)	40/48
Corr./Total	241/252	79/124	46/46	40/40	**406/462 (87,9 %)**

Unidad etiquetada	Unidad resultante del clasificador B1	B2	B3	B4	Correctas/ Total
B1	131 (92,9 %)	0 (0 %)	7 (5 %)	3 (2,1 %)	131/141
B2	0 (0 %)	34 (100 %)	0 (0 %)	0 (0 %)	34/34
B3	0 (0 %)	0 (0 %)	40 (100 %)	0 (0 %)	40/40
B4	3 (18,8 %)	0 (0 %)	0 (0 %)	13 (81,3 %)	13/16
Corr./Total	131/134	34/34	40/47	13/16	**218/231 (94,4 %)**

Tabla 7.1: *Matrices de confusión del análisis de expresiones con el vídeo "exp.ggm1.avi" y clasificador de distancia a la media. Los datos indican el número de casos presentes de cada tipo. Los porcentajes están en relación al total de la fila. Arriba aparecen los resultados para los ojos (ambos agrupados) y abajo para la boca. En la celda inferior derecha de cada tabla se muestra el porcentaje total de clasificación correcta.*

Clasificación mediante vecino más próximo, vídeo "exp.ggm1.avi"

Unidad etiquetada	Unidad resultante del clasificador OX1	OX2	OX3	OX4	Correctas/ Total
OX1	257 (96,6 %)	9 (3,4 %)	0 (0 %)	0 (0 %)	257/266
OX2	11 (12,2 %)	77 (85,6 %)	2 (2,2 %)	0 (0 %)	77/90
OX3	6 (10,3 %)	0 (0 %)	52 (89,7 %)	0 (0 %)	52/58
OX4	2 (4,2 %)	0 (0 %)	6 (12,5 %)	40 (83,3 %)	40/48
Corr./Total	257/276	77/86	52/60	40/40	**426/462 (92,2 %)**

Unidad etiquetada	Unidad resultante del clasificador B1	B2	B3	B4	Correctas/ Total
B1	132 (93,6 %)	1 (0,7 %)	2 (1,4 %)	6 (4,3 %)	132/141
B2	1 (2,9 %)	25 (73,5 %)	5 (14,7 %)	3 (8,8 %)	25/34
B3	0 (0 %)	6 (15 %)	34 (85 %)	0 (0 %)	34/40
B4	1 (6,3 %)	0 (0 %)	0 (0 %)	15 (93,8 %)	15/16
Corr./Total	132/134	25/32	34/41	15/24	**206/231 (89,2 %)**

Tabla 7.2: *Matrices de confusión del análisis de expresiones con el vídeo "exp.ggm1.avi" y clasificador de vecino más próximo. Los datos indican el número de casos presentes de cada tipo. Los porcentajes están en relación al total de la fila. Arriba aparecen los resultados para los ojos (ambos agrupados) y abajo para la boca. En la celda inferior derecha de cada tabla se muestra el porcentaje total de clasificación correcta.*

tables para una aplicación como la propuesta, donde los fallos no resultan críticos. Hay que tener también en cuenta que en ciertos casos el error puede ser debido a la propia ambigüedad en el etiquetado manual; en muchas ocasiones la frontera para etiquetar uno u otro estado puede resultar difusa incluso para el operador humano. Un error del 5 % puede considerarse muy próximo al óptimo alcanzable de forma teórica.

Clasificación mediante *k* medias, vídeo "exp.ggm1.avi"

Unidad etiquetada	Unidad resultante del clasificador OX1	OX2	OX3	OX4	Correctas/ Total
OX1	244 (91,7 %)	22 (8,3 %)	0 (0 %)	0 (0 %)	244/266
OX2	10 (11,1 %)	80 (88,9 %)	0 (0 %)	0 (0 %)	80/90
OX3	1 (1,7 %)	5 (8,6 %)	52 (89,7 %)	0 (0 %)	52/58
OX4	3 (6,3 %)	3 (6,3 %)	2 (4,2 %)	40 (83,3 %)	40/48
Corr./Total	244/258	80/110	52/54	40/40	**416/462 (90 %)**

Unidad etiquetada	Unidad resultante del clasificador B1	B2	B3	B4	Correctas/ Total
B1	131 (92,9 %)	2 (1,4 %)	0 (0 %)	8 (5,7 %)	131/141
B2	4 (11,8 %)	18 (52,9 %)	8 (23,5 %)	4 (11,8 %)	18/34
B3	0 (0 %)	5 (12,5 %)	35 (87,5 %)	0 (0 %)	35/40
B4	0 (0 %)	1 (6,3 %)	0 (0 %)	15 (93,8 %)	15/16
Corr./Total	131/135	18/26	35/43	15/27	**199/231 (86,1 %)**

Tabla 7.3: *Matrices de confusión del análisis de expresiones con el vídeo "exp.ggm1.avi" y clasificador de k medias. Los datos indican el número de casos presentes de cada tipo. Los porcentajes están en relación al total de la fila. Arriba aparecen los resultados para los ojos (ambos agrupados) y abajo para la boca. En la celda inferior derecha de cada tabla se muestra el porcentaje total de clasificación correcta.*

2. **Probabilidades a priori.** Obviamente, no todas las unidades de activación aparecen con la misma frecuencia. El estado más común es una expresión neutra, con los ojos abiertos y la boca cerrada, como se puede apreciar en los totales acumulados. De esta observación se deduce que las *probabilidades a priori* de los diferentes estados no son uniformes, lo cual sugiere introducir una clasificación *bayesiana*, donde las probabilidades a posteriori dependan también de las estimadas a priori. Las técnicas que hemos propuesto no hacen uso de este hecho, a pesar de lo cual consiguen buenos resultados.

3. **Comparación entre métodos.** No existe un claro ganador entre los diferentes clasificadores. El método de distancia a la media logra los mejores resultados para la boca, y el de vecino más próximo para los ojos. Los porcentajes para *k* medias se encuentran siempre ligeramente por debajo. En cualquier caso, los tres se mueven en rangos comparables, posiblemente dentro del error de medición.

4. **Confusión entre estados.** En general, las mayores fuentes de confusión se encuentran entre las unidades que presentan cierta graduación. Por ejemplo, en el caso de los ojos ocurre con la distinción entre ojo abierto/cerrado, y en la boca con los estados abierta/entreabierta. Esto es debido, como ya se ha mencionado, a la ambigüedad implícita de ciertas expresiones, que afecta no sólo al etiquetado manual sino también a los clasificadores. Pero estos fallos no son importantes si ocurren dentro de una transición abrir/cerrar; simplemente se variaría el instante en el que el avatar cambia su apariencia. De alguna forma, todo esto aconseja usar las probabilidades para generar las imágenes según los grados encontrados y no sólo de forma discreta.

5. **Otras fuentes de error.** Existen otros errores no atribuibles a la anterior categoría, aunque

en proporción bastante inferior. Por ejemplo, uno de estos casos es la confusión boca cerrada/enseñando dientes, que aparece en todos los métodos. Estos errores están causados principalmente por las imprecisiones de localización, puesto que en todas las secuencias de prueba existen variaciones de posición y orientación de las caras.

6. **Eficiencia computacional.** En relación a los tiempos de ejecución, todos los métodos son extremadamente rápidos. Es fácil observar que las operaciones requeridas no son excesivamente costosas en ningún caso (proyección, alineamiento, comparación, etc.). En un ordenador con procesador Pentium IV a 2,60GHz, el reconocedor de expresiones con distancia a la media tarda en promedio unos 0,6 ms. Las otras técnicas son ligeramente más lentas. El clasificador de k medias tarda 0,9 ms y el de vecino más próximo 1,7 ms. Estos tiempos incluyen todo el proceso, excepto la generación del avatar.

Con el propósito de estudiar la evolución temporal del análisis de expresiones, en la figura 7.10 presentamos los resultados de la secuencia "exp.ggm1.avi", con las probabilidades obtenidas a lo largo del tiempo para el clasificador de distancia a la media.

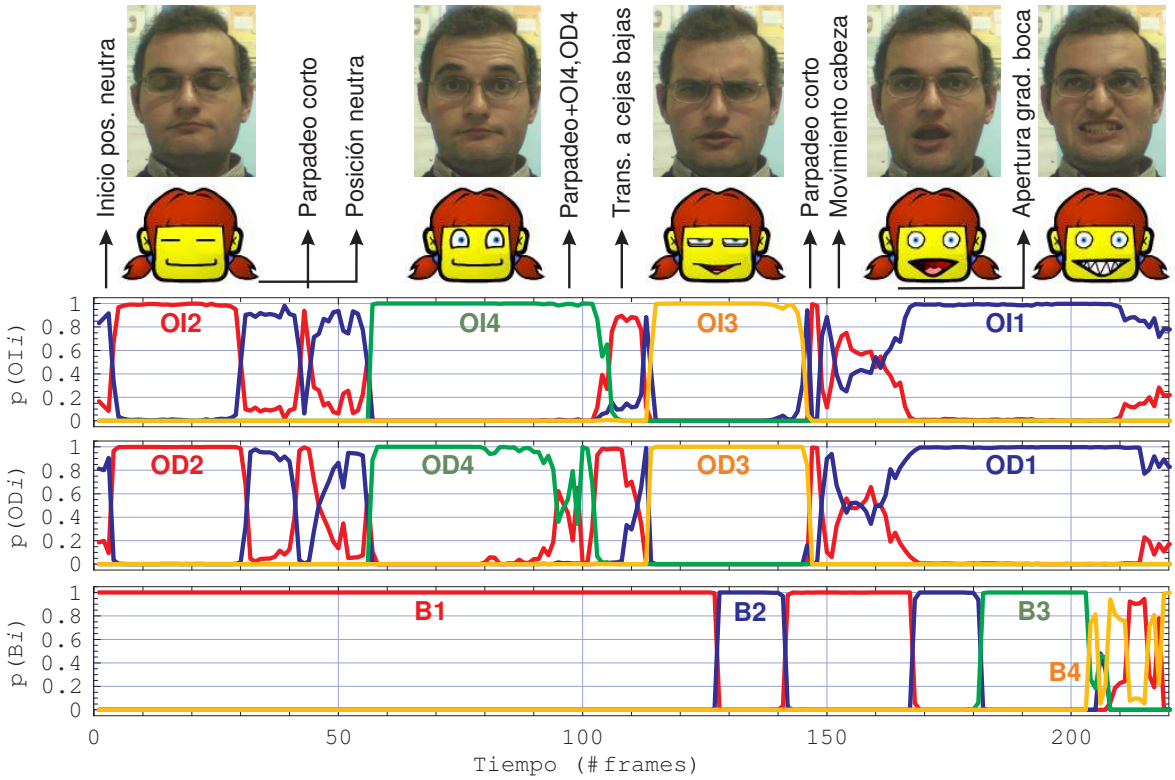


Figura 7.10: *Evolución del análisis de expresiones faciales en una secuencia de vídeo. Se utiliza el clasificador de distancia a la media sobre el vídeo "exp.ggm1.avi". En la parte superior aparecen extractos de las caras, los avatares generados, y algunos comentarios de las situaciones más notables, aproximadamente en los puntos donde suceden. De arriba abajo, las gráficas corresponden a las probabilidades para las unidades del ojo izquierdo ($p(OI1)$, $p(OI2)$, $p(OI3)$ y $p(OI4)$), el derecho ($p(OD1)$, $p(OD2)$, $p(OD3)$ y $p(OD4)$) y la boca ($p(B1)$, $p(B2)$, $p(B3)$ y $p(B4)$).*

Por un lado, podemos ver que en los periodos donde la expresión no varía las probabili-

dades se mantienen estables, aun cuando ocurren ligeros cambios de posición y orientación en toda la secuencia. Por otro lado, en la transición entre estados las variaciones son normalmente progresivas; una señal baja a medida que la otra sube, a lo largo de 3 ó 4 *frames*. Esto apoya la idea de usar las probabilidades como una estimación de una situación intermedia entre dos unidades de activación. Sólo al final de la secuencia ocurre una cierta inestabilidad de los resultados, con una expresión enseñando los dientes. En parte, es debida a un pequeño desajuste del seguimiento, que tiende a subir la posición de la boca. Durante un tramo cercano al *frame* 215, se obtiene un valor erróneo para la boca, que se clasifica como si estuviera cerrada.

El mismo proceso de evaluación aplicado sobre el primer vídeo de prueba ha sido repetido también para las dos secuencias restantes, en las que cambian de manera significativa las condiciones de iluminación, como se puede apreciar en la figura 7.9. Sin embargo, no se ha modificado el entrenamiento, que fue realizado con luz de interior. De esta forma pretendemos medir la capacidad de generalización del método, en cuanto a su robustez frente a los cambios en las fuentes luminosas.

Para evitar el exceso de información, presentamos en la tabla 7.4 los porcentajes finales de clasificación correcta para los tres vídeos de prueba con los tres métodos disponibles.

Secuencia	**Distancia a la media**			**Vecino más próx.**			***k* Medias**		
de prueba	**Ojos**	**Boca**	**Total**	**Ojos**	**Boca**	**Total**	**Ojos**	**Boca**	**Total**
exp.ggm1.avi	87,9	94,4	90,0	92,9	89,2	**91,2**	90,0	96,1	88,7
exp.ggm2.avi	92,4	64,2	**83,0**	92,1	51,6	78,6	88,6	37,7	71,6
exp.ggm3.avi	86,9	83,3	**85,7**	81,7	71,1	78,2	86,9	73,1	82,3

Tabla 7.4: *Resultados totales del análisis de expresiones faciales con los 3 vídeos de prueba y los 3 tipos de clasificadores. Los números indican los porcentajes de clasificación correcta para los ojos (ambos agrupados), para la boca y el total. En negrita se señalan los mejores ratios totales para cada vídeo.*

Hacemos algunas valoraciones sobre estos nuevos resultados:

1. **Valoración global.** Como cabía esperar, los porcentajes disminuyen al cambiar la iluminación de la escena. No obstante, se siguen encontrando en la mayoría de los casos en rangos aceptables. La reducción media total es de 10 puntos. Sólo en un caso (clasificación de la boca con *k* medias en el segundo vídeo) el porcentaje baja del 50 %.

2. **Robustez en ojos y boca.** La pérdida de efectividad es más destacada en la boca que en los ojos. Mientras que en los ojos la reducción media es únicamente del 2 %, para la boca supera el 26 %. Podemos afirmar que la variación de apariencia es mucho más destacada en la zona de la boca debido, posiblemente, a la aparición de sombras. Por su parte, la modificación en las proyecciones de ojos se limita principalmente a un cambio de intensidad, que es compensado adecuadamente en la técnica propuesta.

   Si comparamos las dos últimas secuencias entre sí, es curioso que en la segunda se clasifiquen mejor los ojos, y en la tercera mejor las bocas. Esto significa que la variación en las fuentes de luz no afecta por igual a todos los componentes faciales. Ciertos cambios

se pueden manejar correctamente y otros producen un número muy elevado de fallos; véanse, por ejemplo, los resultados para la boca en el segundo vídeo.

3. **Generalización de los clasificadores.** El método de distancia a la media supera sensiblemente a los otros dos clasificadores, demostrando una mayor capacidad de generalización. Su reducción es de 7 puntos en el segundo vídeo y menos de 5 para el tercero. Es más, los resultados son mejores si nos fijamos sólo en la clasificación de los ojos.

   Es significativo el caso de vecino más próximo, que conseguía los mejores resultados para el primer vídeo, pero se muestra inadecuado para las otras secuencias. En este dato influye el hecho de que en el entrenamiento no se dispone de ningún ejemplo con iluminación exterior. Claramente, el método mejoraría usando unos ejemplos de entrenamiento específicos de esa situación.

4. **Fuentes de confusión.** En cuanto a las matrices de confusión obtenidas –aunque no sean mostradas aquí–, podemos decir que aumenta el número de confusiones "no justificables", es decir, las que no proceden de una ambigüedad en la expresión. Destaca por encima de todas la asignación del estado "enseñando dientes" cuando realmente la boca está cerrada. Entre todas las técnicas y todos los casos, aproximadamente el 25 % de las imágenes con la boca cerrada se clasifican erróneamente con la unidad B4. No hay un argumento objetivo para explicar este problema. Es simplemente uno de los efectos colaterales del cambio de apariencia por la iluminación. Con mucha seguridad, el uso de probabilidades a priori podría aliviar este inconveniente.

7.1.5. Conclusiones finales

Pensamos que ha quedado suficientemente comprobado que el método propuesto alcanza los **requisitos básicos** en la aplicación objeto de estudio: es sencillo de implementar, bastante estable, tiene una adecuada robustez frente a cambios de iluminación, resulta muy eficiente computacionalmente, y puede ser entrenado con relativa facilidad.

Aunque el sistema presentado contiene un número reducido de estados, resulta evidente que el método es fácilmente **extensible**. Cada nueva unidad de activación considerada implicaría añadir una clase más a los mecanismos de clasificación. Lógicamente, a medida que aumente el número de clases se pueden incrementar también los errores. Pero esta situación se puede paliar de varias formas que ya hemos mencionado: añadiendo información sobre las probabilidades a priori de cada estado, y considerando las probabilidades como medidas de graduación entre estados. También se podrían añadir proyecciones horizontales o en otros ángulos de interés, o proyección de las imágenes de bordes. Otra idea aplicable en vídeo consiste en tener en cuenta la **evolución temporal** de los estados, que puede aportar una información difícil de deducir con imágenes estáticas.

Siendo más exigentes, sería interesante investigar mecanismos de clasificación más complejos, que permitieran construir un sistema capaz de trabajar con diferentes usuarios. Otra

limitación del método propuesto es la dificultad para analizar detalles finos, como las denominadas "características transitorias" [108, capítulo 11], por ejemplo, las arrugas asociadas a las expresiones. A mayor nivel de detalle se requerirá usar más información, y el uso de modelos deformables 2D o 3D puede ser más recomendable para esas situaciones.

7.2. Estimación de pose mediante proyecciones

Estimar la posición y orientación 3D del rostro es otro de los grandes problemas de interés en análisis de caras. Los resultados pueden tener utilidad directa en un número de aplicaciones, como en interfaces perceptuales, monitorización de personas y ayuda a discapacitados, por sí solos o en conjunción con el análisis de expresiones faciales.

La utilización de integrales proyectivas en la estimación de pose puede parecer un tanto paradójica: proyectamos imágenes 2D a señales 1D con el fin de extraer información 3D. Pero es bien conocido que muchos problemas de naturaleza tridimensional se pueden resolver sin utilizar modelos 3D explícitos. Un ejemplo es la *interpolación de vistas tridimensionales*: a partir de tres vistas de un objeto 3D rígido –y suponiendo conocidos los puntos correspondientes en ellas–, obtener la vista interpolada en cualquier posición intermedia. Está comprobado, [184], que en condiciones afines –o, equivalentemente, con *perspectiva débil*– la obtención de las vistas interpoladas se puede conseguir mediante combinaciones lineales, sin necesidad de manejar información tridimensional de forma explícita.

En las propuestas desarrolladas en esta sección se utilizan los métodos de detección, localización y seguimiento de caras descritos en los capítulos anteriores. Entre otras cosas, esto implica que el rango de giros permitidos será el asociado al método de seguimiento utilizado. Debemos aclarar que nuestro objetivo aquí es analizar una cara con posiciones dadas, no definir un nuevo mecanismo de relocalización o seguimiento 3D.

En concreto, vamos a describir dos técnicas alternativas en los apartados 7.2.1 y 7.2.2. La primera de ellas está basada en un conjunto de heurísticas *ad hoc*, definidas sobre la proyección vertical de la cara y la proyección horizontal de los ojos. El propósito no es determinar exactamente los ángulos de giro de la cara, sino ofrecer unos valores que sean útiles como señales de control en un interface perceptual. La segunda técnica, explicada en el apartado 7.2.2, es una estimación de pose orientada al análisis de una secuencia vídeo, suponiendo que la posición del individuo no cambia (por ejemplo, está sentado en una silla). Gracias a esta fuerte restricción, veremos que se puede conseguir una alta fiabilidad en la estimación. En el apartado 7.2.3 se detalla el desarrollo e implementación de un interface perceptual –disponible públicamente– basado en la estimación heurística. Por último, las conclusiones y valoraciones finales se encuentran en el apartado 7.2.4.

7.2.1. Estimación heurística de la posición 3D

En adelante, suponemos que tenemos una cara que ha sido encontrada en cierta imagen i, y que es descrita mediante las posiciones del ojo izquierdo, $o1 = (o1_x, o1_y)$, del derecho, $o2 = (o2_x, o2_y)$, y de la boca, $b = (b_x, b_y)$. Además, la imagen ha sido procesada con algún método basado en integrales proyectivas, y hemos calculado la proyección vertical de la cara, PV_{cara}, y la proyección horizontal de los ojos, PH_{ojos}. El objetivo es obtener valores aproximados para: la posición 3D del centro de la cara, $p = (p_x, p_y, p_z)$; la inclinación respecto del plano de la imagen, que denotaremos por *roll*; el giro horizontal –es decir, mirada derecha/izquierda–, que llamamos *yaw*; y el giro vertical –mirada arriba/abajo–, que indicamos con *pitch*. En la figura 7.11 se muestran algunos ejemplos de caras con estos giros.

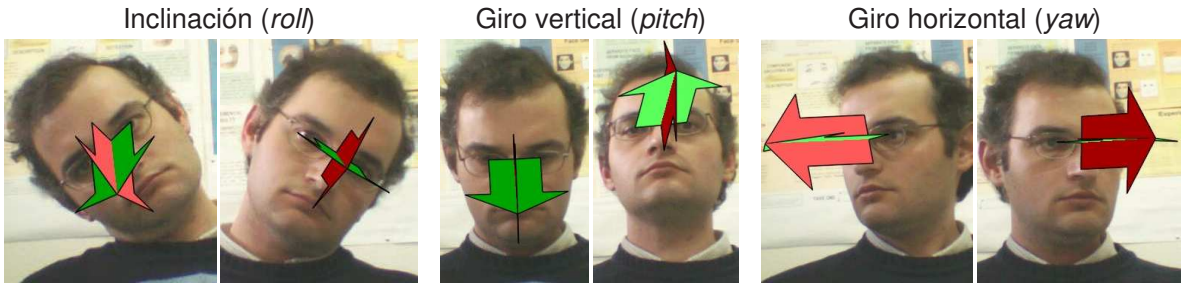


Figura 7.11: *Ejemplos de giros tridimensionales de caras. De izquierda a derecha, inclinación respecto del plano de imagen (roll), giro en sentido vertical (pitch) y en horizontal (yaw). Todos los valores mostrados han sido estimados mediante los métodos propuestos en esta sección.*

Para empezar, cabría preguntarse por la posibilidad de obtener una solución analítica al problema. De hecho, existen 6 grados de libertad y –si consideramos sólo las posiciones de entrada, $o1$, $o2$ y b– tenemos 3 puntos, cada uno de ellos con 2 valores. En teoría, la resolución es posible. Debemos recordar, sin embargo, que en el localizador mediante proyecciones la posición horizontal de la boca se sitúa siempre entre ambos ojos, por lo que realmente sólo existen 5 grados de libertad en la entrada. Esto no ocurre en otros métodos, como los basados en auto-objetos o en el seguidor de Lucas y Kanade; pero son precisamente esas técnicas las que cometen mayor error en la localización de la boca, por lo que la estimación de los parámetros de pose sería muy imprecisa.

En consecuencia, vamos a proponer una alternativa haciendo uso de las integrales proyectivas. El objetivo final es ofrecer valores de control para una aplicación de interface humana basada en las caras. Por lo tanto, la obtención de estimaciones precisas queda por detrás de otros propósitos más relevantes, como que haya estabilidad en los valores de salida, la robustez frente a localizaciones inexactas en el seguimiento, la invarianza con baja calidad de imagen, y la eficiencia computacional del método. Esto justifica algunas de las aproximaciones y simplificaciones que introduciremos en los cálculos.

Estimación de la posición central: p

Supongamos que el sistema de coordenadas en el universo 3D del sujeto está centrado en la cámara, como se muestra en la figura 7.12. El centro de la cara de la persona –que definimos como la posición media entre los ojos y la boca– es el punto $p = (p_x, p_y, p_z)$; la distancia focal la denotamos con f; y la cara aparece en la imagen en el punto $c = (c_x, c_y)$.

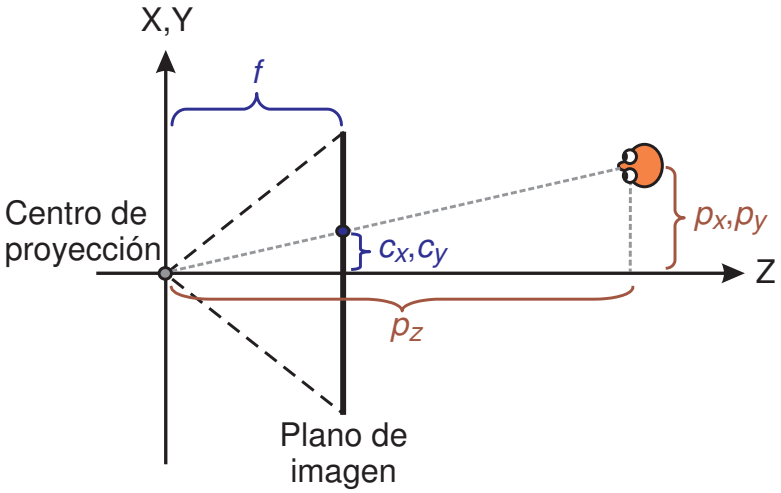


Figura 7.12: *Modelo 3D de la cara y del sistema de adquisición. Se utiliza un modelo de proyección perspectiva. El individuo está situado en la posición $p = (p_x, p_y, p_z)$.*

Considerando una proyección perspectiva perfecta[1], la relación entre los puntos p y c se puede obtener por simple semejanza de triángulos:

$$\frac{c_x}{f} = \frac{p_x}{p_z} \; ; \; \frac{c_y}{f} = \frac{p_y}{p_z} \tag{7.4}$$

Donde:

$$c_x = \frac{o1_x + o2_x + b_x}{3} \; ; \; c_y = \frac{o1_y + o2_y + b_y}{3} \tag{7.5}$$

Luego podemos despejar los valores buscados:

$$p_x = \frac{o1_x + o2_x + b_x}{3} \frac{p_z}{f} \; ; \; p_y = \frac{o1_y + o2_y + b_y}{3} \frac{p_z}{f} \tag{7.6}$$

Obsérvese que ambos están dados en función de la distancia de la cara al centro de proyección de la cámara, p_z, y de la distancia focal, f. Obtener un valor preciso de p_z no es trivial en el caso general. No obstante, es evidente que existe una relación inversa entre p_z y el tamaño de la cara en la imagen: cuando el usuario esté próximo a la cámara, el rostro se verá más

[1] Esto es, obviamos problemas debidos a imperfecciones del sistema de adquisición, como el ratio de aspecto y el sesgo (*skew*) de la matriz de píxeles, la distorsión radial y la imprecisión en el punto principal. Todos ellos tienen un efecto muy secundario para nuestros fines. En algún caso, incluso, se ha comprobado su poca relevancia práctica en el problema general, como describimos en [155] para el caso del punto principal.

grande en las imágenes, y viceversa. Si denotamos por r el tamaño observado de la cara en la imagen i, la distancia a la cámara será:

$$p_z = f\frac{t}{r} \tag{7.7}$$

Siendo t el tamaño real de la cara en el mundo 3D. Pero calcular el "tamaño de la cara en la imagen" no es inmediato. Si lo asociamos con la distancia interocular (una de las elecciones más frecuentes), el "tamaño" se reducirá con los giros horizontales, cuando debería mantenerse constante. Algo parecido ocurre si usamos la distancia de los ojos a la boca. Vamos a definir una alternativa más adecuada, aunque no deja de ser aproximada.

Consideremos una circunferencia hipotética que pasa por los dos ojos y la boca, como se representa en la figura 7.13. La vista en perspectiva de una circunferencia es siempre una elipse, independientemente de la posición y rotación 3D de la figura. Es más, en condiciones de perspectiva débil, el radio mayor de la elipse será siempre el mismo para cualquier rotación. La demostración es sencilla: todas las orientaciones posibles de la circunferencia generan una esfera, y la elipse sólo es una sección central de la misma, de manera que su radio mayor coincide con el de la esfera. Se pueden ver algunos ejemplos en la figura 7.13.

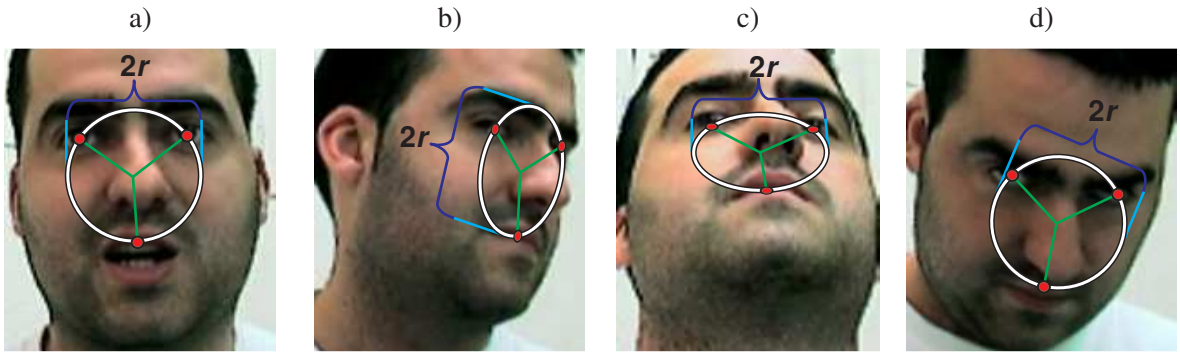


Figura 7.13: *Estimación del tamaño de cara para el cálculo de la profundidad. Aunque la profundidad es parecida en todas las imágenes, la distancia entre los ojos o de los ojos a la boca cambia con la orientación 3D. Sin embargo, el eje mayor, r, de la elipse que contiene a los ojos y la boca se mantiene constante.*

Teniendo en cuenta que los ojos y la boca forman, aproximadamente, un triángulo equilátero, se pueden considerar como un *muestreo uniforme* de la hipotética elipse. En consecuencia, la matriz de covarianzas de los tres puntos está relacionada estrechamente con la elipse. La matriz toma la siguiente forma:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = 1/3 \cdot P^{\mathrm{T}} \cdot P \; ; \; \text{con } P = \begin{bmatrix} o1_x - c_x & o1_y - c_y \\ o2_x - c_x & o2_y - c_y \\ b_x - c_x & b_y - c_y \end{bmatrix} \tag{7.8}$$

Concretamente, el radio mayor viene dado por la dirección de máxima varianza de Σ. Por lo tanto, el primer autovalor de Σ indica esa varianza; y el tamaño estimado de la cara, r, se

puede obtener multiplicándolo por 2. Para la pequeña matriz Σ de 2×2, el primer autovalor se puede calcular con una fórmula cerrada:

$$r = \sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2} \tag{7.9}$$

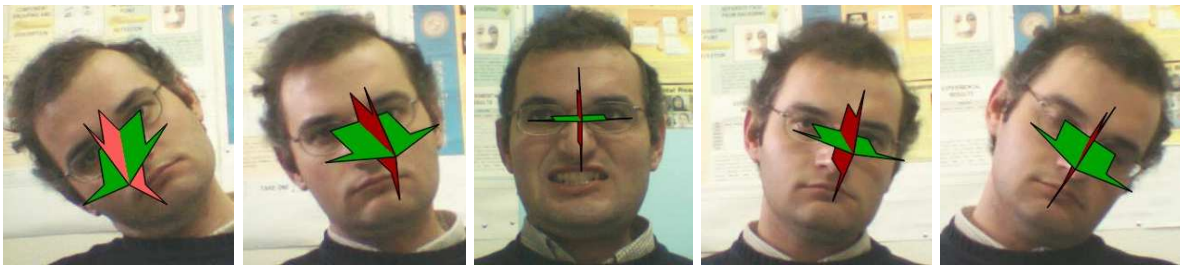
Quedan pendientes los parámetros de distancia focal, f, y el tamaño de la cara en el mundo real, t, de la ecuación 7.7. Ambos se puede resolver a priori mediante calibración. Por ejemplo, el valor de t está típicamente alrededor de los 4 cm. Pero en realidad, si observamos las ecuaciones 7.7 y 7.6, podemos apreciar que ambos parámetros (f y t) influyen únicamente en el factor de escala. De esta manera, podemos asignarles valores fijos, y tener en cuenta que todas las distancias obtenidas serán relativas a la escala usada.

Estimación de la inclinación: *roll*

Cuando los otros ángulos son pequeños, la inclinación de la cara se puede asociar fácilmente con el ángulo de rotación observado en la imagen, es decir, el de la recta que pasa por ambos ojos. Por lo tanto, el cálculo de la inclinación es trivial:

$$roll = \arctan \frac{o2_y - o1_y}{o2_x - o1_x} \tag{7.10}$$

En la figura 7.14 se pueden ver algunos ejemplos de valores obtenidos de este ángulo en un vídeo de prueba.



roll = -43,7º *roll* = -19,6º *roll* = -2,8º *roll* = 15,9º *roll* = 34,6º

Figura 7.14: *Estimación heurística de la inclinación facial. Se muestran algunas caras y el valor resultante. Obsérvese que algunas imágenes presentan distintas expresiones faciales.*

En general, los cálculos para esta medida son bastante fiables. El ángulo de *roll* es uno de los parámetros más precisos en la estimación de pose. No obstante, debemos advertir que la fórmula 7.10 no siempre es correcta en todos los casos. Por ejemplo, si la cámara está situada a distinta altura de la persona, aunque el rostro no esté inclinado, un desplazamiento o un giro lateral pueden hacer que ambos ojos aparezcan en la imagen con diferente valor en Y; pero, obviamente, la ecuación 7.10 devolverá una inclinación distinta de 0.

Estimación del giro horizontal: *yaw*

Algunos autores han sugerido varias formas de estimar el giro horizontal de la cara –mirada izquierda/mirada derecha–, mediante sencillos criterios heurísticos: usando la posición de la nariz; considerando la posición de los ojos en relación a la forma global de la cara; etc. Nosotros proponemos una alternativa basada exclusivamente en las integrales proyectivas.

Si nos fijamos en la proyección horizontal de la región de los ojos, PH_{ojos}, podemos distinguir claramente dos mínimos locales, correspondientes a las partes más oscuras de ambos ojos. La zona de la nariz se destaca por ser más clara y aparecer centrada entre ambos. Se muestran varios ejemplos en la figura 7.15, donde las proyecciones han sido aplicadas sobre la región de interés.

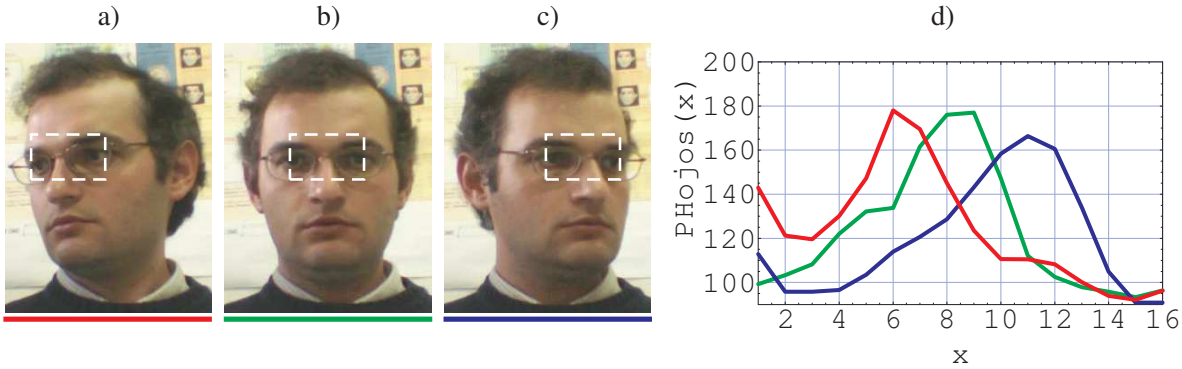


Figura 7.15: *Proyección horizontal de los ojos para la estimación de pose. a,b,c) Tres caras con distintos ángulos de giro horizontal. Se muestra (en línea discontinua) la región de ojos usada en las proyecciones. d) Proyecciones horizontales de las regiones señaladas. Observar que el pico (correspondiente a la zona más clara entre los ojos) se desplaza según el grado del giro. El tamaño de las señales es 16.*

A medida que la cara gira hacia uno u otro lado, el punto al que se proyecta la nariz aparece descentrado hacia el lado correspondiente en PH_{ojos}. Por lo tanto, en condiciones ideales, el ángulo de giro horizontal se puede deducir a partir de la posición del máximo de la nariz en relación a los mínimos de los ojos. Más concretamente, si la cara está alineada horizontalmente respecto del modelo de MH_{ojos} (lo cual ocurrirá si hemos usado el localizador mediante proyecciones), los mínimos están situados en posiciones fijadas de antemano, x_{ojos} y $w - x_{ojos}$, en relación al ancho w de la cara en PH_{ojos}. Así pues, buscamos:

$$yaw' = \text{argmáx}_{\forall x \in \{x_{ojos},\ldots,w-x_{ojos}\}} PH_{ojos}(x) \tag{7.11}$$

Si las proyecciones PH_{ojos} son de tamaño no muy grande –como ocurre en la figura 7.15d)–, los valores obtenidos con la ecuación 7.11 varían muy poco y de forma discreta. Para mejorar la resolución podemos hacer una interpolación alrededor del máximo, del tipo:

$$yaw'' = \frac{\sum_{i=-k}^{k}(yaw' + k) \cdot PH_{ojos}(yaw' + k)}{\sum_{i=-k}^{k} PH_{ojos}(yaw' + k)} \tag{7.12}$$

En un uso típico de la estimación de pose –al estilo del que proponemos más adelante–,

podemos tomar directamente el valor $giro_{derizq} = yaw'' - w/2$, como un parámetro de control giro izquierdo/giro derecho. Si necesitamos una aproximación al valor concreto del ángulo, podemos obtenerlo mediante una simple construcción geométrica. Suponiendo que el tabique nasal sobresale n_r centímetros de la profundidad de los ojos, y que la distancia interocular es de m_r centímetros, el valor resultante sería:

$$yaw = \arcsin \frac{yaw'' - w/2}{n_r \cdot (w - 2x_{ojos})/m_r} \tag{7.13}$$

Nótese que el valor $w - 2x_{ojos}$ representa la distancia interocular en píxeles. En la figura 7.16 se presentan varios casos de estimación del ángulo de giro horizontal. En estos ejemplos se han tomado los valores típicos: n_r = 2 cm y m_r = 7 cm.

$giro_{derizq}$ = -1,8 yaw = -31,7º | $giro_{derizq}$ = -0,6 yaw = -10,1º | $giro_{derizq}$ = -0,1 yaw = -1,7º | $giro_{derizq}$ = 0,8 yaw = 13,5º | $giro_{derizq}$ = 1,6 yaw = 27,8º

Figura 7.16: *Estimación heurística del giro horizontal de la cara. Se muestran algunas caras y los valores resultantes: distancia del pico máximo ($giro_{derizq}$), y ángulo estimado (yaw).*

El método propuesto suele ser bastante fiable, siempre que la localización de los ojos sea más o menos estable. En caso de no serlo, pueden ocurrir oscilaciones en los valores resultantes. Por ejemplo, en un caso típico, una variación de 1 punto en la posición del máximo de la ecuación 7.13, puede modificar el resultado alrededor de 15º. Para evitar la oscilación, se podría introducir algún tipo de suavizado temporal en los resultados obtenidos, por ejemplo mediante filtrado de Kalman.

Estimación del giro vertical: *pitch*

Si en la obtención del giro horizontal hemos utilizado la proyección horizontal de los ojos, para el cálculo del giro vertical (mirada arriba/abajo) aprovechamos la proyección vertical de la cara. El efecto de esta rotación sobre las proyecciones no es tan evidente. Veamos algunos ejemplos en la figura 7.17.

Si analizamos las proyecciones de la figura 7.17d), podemos apreciar una cierta progresividad en los valores de gris de las mismas. En posición de mirada hacia abajo –figura 7.17a)–, la proyección obtenida toma valores más oscuros; mientras que para mirada hacia arriba –figura 7.17c)– es más clara. Este hecho es más notable en la parte superior de la señal, aproximadamente entre los valores 1 y 10 de Y, correspondientes a la zona de los ojos.

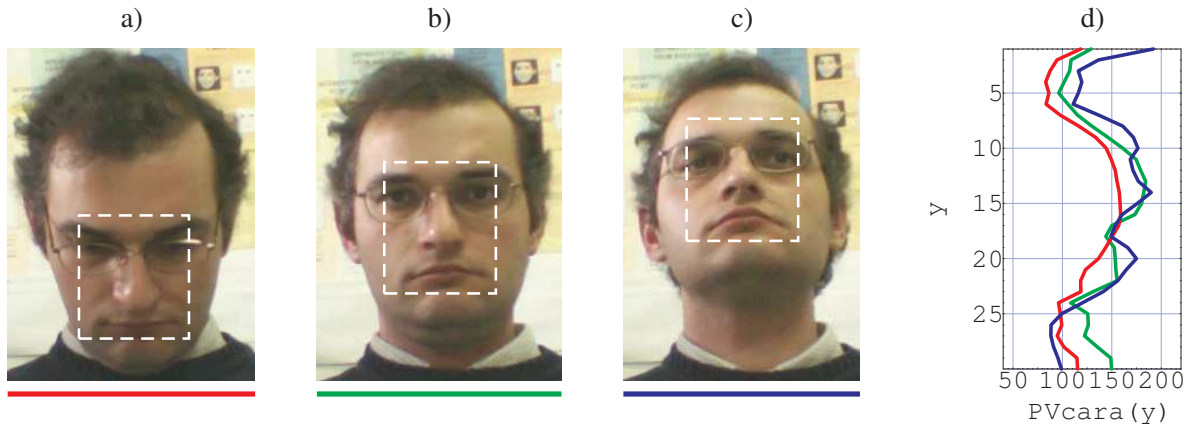


Figura 7.17: *Proyección vertical de la cara para la estimación de pose. a,b,c) Tres caras con distintos ángulos de giro vertical. Se muestra (en línea discontinua) la región proyectada. d) Proyecciones verticales de las regiones señaladas.*

Existen dos explicaciones razonables para este fenómeno. En primer lugar, al mirar hacia abajo –figura 7.17a)–, los ojos y las cejas aparecen más juntos, por lo que la zona aparece más oscura. Por otro lado, puesto que la iluminación procede de arriba, la luz que recibe la cara depende de su orientación vertical. Al agachar la cabeza, se oculta de la fuente de luz haciéndola aparecer más oscura. Al levantarla –figura 7.17c)–, existen menos sombras, y la cara tiene tonos más claros.

Está claro que surgen inconvenientes relacionados con este criterio meramente heurístico. En primer lugar, ¿qué ocurre cuando la iluminación no sea superior sino, por ejemplo, lateral? O, ¿qué pasa si cambian las condiciones luminosas? En segundo lugar, la medida resultante no se puede tomar en términos absolutos, sino en comparación con las obtenidas en imágenes anteriores; es decir, servirá para indicar si hay movimiento hacia arriba o hacia abajo. Estas dos limitaciones son asumibles para los propósitos del interface perceptual. Por un lado, el uso típico será en un entorno de interior, donde los focos de luz están situados en el techo. Por otro lado, el interface maneja una fuente de vídeo, y se puede dar por hecho que en la primera imagen la cara mira de frente.

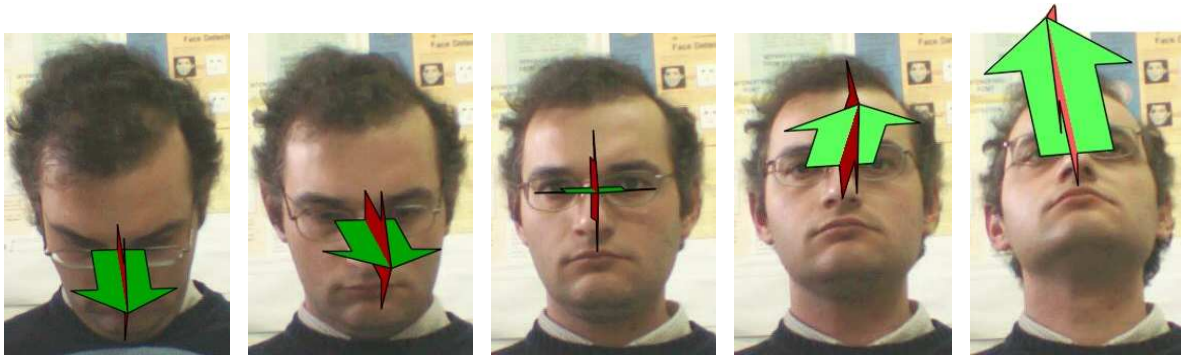
En definitiva, la estimación del giro vertical sería de la siguiente forma. En primer lugar, obtenemos la proyección vertical de la cara, PV_{cara}. Después se calcula la media de esa señal entre ciertas posiciones. En concreto, proponemos usar el primer tercio de la señal; como se observa en la figura 7.17d), es en el que más claramente se refleja el fenómeno descrito. Tenemos:

$$pitch = \sum_{i=0}^{h/3} PV_{cara}(i) - pitch_{ini} \tag{7.14}$$

Siendo h el tamaño de la proyección PV_{cara}, y $pitch_{ini}$ el valor del sumatorio de la fórmula 7.14 para el primer *frame* de la secuencia.

Se presentan algunas muestras de estimación del giro vertical de la cara en la figura 7.18.

Tal y como se deduce de la ecuación 7.14, los valores obtenidos no son grados sino simples medidas relativas de la cantidad de rotación.



pitch = -11,17 *pitch* = -7,66 *pitch* = -0,87 *pitch* = 8,23 *pitch* = 18,03

Figura 7.18: *Estimación heurística del giro vertical de la cara. Se muestran algunas caras y los valores resultantes de pitch (valor relativo).*

En la práctica, los resultados de esta heurística permiten lograr una buena distinción de cuándo la cara mira hacia arriba o hacia abajo. También hay una adecuada proporcionalidad del valor con el mayor o menor grado de giro. Un inconveniente de este método –además de los indicados previamente– es que resulta ligeramente sensible a las expresiones faciales. Por ejemplo, un gesto de levantar las cejas tiende a aumentar un poco el valor de *pitch*.

7.2.2. Estimación basada en suposición de posición fija

El problema de estimar la pose facial se simplifica considerablemente al introducir dos suposiciones: (1) la posición del individuo no cambia a lo largo de la secuencia (esto es, la cabeza puede girar pero tanto la cámara como el cuerpo permanecen fijos), y (2) inicialmente el sujeto se encuentra mirando de frente a la cámara. Estas condiciones se pueden dar, por ejemplo, si el usuario permanece sentado en una silla. Si no están garantizadas, se podría aplicar una etapa inicial de detección y seguimiento del cuerpo para compensar el movimiento encontrado. En cualquier caso, nuestro único propósito con este método es analizar comparativamente la estimación basada en proyecciones.

Aprovechando estas dos premisas, es posible estimar los 6 parámetros de pose 3D usando exclusivamente las posiciones de los elementos faciales. Por lo tanto, esta alternativa no hace uso de las proyecciones de manera implícita (aunque pueden haberse aplicado para realizar el seguimiento).

Estimación de la posición central y de la inclinación

La forma de estimar la posición central del rostro es la misma que la propuesta en el apartado 7.2.1. Recordemos que el procedimiento estaba basado en las localizaciones de los componentes obtenidas mediante el seguimiento, de manera que el cálculo es exactamente igual para ambos casos.

También resulta idéntica la estimación del ángulo de inclinación de la cara, *roll*. Por definición, tomamos la rotación respecto del plano de la imagen, esto es, el ángulo observado de la línea que pasa por ambos ojos.

Estimación del giro vertical y horizontal

De manera simplificada –y posiblemente no muy rigurosa–, el sistema cuello/cabeza de un humano se puede modelar como un brazo robótico con tres grados de libertad rotacionales, que llamamos *roll*, *pitch* y *yaw*, como se muestra en la figura 7.19. Puesto que el primero ha sido estimado de forma independiente, deshacemos a priori su efecto sobre las imágenes, reduciendo el problema a dos variables: estimar el giro vertical y el horizontal.

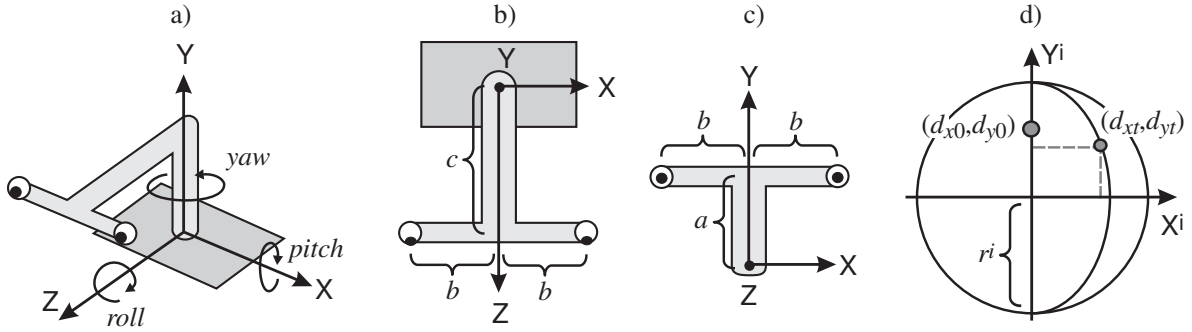


Figura 7.19: *Modelo simplificado de la cara para la estimación de pose. a) Esquema de la cara e interpretación de los tres giros. b,c) Vista superior y frontal del esquema, respectivamente. Los parámetros indican: a - altura de los ojos al eje de giro; b - mitad de la distancia interocular; c - distancia horizontal del centro de los ojos al cuello. d) El punto medio de los ojos en coordenadas de la imagen, (d_x, d_y), y el círculo generado para distintos giros.*

Como es habitual en el dominio del procesamiento facial, consideramos que el efecto de la perspectiva es despreciable, esto es, trabajamos con *perspectiva débil* [108, capítulo 4]. Suponiendo que el eje de giro del cuello se encuentra inicialmente en el punto $p_0 = (p_{x0}, p_{y0}, p_{z0})$, el ojo izquierdo estará situado en $(p_{x0} - b, p_{y0} + a, p_{z0} - c)$ y el derecho en $(p_{x0} + b, p_{y0} + a, p_{z0} - c)$, siendo los parámetros a, b y c los definidos en la figura 7.19. El punto medio entre los ojos será $(p_{x0}, p_{y0} + a, p_{z0} - c)$.

No es difícil deducir que en el modelo "robótico" ideal cualquier punto de la cabeza se mueve en una esfera al variar los valores de *pitch* y *yaw*. En consecuencia, el problema se reduce a una clásica estimación de longitud/latitud en una órbita esférica. Obviamente, este resultado no deja de ser una aproximación –a causa de las simplificaciones introducidas–. Pero la propia anatomía de la cabeza reduce los posibles grados de giro permitidos, de manera que la aproximación es válida y precisa para ángulos no muy elevados.

Si nos fijamos en el punto medio de los ojos, la hipotética esfera en la que se mueve tendría de radio (en unidades reales del mundo 3D):

$$r^m = \sqrt{a^2 + c^2} \tag{7.15}$$

En coordenadas de la imagen, la esfera se proyecta en un círculo de radio:

$$r^i = r^m \frac{f}{p_{z0}} \tag{7.16}$$

Haciendo uso de la suposición (2), si el punto medio de los ojos en la imagen inicial de la secuencia es $d_0 = (d_{x0}, d_{y0})$, el centro del círculo sería el punto:

$$(d_{x0}, d_{y0} - a\frac{f}{p_{z0}}) \tag{7.17}$$

Una vez obtenidos estos valores, podemos calcular los ángulos *pitch* y *yaw* para cada nueva imagen i_t, con punto medio de los ojos en $d_t = (d_{xt}, d_{yt})$. En concreto, el giro en sentido vertical sería:

$$pitch = \arcsin \frac{d_{yt} - (d_{y0} - a \cdot f/p_{z0})}{r^i} - \arcsin \frac{a}{c} \tag{7.18}$$

Obsérvese que se resta siempre $\arcsin(a/c)$, debido a que la posición de mirada al frente, $pitch = 0$, no corresponde necesariamente con el centro del círculo.

Por su parte, el giro horizontal se puede calcular fácilmente con:

$$yaw = \arcsin \frac{d_{xt} - d_{x0}}{r^i \cdot \cos(pitch + \arcsin(a/c))} \tag{7.19}$$

Ejemplos comparativos de estimación de pose

Para contrastar la fiabilidad de los dos métodos propuestos de estimación de pose, hemos grabado un vídeo en el que se realiza una variedad de giros de la cabeza: arriba/abajo; izquierda/derecha; arriba/abajo mirando a la izquierda; arriba/abajo mirando a la derecha; izquierda/derecha mirando arriba; e izquierda/derecha mirando abajo. La posición del individuo permanece fija. Sobre la secuencia se aplica el seguimiento basado en proyecciones, y las posiciones obtenidas se entregan a ambos métodos para el cálculo de la pose.

Los resultados conseguidos se muestran en la figura 7.20. Nos centramos en las estimaciones de los giros vertical y horizontal, puesto que los demás parámetros se calculan de manera más o menos directa a partir de las posiciones (su fiabilidad se analizó ya extensamente en el capítulo 5).

A la vista de los resultados, podemos hacer algunas valoraciones sobre los estimadores de pose desarrollados:

1. **Valoración global.** En términos generales, ambos métodos ofrecen una buena respuesta a los movimientos de la cara, produciendo una salida proporcional al grado de giro y más o menos estable en condiciones no extremas. Los dos resultados presentan una alta correlación en los valores obtenidos, lo que apoya la fiabilidad de ambas estimaciones.

2. **Dependencia entre giros.** En ocasiones, la estimación heurística puede dar lugar a una ligera interdependencia entre los ángulos de *pitch* y *yaw*. Este fenómeno es evidente

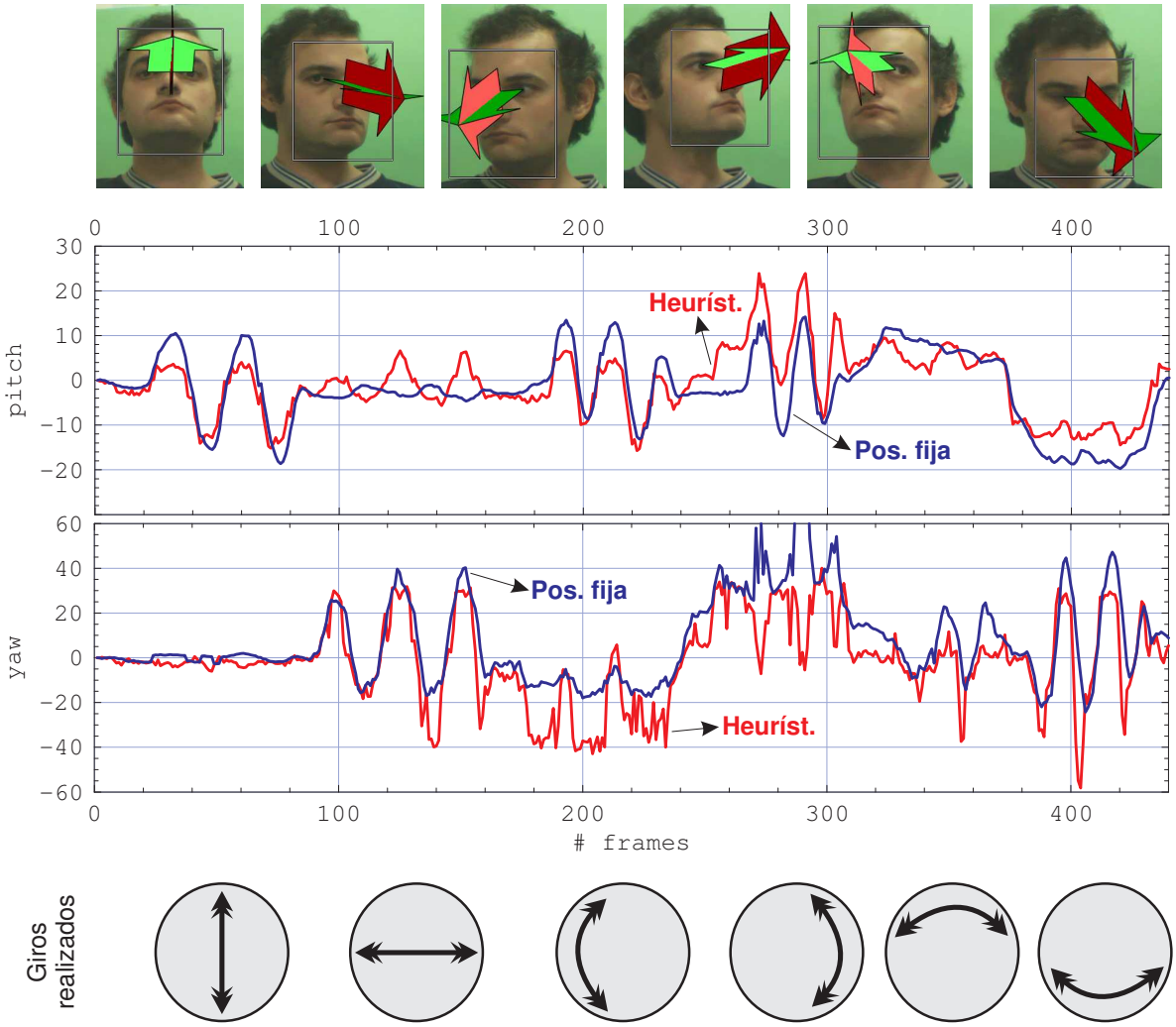


Figura 7.20: *Ejemplos de estimación de pose en una secuencia de vídeo. Se muestran los valores de giro vertical (pitch) y horizontal (yaw) a lo largo del tiempo, para el método heurístico (en rojo) y el basado en posición fija (en azul). En la parte superior aparecen algunos extractos de caras, en las posiciones aproximadas donde ocurren (escala "# frames"). Abajo, se interpretan los giros realizados en la secuencia, también a lo largo del tiempo.*

entre los *frames* 110 y 150, donde el valor de *pitch* varía regularmente cuando debería permanecer más o menos fijo. También ocurre una variación esporádica de *yaw* con movimientos arriba/abajo cuando la cara mira a izquierda o derecha, entre los *frames* 180 y 280.

3. **Giros combinados.** Normalmente, los giros combinados son los más problemáticos. Un ejemplo extremo ocurre entre los *frames* 280 y 300, donde existe una gran rotación a la derecha (por encima de los 40º) al mismo tiempo que se mira hacia arriba. La estimación de *pitch* es fiable en ambos casos, pero el valor de *yaw* se "descontrola" tanto para el método heurístico como para el basado en posición fija. Este mal resultado es debido en buena parte a una escasa fiabilidad del seguimiento, que produce posiciones con alto

error para esos casos.

4. **Márgenes de trabajo admitidos.** En relación con lo anterior, podemos decir, de forma aproximada, que el intervalo de ángulos donde ambos métodos funcionan de manera óptima está entre los $\pm 20^o$ para el caso del giro vertical, y los $\pm 40^o$ en los giros laterales. En principio, no hay limitación en la inclinación respecto al plano de la imagen, el ángulo *roll*.

5. **Eficiencia computacional.** En cuanto a los tiempos de ejecución, es fácil intuir que ambos métodos tienen una escasa complejidad computacional, ya que los cálculos son sencillos en todos los casos. En concreto, en un Pentium IV a 2,60GHz la estimación heurística tarda una media de 0,18 ms, siendo el máximo de 0,42 ms. Obviamente, la técnica de posición fija es aún más rápida, puesto que usa un número constante de operaciones. El tiempo medio para el ejemplo se reduce hasta los 0,006 ms.

7.2.3. Desarrollo de un interface perceptual

Con el fin de demostrar la viabilidad de la técnica heurística de estimación de pose, hemos desarrollado un interface perceptual que pone en práctica las propuestas realizadas en el apartado 7.2.1 [64, 61]. La aplicación consiste en un mundo tridimensional similar a un juego de acción en primera persona, que es controlado mediante los movimientos de la cabeza del usuario. Además de la estimación de pose, también se utilizan los métodos de localización y seguimiento de caras basados en integrales proyectivas, de manera que se ponen en funcionamiento gran parte de las propuestas realizadas en esta tesis.

El desarrollo de esta aplicación[2] ha sido realizado en C++ utilizando el entorno Microsoft Visual Studio .NET 2003. En lo relativo a la parte gráfica, se han manejado las funcionalidades proporcionadas por DirectX 9.0 [189]. Y en la parte de procesamiento de imágenes utilizamos, como en el resto de esta tesis, las librerías de Intel OpenCV e IPL [35].

Descripción del mundo virtual 3D

En la figura 7.21 se muestra una vista global del universo en el que se desenvuelve el interface perceptual, manejado a través de los movimientos faciales del usuario.

Para la parte de generación del mundo virtual se ha creado un motor gráfico sencillo bajo DirectX 9.0. No obstante, el motor incorpora algunas técnicas avanzadas de renderización, como los *portales* y el denominado *frustum culling* [189], consistente en eliminar todos los polígonos del entorno que están fuera del alcance de visión. Con ello, se consigue una frecuencia de refresco de unos 40-50 *frames* por segundo en un ordenador medio.

Básicamente, el entorno virtual consta de una serie de *habitaciones* o *estancias* comunicadas entre sí por pasillos. A su vez, estas habitaciones se dividen en *celdas* con forma de

[2]Parte del interface perceptual que describimos en este apartado –en concreto, la correspondiente al control y a la renderización gráfica 3D– ha sido desarrollada en el contexto de un proyecto fin de carrera [54]. El programa creado y la documentación del proyecto están disponibles públicamente en: `http://dis.um.es/~ginesgm/th`.

Figura 7.21: *Una vista de pájaro del entorno virtual de "Tierra Inhóspita", el mundo tridimensional en el que se aplica el interface perceptual.*

cuadriláteros convexos. En cada celda se definen las partes visibles del mapa y las que no lo son. Estas celdas tienen también un papel importante en la descripción de las paredes y la detección de colisiones con las mismas. En la figura 7.22a) se pueden ver las celdas existentes para un trozo del mapa. También se presentan algunos ejemplos de vistas del mundo en primera persona en las figuras 7.22b-f).

Se han incorporado algunos objetos 3D complejos, texturas y un efecto de niebla para dar mayor ambiente y sensación de realismo a las escenas, aunque no era el objetivo primordial del programa.

Parámetros de control del entorno

El aspecto más interesante del entorno desarrollado es la integración de la generación de imágenes con la estimación visual de la pose. Tras la aplicación del seguimiento, los resultados del método heurístico son utilizados como los parámetros de control de movimiento en el interface perceptual. En concreto, los seis valores obtenidos en el proceso son utilizados de la siguiente forma:

- **Posición en X.** La localización horizontal de la cara se aprovecha para realizar desplazamientos laterales del personaje sin cambiar el punto de mirada. Las posiciones se toman siempre en relación a la inicial. De esta forma, la cantidad del desplazamiento está en función de la distancia en X respecto de la situación de la cara en el primer *frame*. Existe un cierto umbral configurable para no realizar desplazamientos cuando la diferencia es

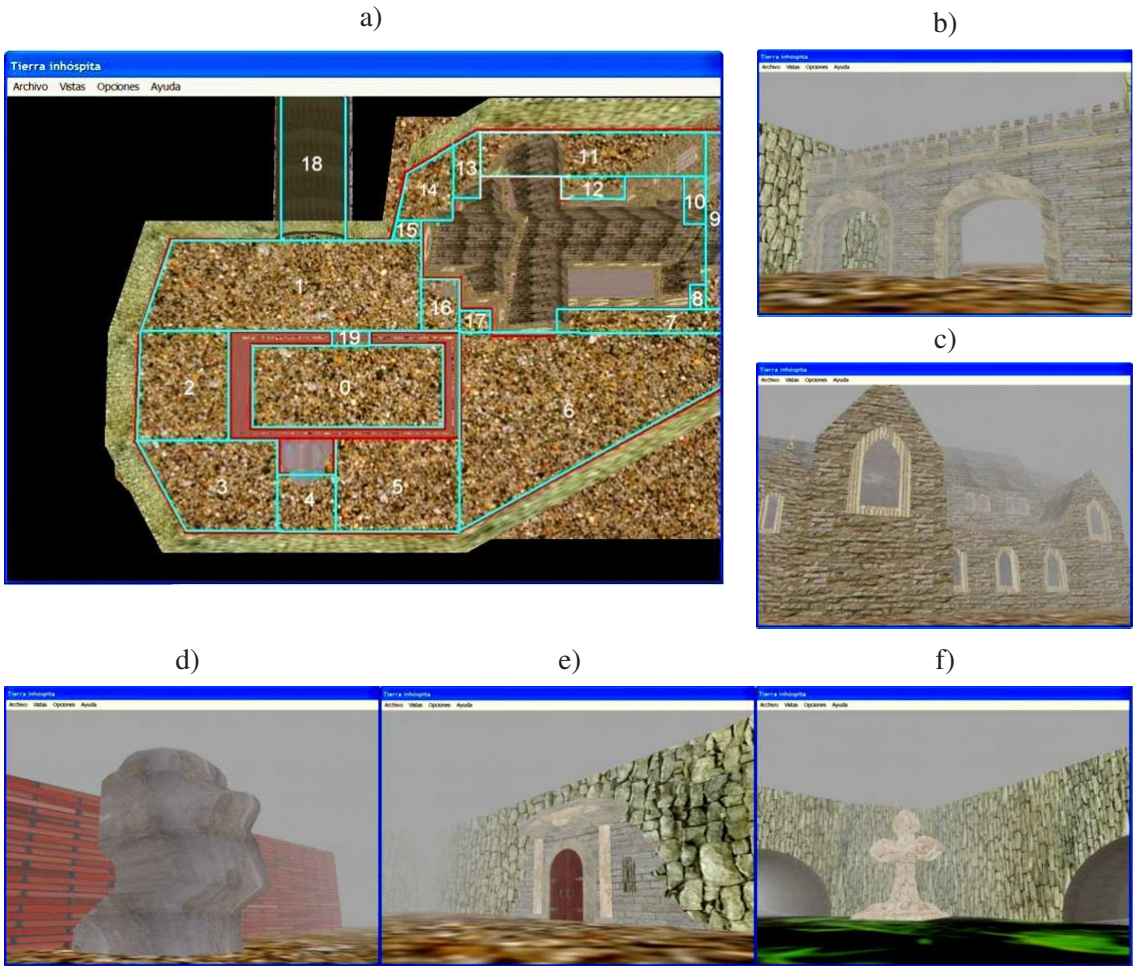


Figura 7.22: *Definición de celdas y vistas en primera persona del entorno virtual. a) Un ejemplo de las celdas definidas para un trozo del mapa. b-f) Algunas vistas del entorno en el modo normal de uso. Los objetos 3D introducidos en las escenas han sido obtenidos de fuentes públicas en la red.*

pequeña.

- **Posición en Y.** Como en otros muchos juegos en primera persona, el personaje puede caminar agachado, en posición normal o de puntillas; es decir, la altura del jugador respecto del plano del suelo puede ser baja, media o alta, respectivamente. El valor en Y resultante de la estimación de pose se utiliza para controlar el modo de andar. Las alturas posibles están en un rango limitado entre el modo bajo y el alto. Como en el caso anterior, el control es relativo a la posición inicial de la cara en Y.

- **Posición en Z.** La distancia del rostro a la cámara, también medida de forma relativa, se utiliza para avanzar o retroceder. Cuando disminuye la distancia se camina hacia delante, y cuando aumenta se anda hacia atrás (pero sin darse la vuelta). La cantidad de movimiento está graduada, de manera que cuanto más se acerque el usuario a la cámara se avanzará más rápidamente.

- **Giro en *roll* y *yaw*.** Estos dos valores se utilizan para controlar la rotación a derecha e izquierda del personaje. Si el valor absoluto del ángulo *roll* supera cierto umbral mínimo, se realiza una rotación hacia el lado correspondiente. En caso contrario, se analiza el valor de *yaw*; si es significativo, se lleva a cabo el giro. En ambas situaciones, la velocidad de giro depende del ángulo. Hay varias razones para la aparente redundancia de usar dos parámetros en un mismo control. Por un lado, no aparecen muchos controles más de interés en un interface como el desarrollado. Por otro lado, se evita la posible interdependencia entre ambas estimaciones.

- **Giro en *pitch*.** Normalmente, en su desplazamiento por el mundo virtual, el personaje estará mirando hacia delante. Pero también puede dirigir la vista hacia arriba o hacia abajo. Para este control se utiliza el valor estimado de *pitch*. Ya vimos que el dato obtenido no es un ángulo, sino un número relativo. Por ello, no existe una graduación sino que únicamente se usa para analizar si el usuario está mirando de frente, arriba o abajo. En los dos últimos casos, la mirada del personaje girará verticalmente en el sentido que corresponda. Cuando no se detecta un valor de *pitch* significativo, el punto de mira se mueve progresivamente hacia el frente (es decir, es la posición por omisión).

Debemos comentar otros dos aspectos sobre la forma de traducir la estimación de pose en parámetros de control en el mundo virtual. En primer lugar, en una aplicación de este tipo la **estabilidad** de los controles es un requerimiento básico. El método heurístico puede producir ligeras oscilaciones, pero que serían muy molestas si son trasladadas a las señales de control. Para evitarlo se ha introducido un suavizado temporal en todas las señales de 5 *frames*, de manera que los valores usados para la pose son el promedio de las últimas 5 estimaciones obtenidas. Esto introduce un pequeño retardo –que en el peor caso, a 10fps, sería de medio segundo– pero elimina de forma efectiva las oscilaciones por el error de las medidas.

En segundo lugar, como ya hemos mencionado, la estimación de algunos parámetros se vuelve inestable o errónea a medida que los ángulos de giro alcanzan los máximos admitidos. En otras palabras, existe una cierta **interdependencia** entre las diferentes mediciones. Con el fin de paliar estos problemas se ha definido una priorización de las señales de control. Cuando un ángulo estimado alcanza un valor elevado, los menos prioritarios no son tenidos en cuenta. En concreto, la medida de *roll* es la más prioritaria, seguida de *pitch*.

7.2.4. Resultados, conclusiones y valoraciones

En la figura 7.23 se muestran algunos extractos de secuencias de vídeo disponibles en la misma página web del programa.

Fruto de nuestra experiencia, podemos hacer algunas reflexiones sobre la viabilidad de las técnicas propuestas, el sistema desarrollado y sobre los interfaces perceptuales en general:

1. **Valoración global de los resultados.** En término medio, el funcionamiento del interface perceptual creado es bastante satisfactorio. El método de seguimiento de caras median-

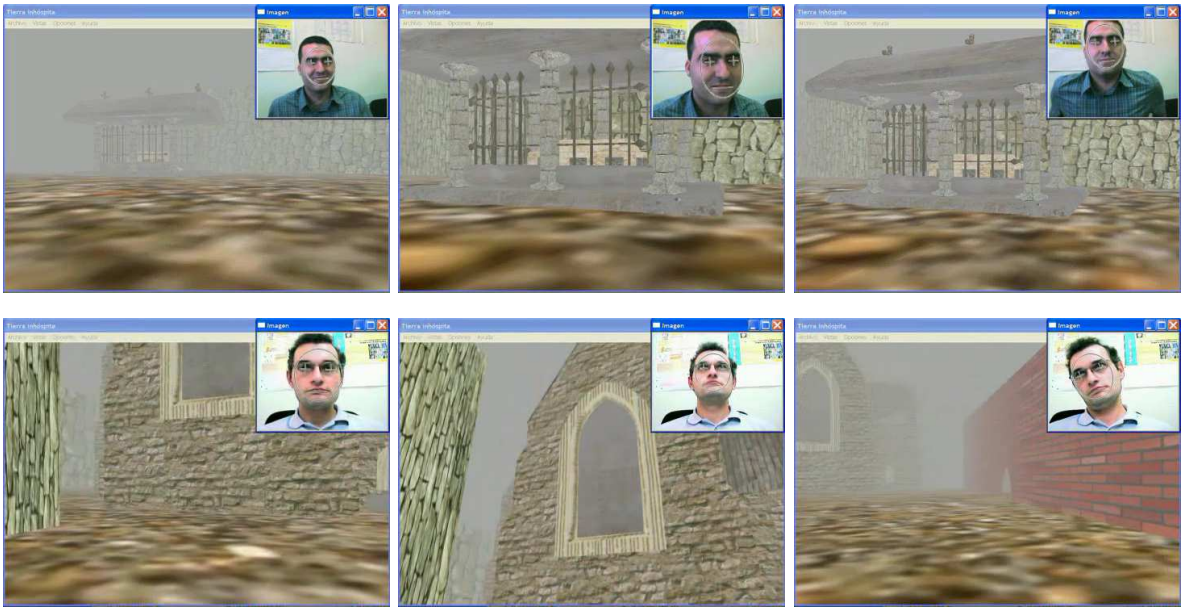


Figura 7.23: *Ejemplos de ejecución del interface perceptual desarrollado. Estos vídeos de prueba se pueden encontrar en la página:* `http://dis.um.es/~ginesgm/th`.

te proyecciones garantiza una alta robustez frente a expresiones faciales, giros, movimientos y cambios de iluminación. La estimación heurística de pose ofrece una rápida indicación de la orientación de la cabeza. Y la forma de producir las señales de control proporciona una estabilidad y velocidad de respuesta suficientes para este tipo de aplicaciones.

2. **Manejo de las situaciones de error.** Las situaciones de pérdida del seguimiento no resultan críticas en una aplicación de este tipo. Cuando la cara desaparece, simplemente dejan de enviarse señales de control. La detección se aplica periódicamente hasta volver a encontrar la cara, y entonces continua el proceso. En nuestro caso, cuando ocurre la pérdida no se vuelven a establecer los valores iniciales para la posición del rostro, sino que se mantienen mientras que no lo indique el usuario.

   Tampoco es grave la limitación en los ángulos de pose o la velocidad máxima de seguimiento. En el segundo caso, es raro que el usuario genere estos movimientos en un uso normal. El primero es más probable que suceda. Pero, si la cámara está situada cerca del monitor, un ángulo de giro grande significa que el usuario no está mirando a la pantalla, de manera que lo más lógico es suprimir las señales de control.

3. **Condiciones de iluminación.** Posiblemente, el factor más crítico en una aplicación de este tipo es la iluminación. Especialmente problemático es el caso de la iluminación exterior, o mixta, cuando la luz solar se recibe lateralmente. Tanto la detección, como el seguimiento y el cálculo de pose presentan dificultades para funcionar correctamente. Un ejemplo de esta situación es una habitación con una ventana al exterior por la que

entra la luz directa del sol. Con mucha probabilidad –en una cámara típica de bajo coste– media cara aparecerá oscura y la otra media saturada a blanco; bajo estas circunstancias, algunas de las heurísticas introducidas se vuelven muy imprecisas.

4. **Aspectos de usabilidad.** Desde un punto de vista más exigente, la viabilidad práctica de los interfaces perceptuales –como el desarrollado u otros existentes– podría encontrarse con ciertos obstáculos. Por un lado, algunos gestos resultan poco naturales o incluso difíciles de conseguir, como el desplazamiento vertical y el acercamiento/alejamiento a la cámara. Un uso prolongado puede acabar produciendo cierta fatiga [16], y el usuario podría "echar de menos" el teclado. Para paliar este problema, nuestro interface perceptual permite usar el teclado al mismo tiempo que la entrada visual. Una estimación algunos órdenes de magnitud más precisa permitiría ir más allá en los usos del interface. Por ejemplo, eliminaría la necesidad de realizar grandes movimientos con la cabeza, reduciendo el riesgo de fatiga. Pero, lógicamente, se requerirían más recursos computacionales, disminuyendo así la respuesta del proceso.

5. **Aplicaciones prácticas.** Una posible aplicación práctica para las técnicas desarrolladas serían los sistema de ayuda a discapacitados. El seguimiento del rostro evitaría el uso de dispositivos de control mecánicos sujetos a la cabeza. No obstante, este dominio de trabajo obliga a poner un énfasis especial en los requerimientos de robustez, puesto que las situaciones de error son más críticas.

En conclusión, pensamos que un interface perceptual no debe suponer necesariamente la eliminación de la entrada de teclado: se añaden nuevas formas de controlar las aplicaciones, pero sin eliminar las existentes. De esta forma, el usuario ve aumentadas sus posibilidades de interacción con la máquina, sin desprenderse de las que le son más habituales y conocidas. Por ejemplo, en un juego se podrían utilizar las teclas para el movimiento del personaje, y la estimación de pose para el punto de mirada. En el futuro, a medida que aumente la precisión de las estimaciones, veremos multiplicados los usos posibles de estos sistemas. La iluminación deficiente será uno de los grandes obstáculos que se deberán abordar.

7.3. Resumen

El universo de posibles aplicaciones que se abre con la resolución fiable y eficiente de la detección, localización y seguimiento de caras, es prácticamente ilimitado. Una vez resueltos estos problemas preliminares, la extracción de información facial permite enriquecer el modo en el que las personas se comunican con las máquinas. A lo largo del presente capítulo hemos abordado dos aplicaciones concretas en el dominio del análisis de caras: la clasificación de expresiones faciales, y la estimación de pose orientada al uso en un interface perceptual.

Podemos señalar algunos de los aspectos más relevantes de las aportaciones realizadas en ambos problemas:

- Se ha propuesto un método extensible para el **análisis de expresiones** faciales, basado en la obtención de integrales proyectivas asociadas a partes predefinidas de la cara. Con los resultados del seguimiento, el proceso delimita las regiones de ojos y boca, y calcula las proyecciones verticales de las mismas. Estas señales se someten a clasificadores multiclase, que son los encargados de seleccionar la unidad de activación más probable para cada componente del rostro.

- Hemos sugerido e implementado varios **métodos de clasificación** alternativos, basados en diferentes formas de modelar la distribución de probabilidad condicionada de las clases: mediante distancia a la media, vecino más próximo, y con k medias. Aunque el segundo método resulta más adecuado cuando las condiciones de uso son similares a las de entrenamiento, el primero demuestra una mayor capacidad de generalización frente a los cambios de iluminación. En cualquier caso, los mejores ratios de clasificación correcta se encuentran típicamente entre el 85 % y el 92 %.

- El resultado del análisis de expresiones se aplica en un sistema sencillo de **generación de avatares**. Algunas cuestiones quedan abiertas de cara a la mejora de este sistema: aumentar el número de unidades de activación permitido; utilizar las probabilidades de pertenencia a las clases como una forma de graduar las expresiones; introducir otros métodos de clasificación que permitan un entrenamiento para múltiples usuarios; y estudiar el uso de las proyecciones en combinación con otras técnicas de análisis.

- La segunda aplicación abordada es la construcción de un **interface perceptual**, controlado con los **giros y movimientos faciales** del usuario. Partiendo de las posiciones devueltas por el seguidor de caras, y aprovechando las integrales proyectivas calculadas en el proceso, es posible derivar métodos para estimar los 6 parámetros de pose. Algunos de ellos tienen una justificación teórica más sólida y otros son simples criterios heurísticos basados en observaciones experimentales.

- Para contrastar los resultados del estimador de pose, se ha creado un método alternativo basado en suponer una **posición fija del usuario** a lo largo de la secuencia. El nivel de respuesta y fiabilidad para ángulos reducidos resulta muy interesante. El modo óptimo de trabajo se encuentra entre los $\pm 40^{o}$ para las rotaciones en sentido horizontal y los $\pm 20^{o}$ en vertical; no hay un límite para la inclinación máxima permitida.

- Los parámetros de pose se traducen en señales de control para la **navegación en un entorno virtual**. Hemos realizado algunas consideraciones orientadas a mejorar la estabilidad y robustez del sistema que, en general, producen un nivel de usabilidad bastante satisfactorio. La aplicación creada está disponible para el dominio público.

CAPÍTULO 8

"Visiones del Quijote", Octavio Ocampo, 1989

Conclusiones y Perspectivas

> *"El arquitecto meramente práctico no es capaz de asignar las razones suficientes para las formas que él adopta; y el arquitecto de teoría falla también, agarrando la sombra en vez de la substancia. El que es teórico así como también práctico, construyó doblemente; es capaz no sólo de probar la conveniencia de su diseño, sino igualmente de llevarlo en ejecución."*
>
> MARCO VITRUVIO, *De Architectura*, C. 27 AC.

Hoy por hoy, el procesamiento de caras humanas sigue siendo una de las áreas más activas y atractivas en la comunidad de visión artificial. Como hemos ido discutiendo a lo largo de los diferentes capítulos, la disciplina se encuentra en un estado intermedio entre la madurez –por la aparición de soluciones públicas y bien diseñadas– y la rápida expansión –por el creciente número de nuevos usos prácticos que surgen continuamente–. Hasta la fecha, se han sugerido diversos enfoques para los diferentes problemas que se derivan de este dominio específico; en la presente tesis los hemos abordado desde la perspectiva de las integrales proyectivas.

Partiendo de una sólida base teórica plasmada en el capítulo 2, se ha estudiado exhaustivamente la viabilidad de las proyecciones para resolver de manera fiable, precisa, robusta y eficiente todos los grandes problemas dentro del procesamiento visual de caras: detección en imágenes estáticas, localización de componentes faciales, seguimiento en vídeo, reconocimiento de personas, y extracción de información facial.

La estructura de este último capítulo es la siguiente. En la sección 8.1, sintetizamos los aspectos más relevantes y novedosos de las propuestas introducidas, y esquematizamos las técnicas diseñadas para los diferentes problemas. La sección 8.2 hace una valoración global de los resultados de la amplia serie de experimentos llevados a cabo. Finalmente, en la sección 8.3 vislumbramos algunas líneas de investigación futura en relación con el trabajo aquí descrito.

8.1. Aportaciones y originalidades

Desde un punto de vista genérico, las integrales proyectivas presentan una serie de características que hacen muy interesante su uso en diferentes aplicaciones de visión. De hecho, constituyen una de las herramientas básicas del análisis de imágenes, y ya desde los primeros trabajos del procesamiento de caras tuvieron un papel destacado, [93]. La transformación de integral proyectiva se puede entender como una técnica más de **proyección en subespacios lineales**, donde la base estaría compuesta por todos los segmentos que atraviesan la imagen, con distintas posiciones y orientaciones. En nuestro trabajo hemos puesto un énfasis especial en tratar las proyecciones de manera rigurosa. Podemos destacar las siguientes propiedades fundamentales de las proyecciones:

- Hemos comprobado, tanto de forma teórica como práctica, que la operación de proyección es **invertible**. Esto significa que no hay ninguna pérdida de información implícita al proceso, más que la que se deriva de usar un conjunto reducido de proyecciones.
- Las proyecciones aportan una notable **inmunidad frente al ruido**, al compensar las desviaciones producidas en los píxeles mediante el promediado de valores.
- En contraste con otras técnicas de reducción a subespacios, las integrales proyectivas conservan la **propiedad de vecindad local** entre los píxeles: dos puntos adyacentes de las proyecciones corresponden a regiones próximas en la imagen.
- La anterior propiedad es esencial, ya que de ella se deriva la posibilidad de aplicar **transformaciones** en el dominio y en el valor de las señales obtenidas, que resultarían inviables en técnicas como PCA, LDA o ICA. Estas técnicas son extremadamente sensibles al alineamiento de las imágenes respecto de los patrones definidos. Sin embargo, con las integrales proyectivas el **alineamiento** puede tener lugar a posteriori.

En el plano meramente teórico, la principal novedad de esta tesis consiste en plantear la necesidad de definir un contexto formalizado para el uso de las integrales proyectivas, al igual que sucede en otros dominios de aplicación [149, 95], huyendo de los acercamientos heurísticos usados hasta la fecha. Las aportaciones más claras a este respecto son la introducción de los **modelos de proyección** y la definición del **algoritmo rápido de alineamiento** de proyecciones, basado en un esquema de muestreo y acotación.

Este algoritmo y la definición del mecanismo para el modelado de proyecciones son los elementos que fundamentan en buena parte las soluciones para los problemas abordados en el procesamiento de caras. En concreto, se han desarrollado los siguientes métodos:

- Para el problema de **detección de caras humanas**, se ha diseñado un algoritmo capaz de encontrar un número arbitrario de rostros en situaciones complejas usando exclusivamente integrales proyectivas. El esquema desarrollado se puede calificar como *basado en apariencia 1,5D*. El proceso lleva a cabo una búsqueda exhaustiva multiescala, y consta

de tres grandes pasos: (1) buscar el modelo de PV_{cara} en las proyecciones verticales por tiras de la imagen; (2) verificar los candidatos con el modelo de PH_{ojos}; y (3) agrupar los candidatos resultantes.

- Tras la detección, se ha abordado el problema de **localización de componentes faciales**. El método diseñado usa también integrales proyectivas, pero evitando los procesos *ad hoc* de análisis de máximos y mínimos de otros trabajos previos. En esencia, el algoritmo se puede definir como un *ajuste fino* de los modelos de proyección, basado en la aplicación del algoritmo de alineamiento. Además, se ha propuesto una forma robusta de aprovechar la simetría de la cara para resolver el problema de estimar la inclinación del rostro en la imagen.

- Un esquema similar al de la localización se ha aplicado exitosamente para el caso del **seguimiento en secuencias** de vídeo. Las principales diferencias radican en dos aspectos: los modelos de proyección con los que se lleva a cabo el seguimiento se obtienen a partir de la propia secuencia; y el paso de estimar la inclinación se realiza al final del proceso de relocalización.

  El seguimiento implica también una **predicción**; a este respecto, sólo el método basado en color ha proporcionado unos resultados satisfactorios. Con el esquema de combinación propuesto, el seguimiento por color no es alternativo o contradictorio con el basado en proyecciones, sino que ambos se complementan para lograr lo mejor de cada uno: la adaptación a movimientos rápidos del primero y la precisión del segundo.

- Hemos realizado una incursión superficial en el ámbito del **reconocimiento facial de personas**, siempre desde la perspectiva de las integrales proyectivas. Las muestras biométricas tomadas para cada persona consisten en un conjunto de proyecciones obtenidas de la misma. La medida de similitud entre muestras aprovecha el algoritmo de alineamiento para conseguir invarianza a traslación, escala y niveles de brillo.

- Apoyándonos en los métodos de detección, localización y seguimiento, se han desarrollado dos **aplicaciones** finales de procesamiento y análisis de caras. La primera lleva a cabo un **análisis de expresiones faciales** mediante proyecciones, con el propósito de construir un sistema sencillo de generación de avatares. La segunda introduce una serie de métodos aproximados para la **estimación de pose 3D**, cuyo resultado es aplicado en un *interface* perceptual. Ambos son ejemplos simples, pero que tratan de demostrar la viabilidad de los algoritmos propuestos y el uso adicional de las proyecciones para resolver otros tipos de problemas.

Todas las propuestas realizadas en relación a los diversos problemas del análisis de caras han sido implementadas de forma práctica, haciendo uso de una librería cada vez más extendida en la comunidad de visión artificial como es Intel OpenCV [35]. Contando sólo las

funciones de procesamiento de caras desarrolladas, se han escrito unas 23.000 líneas de código C/C++. A esto hay que añadir unas 12.000 líneas en una veintena de aplicaciones creadas en el entorno visual de Borland C++ Builder, para la ejecución de los experimentos (sin contar las aplicaciones de desarrollo, depuración y prueba, y las escritas específicamente para la documentación de esta tesis). Creemos que merece la pena aprovechar este enorme esfuerzo, dejando la mayor parte del código a disposición del dominio público. No obstante, esta orientación al uso de terceros implica un trabajo adicional –simplificar los interfaces, eliminar ramas que han sido desechadas, y documentar el uso de las librerías–, que se perfila como una labor para un futuro inmediato.

8.2. Valoración de los resultados experimentales

Globalmente, los extensos experimentos llevados a cabo han demostrado la excelente capacidad de las integrales proyectivas para conservar la información relevante en los diversos problemas del dominio específico del procesamiento visual de caras. En condiciones equiparables, exhiben unos niveles de generalización y de discriminación muy superiores a las *autocaras* y al uso de patrones 2D. Valorando brevemente los diferentes problemas, podemos señalar los siguientes resultados:

- En el problema de **detección de caras**, las proyecciones alcanzan unos ratios comparables a los de métodos más complejos, y siempre muy por encima de la técnica basada en búsqueda de patrones 2D. El algoritmo propuesto es especialmente bueno con fuentes de vídeo como webcam o televisión, con porcentajes próximos al 90 % para 1 falso positivo por cada 5 imágenes. En una imagen típica de 640×480 píxeles, el proceso tarda cerca de 0,1 segundos. Por otro lado, en combinación con otros métodos se alcanzan resultados competitivos con otros trabajos que constituyen el estado del arte.

- La relación coste/precisión del **localizador de componentes faciales** se encuentra claramente por encima del resto de alternativas analizadas. El método diseñado consigue unos errores similares o menores que los otros algoritmos, y es siempre superior en la estimación del ángulo de inclinación y en la localización global del rostro. Y todo esto sin pasar de los 3 milisegundos por imagen. En una base de caras extensa como es FERET [52], el 94 % de los ojos son localizados con un error máximo del 10 % de la distancia interocular (unos 7 milímetros); y el 99,5 % con un máximo del 20 %.

- La fiabilidad, precisión y robustez del método de **seguimiento** es también destacable. En los experimentos se ha hecho especial hincapié en la evaluación de situaciones complejas de posición y orientación 3D, iluminación, expresión facial, baja resolución y movimientos rápidos. En todas ellas alcanza un buen rendimiento, aunque principalmente en las tres últimas.

- En los experimentos de **reconocimiento facial**, las proyecciones vuelven a superar ampliamente al uso de patrones 2D y a las técnicas basadas en autocaras. A pesar de que las proyecciones tienen formas similares para la mayoría de los individuos, siguen conservando información que permite discriminar a unos de otros, incluso cuando existen más de mil sujetos en la galería. En concreto, con las 1200 personas de FERET se sobrepasa el 80 % de identificación correcta. En un escenario más pequeño, con 150 usuarios y varias imágenes por persona, el porcentaje llega fácilmente al 99,5 %; para el caso de verificación, el ratio de error igual se sitúa en el 0,2 %.

- La evaluación de las aplicaciones de **extracción de información facial** no ha sido tan exhaustiva como la del resto de problemas. A través de una serie de ensayos reducidos se ha comprobado la viabilidad de ambos sistemas, aunque es evidente que los dos son susceptibles de ampliación y mejora. El reconocedor de expresiones alcanza en torno al 92 % de clasificación correcta en condiciones similares a las de entrenamiento, y sobre el 85 % con distintas condiciones de iluminación. Por su parte, el estimador de pose resulta fiable en los parámetros de posición, profundidad, inclinación y giro lateral. El cálculo del giro vertical es más impreciso, pero se puede tomar como una indicación *grosso modo* del estado de mirada arriba/abajo.

8.3. Vías futuras de investigación

Adoptando un punto de vista más exigente, está claro que todos los métodos desarrollados permiten unos márgenes de variación –en la expresión, orientación, iluminación, etc.– fuera de los cuales fallarán con mucha probabilidad. En consecuencia, las posibilidades de mejora son aún grandes, y pensamos que queda un largo camino para la consecución de sistemas de detección, localización, seguimiento y reconocimiento absolutamente fiables.

Algunas de las líneas futuras de trabajo que podemos vislumbrar están relacionadas con esta mejora de las técnicas propuestas. Ya hemos adelantado algunas de ellas dentro de los diferentes capítulos. Otras vías se derivan de la generalización de los mecanismos diseñados a otros contextos. En definitiva, podemos mencionar las siguientes líneas:

- **Combinación de métodos.** En todo el desarrollo de esta tesis ha sido nuestro empeño demostrar la capacidad de las integrales proyectivas de resolver por sí solas los diversos problemas planteados del procesamiento y análisis de caras. De esta manera, hemos puesto de relieve su potencia expresiva. No obstante, no pretendemos afirmar que el uso exclusivo de proyecciones ofrezca la solución óptima para todos los problemas. De hecho, en muchos casos hemos podido comprobar las mejoras sustanciales que pueden ofrecer los mecanismos de combinación de técnicas. Podría ser interesante profundizar en estos sistemas combinados, con el fin de aumentar la robustez frente a fallos esporádicos de los métodos elementales.

- **Modelado de proyecciones.** Los simples modelos de proyección media, y media/varianza, han exhibido una sorprendente capacidad de discriminación y generalización. Sin embargo, pensamos que para conseguir una mejora significativa de los resultados será necesario utilizar métodos más avanzados de modelado. El principal obstáculo de los modelos de media es que son unimodales, mientras que las proyecciones pueden adoptar realmente distribuciones multimodales. Una vía prometedora es la aplicación de PCA sobre las proyecciones. El modelo admitiría así diferentes modos de variación, haciendo uso de la distancia al autoespacio (DFFS) y dentro del autoespacio (DIFS).

  El aspecto clave es hacer compatible esta técnica con la medida de distancia señal/modelo y con el algoritmo de alineamiento de proyecciones. Puesto que estos son los elementos en los que se apoyan las aplicaciones posteriores, las implementaciones de los métodos de detección, localización, seguimiento y reconocimiento de caras, no se verían afectadas en absoluto.

- **Otros tipos de proyecciones y métodos de clasificación.** La simetría facial explica la invarianza y el poder expresivo de las proyecciones verticales de la cara y las horizontales de los ojos. Pero no está claro que ésta sea la elección óptima. En el capítulo 7 estudiamos los resultados del reconocimiento facial en función de las proyecciones utilizadas, comprobando que la elección de una u otra puede ser relevante. Sería adecuado desarrollar métodos para la selección de un conjunto óptimo de integrales proyectivas en los diferentes problemas, desde la detección hasta la estimación de pose. También se podrían poner en juego las proyecciones de imágenes de bordes y de la varianza, dejando en manos del mecanismo automático la búsqueda de la combinación más acertada.

  Con una inspiración en el algoritmo AdaBoost [188], cada posible proyección (según el tipo, el ángulo y la región proyectada) constituiría un *clasificador débil*. Un proceso iterativo sería el encargado de producir un clasificador combinado óptimo, en base a ejemplos de proyecciones de cara y de no-cara. Los ejemplos se ponderarían según la dificultad observada para clasificar los mismos. Un mecanismo de este tipo permitiría también una variación no unimodal de las señales.

- **Aplicación a otros dominios.** Estamos convencidos de que las técnicas desarrolladas en esta tesis pueden ser extendidas a problemas similares. No obstante, la aplicación no es inmediata, porque algunas de las consideraciones introducidas –como usar la proyección de los ojos– son específicas del dominio facial. Una posible aplicación es la detección y seguimiento de rostros de perfil. Hemos llevado a cabo algunos experimentos de tamaño reducido que demuestran que el acercamiento desarrollado puede ser trasladado con relativa facilidad. La generalización de los métodos propuestos se presenta, en consecuencia, como otra posible vía de investigación futura.

Referencias

"Si he visto más lejos es porque estoy
sentado sobre los hombros de gigantes."
ISAAC NEWTON

Muchas de las caras utilizadas en los experimentos de esta tesis, y entre ellas algunas de las que aparecen en el presente documento, pertenecen a compañeros, familiares y amigos que han cedido voluntariamente su imagen para la investigación en procesamiento facial. Quiero agradecer especialmente la colaboración de **Begoña Moros**, **Cristina Vicente**, **Sergio Fructuoso**, **Pablo García**, **Luis Miguel García**, **José Luis Fernández**, **Juan José Vera** y **Diego Sevilla**. Agradezco también a **Cristina Vicente**, **Marcos Menárguez** y **Juan Francisco García** por su participación en diversos experimentos, y a **Pablo García Cano** por la cesión de material para el desarrollo de algunos ensayos.

Otra gran parte de las imágenes usadas han sido donadas por diversos investigadores o grupos de trabajo, ya sea de forma particular o pública. Debo hacer una referencia muy especial a los siguientes:

- **José Miguel Buenaposada**, de la Universidad Rey Juan Carlos, por la cesión de secuencias para las pruebas de seguimiento facial [19], y por el permiso para utilizar su imagen en algunas de las figuras de esta memoria.

- **Libor Spacek**, de la Universidad de Essex, por la disponibilidad de las imágenes de caras para la evaluación del reconocimiento [82], y por el permiso para utilizar algunas de ellas en este documento.

- También se han usado imágenes y secuencias de vídeo ofrecidas públicamente por el ARL (base FERET) [52], Olivetti Research Laboratory (base ORL) [159], Henry Rowley (base CMU/MIT) [152], Instituto Tecnológico de Georgia (base GATECH) [127], Dmitry

O. Gorodnichy (base de vídeos faciales del NRC-ITT) [70], Tim Cootes (software de modelos AAM) [32, 34], y otros que han sido usados en diferentes figuras de la tesis.

Finalmente, otra parte de las imágenes utilizadas en los experimentos han sido extraídas de fuentes públicas de televisión, españolas o extranjeras (TVE, Antena 3, Tele 5, Telemadrid, Canal Sur, Cuatro, BBC Word, Fashion TV, A3 Neox, etc.), de diversas películas ("Trece fantasmas", "Charlie y la fábrica de chocolate", "La Guerra de las Galaxias" y "Matrix") y de Internet. Todas estas imágenes son propiedad de sus respectivos dueños. El autor de esta tesis no se reserva ningún derecho sobre su utilización.

Este documento ha sido escrito en LaTeX, a partir de una plantilla proporcionada por **Pedro Enrique López de Teruel**, a quien quiero agradecer su amable ayuda técnica. Debo dar las gracias especialmente a **Alberto Ruiz** por la ingente pero cuidadosa labor de revisar los borradores previos de esta tesis. Igualmente, quiero reconocer la participación de ambos como coautores de algunas publicaciones relacionadas con la investigación aquí descrita, así como también la de **Cristina Vicente**, **Sergio Fructuoso** y **Andrés García**.

De forma más general, es también justo reconocer la inestimable aportación de los proyectos de software libre y código abierto, fundamentalmente en relación a los métodos alternativos usados en los diversos experimentos. Entre las funciones que han sido utilizadas de las librerías Intel® OpenCV [35], debo destacar los trabajos de Dmitry Abrosimov (manejo de autoespacios), Gary Bradski (algoritmo CamShift), Jean-Yves Bouguet (algoritmo piramidal de Lucas y Kanade), Tatiana Cherepanova, Vasily Khudyakov, Alexander Kuranov (procesamiento de caras con contornos), Rainer Lienhart (detector de caras de Haar+AdaBoost), Sergey Molinov y Ara Nefian (reconocimiento facial con HMM); también a Henry Rowley [152], por el detector de caras mediante redes neuronales.

Partes de la investigación llevada a cabo en esta tesis han sido financiadas por los proyectos TIC98-0559 y DPI2001-0469-C03-01 del Ministerio de Ciencia y Tecnología, y por el contrato 6298 de la Universidad de Murcia.

Bibliografía

"La multiplicidad de libros es un gran mal. No hay medida o límite para esta fiebre: todo el mundo quiere ser autor de algo."
MARTÍN LUTERO

"Fuera del perro, los libros son el mejor amigo del hombre. Y dentro del perro, está demasiado oscuro para leer."
GROUCHO MARX

[1] J. Ahlberg. *Model-based Coding–Extraction, Coding and Evaluation of Face Models Parameters*. PhD thesis, Linkoping University, Sweden, 2002.

[2] Y. Amit, D. Geman, and B. Jedynak. Efficient focusing and face detection. *Face Recognition: From Theory to Applications*, 163:124–156, 1998.

[3] J. Ashbourn. *Biometrics: Advanced Identity Verification*. Springer-Verlag, 2000.

[4] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002.

[5] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 1981.

[6] M. S. Bartlett, H. M. Lades, and T. Sejnowski. Independent component representation for face recognition. In *SPIE Symposium on Electronic Imaging: Science and Technology*, pages 528–539, 1998.

[7] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head-tracking. In *International Conference on Computer Vision and Pattern Recognition*, pages 611–616, Vienna, 1996.

[8] P.N. Belhumeur and G.D. Hager. Tracking in 3D: Image variability decomposition for recovering object pose and illumination. *Pattern Analysis and Applications*, 2:82–91, 1999.

[9] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[10] L.M. Bergasa, M. Mazo, A. Gardel, M.A. Sotelo, and L. Boquete. Unsupervised and adaptive gaussian skin-color model. *Image and Vision Computing*, 18:987–1003, 2001.

[11] International Biometric Group. Homepage. URL: `http://www.biometricgroup.com/`.

[12] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[13] D. Blackburn, M. Bone, and P.J. Phillips. Face Recognition Vendor Test 2000. Technical Report, 2001. Disponible en: `http://www.frvt.org`.

[14] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3D morphable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 202–207, 2002.

[15] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Intel Corporation, Microprocessor Research Labs, 2000.

[16] G.D. Bradsky. Computer vision face tracking as a component of a perceptual user interface. In *Workshop on Applications of Computer Vision*, pages 214–219, Princeton, 1998.

[17] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Expression-invariant 3D face recognition. In *4th Intl. Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA)*, pages 62–69, 2003.

[18] R. Brunelli and T. Poggio. Face recognition: Features vs. templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[19] J.M. Buenaposada, E. Muñoz, and L. Baumela. Efficiently estimating facial expression and illumination in appearance-based tracking. In *Proceedings of BMVC*, Edimburgh, UK, 2006.

[20] J.M. Buenaposada Biencinto. *Análisis de expresiones faciales mediante visión por computador*. PhD thesis, Universidad Politécnica de Madrid, 2004.

[21] J. Cai, A. Goshtasby, and C. Yu. Detecting human faces in color images. In *Proc. Intl. Workshop Multi-Media Database Management*, pages 124–131, 1998.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[23] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000.

[24] M. Castrillón, J. Lorenzo, O. Déniz, J. Isern, and A. Falcón. Multiple face detection at different resolutions for perceptual user interfaces. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Estoril, Portugal, 2005.

[25] D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In *Proc. Third Intl. Conf. Automatic Face and Gesture Recognition*, pages 124–129, 1998.

[26] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–741, 1995.

[27] Y. Chen, Y. Rui, and T. Huang. JPDAF-based HMM for real-time contour tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 543–550, Kauai, Hawaii, 2001.

[28] D. Chetverikov and A. Lerch. Multiresolution face detection. *Theoretical Foundations of Computer Vision*, 69:131–140, 1993.

[29] G. Chow and X. Li. Towards a system for automatic facial feature detection. *Pattern Recognition*, 26:1739–1755, 1993.

[30] A.J. Colmenarez and T.S. Huang. Face detection with information-based maximum discrimination. In *IEEE Conf. Computer Vision and Pattern Recogn.*, pages 782–787, 1997.

[31] The Biometric Consortium. Homepage. URL: `http://www.biometrics.org`.

[32] T.F. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[33] T.F. Cootes and C.J. Taylor. Active shape models–smart snakes. In *Proc. of British Machine Vision Conference*, pages 266–275, 1992.

[34] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[35] Intel Corporation. The Open Source Computer Vision (OpenCV) Library Homepage. URL: `http://www.intel.com/research/mrl/research/opencv/`.

[36] I.J. Cox, J. Ghosn, and P.N. Yianilos. Feature-based face recognition using mixture-distance. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 209–216, 1996.

[37] I. Craw, H. Ellis, and J. Lishman. Automatic extraction of face features. *Pattern Recognition Letters*, 5:183–187, 1987.

[38] D. Cristinacce and T.F. Cootes. Facial feature detection using AdaBoost with shape constraints. In *Proc. of BMVC2003*, volume 1, pages 231–240, 2003.

[39] J.L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 640–647, 1997.

[40] Y. Dai and Y. Nakano. Face-texture model based on SGLD and its application in face detection in a color scene. *Pattern Recognition*, 29(6):1007–1017, 1996.

[41] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *Intl. Journal Computer Vision*, 37(2):175–185, 2000.

[42] S.R. Deans. *The Radon Transform and Some of Its Applications*. John Wiley & Sons, New York, 1983.

[43] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Intl. Journal Computer Vision*, 38(72):99–127, 2000.

[44] F. Dornaika and J. Ahlberg. Face and facial feature tracking using deformable models. *International Journal of Image and Graphics*, DM-02, 2003.

[45] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972.

[46] N. Duta and A.K. Jain. Learning the human face concept from black and white pictures. In *Proc. Intl. Conf. Pattern Recognition*, pages 1365–1367, 1998.

[47] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Learning to identify and track faces in image sequences. In *Proc. Sixth IEEE Intl. Conf. on Computer Vision*, pages 317–322, 1998.

[48] P. Ekman and W. Friesen. *The Facial Action Coding System: A Technique for the Mesurement of Facial Movement*. Consulting Psychologists Press, San Francisco, 1978.

[49] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[50] G.C. Feng and P.C. Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters*, 19:899–906, 1998.

[51] G.C. Feng and P.C. Yuen. Multi-cues eye detection on gray intensity image. *Pattern Recognition*, 34:1033–1046, 2001.

[52] Programa FERET (Face Recognition Technology). Página principal. URL: `http://www.nist.gov/humanid/feret/`.

[53] B. Fröba and C. Küblbeck. Face detection and tracking using edge orientation information. In *SPIE Visual Communications and Image Processing*, pages 583–594, 2001.

[54] S. Fructuoso Muñoz. Tierra Inhóspita: Desarrollo de un interface perceptual para la navegación en un mundo virtual 3D. Proyecto fin de carrera, dirigido por: G. García Mateos, Universidad de Murcia, 2004.

[55] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[56] Y. Gao and M.K.H. Leung. Face recognition using line edge map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):764–779, 2002.

[57] G. García Mateos. Refining face tracking with integral projections. In *4th Intl. Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA)*, volume LNCS 2688, pages 360–368, Guildford, UK, 2003.

[58] G. García Mateos and C. Vicente Chicote. A new model and process architecture for facial expression recognition. In *Joint IAPR International Workshops SSPR+SPR*, volume LNCS 1876, pages 716–726, Alicante, 2000. Springer.

[59] G. García Mateos and C. Vicente Chicote. Face detection on still images using HIT maps. In *3rd Intl. Conf. on Audio- and Video-based Biometric Person Authentication, AVBPA*, volume LNCS 2091, pages 102–107, Halmstad, Suecia, 2001. Springer-Verlag.

[60] G. García Mateos and C. Vicente Chicote. A unified approach to face detection, segmentation and location using HIT maps. In *IX Spanish Symposium on Pattern Recognition and Image Analysis*, pages 61–66, Benicasim, Castellón, 2001.

[61] G. García Mateos and S. Fructuoso Muñoz. Tierra Inhóspita: Exploring a virtual world with your face. In *ACM-IEEE Int. Conf. on Advances in Computer Entertainment Technology (ACE)*, Valencia, 2005.

[62] G. García Mateos, A. Ruiz García, and P.E. López de Teruel. Face detection using integral projections models. In *Joint IAPR International Workshops SSPR+SPR*, volume LNCS 2396, pages 644–653, Windsor, Canadá, 2002. Springer-Verlag.

[63] G. García Mateos, A. García Meroño, C. Vicente Chicote, A. Ruiz García, and P.E. López de Teruel. Time and date OCR in CCTV video. In *13th International Conference on Image Analysis and Processing*, 2005.

[64] G. García Mateos and S. Fructuoso Muñoz. A perceptual interface using integral projections. In *7th Intl. Conf. on Pattern Recognition and Image Analysis (PRIA)*, San Petersburgo, Rusia, 2004.

[65] J.D. Gaskill. *Linear Systems, Fourier Transforms, and Optics*. John Wiley & Sons, New York, 1978.

[66] R. Göcke, J. Bruce Millar, A. Zelinsky, and J. Robert-Ribes. Automatic extraction of lip feature points. In *Proceedings of the Australian Conference on Robotics and Automation, ACRA2000*, 2000.

[67] S.B. Gokturk, J.Y. Bouguet, and R. Grzeszczuk. A data-driven model for monocular face tracking. In *Proc. IEEE International Conference on Computer Vision*, pages 701–708, 2001.

[68] S. Gong, S. J. McKenna, and A. Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, 2000.

[69] D.O. Gorodnichy. Associative neural networks as means for low-resolution video-based recognition. In *International Joint Conference on Neural Networks (IJCNN'05)*, Montreal, Quebec, Canada, 2005. NRC 48217.

[70] D.O. Gorodnichy. Video-based framework for face recognition in video. In *Second Workshop on Face Processing in Video (FPiV'05) in Proc. of 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 330–338, Victoria, BC, Canada, 2005. NRC 48216.

[71] V. Govindaraju. Locating human faces in photographs. *Intl. Journal Computer Vision*, 19(2):129–146, 1996.

[72] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *Proc. First Intl. Workshop on Automatic Face and Gesture Recognition*, pages 41–46, 1995.

[73] S. R. Gunn and M. S. Nixon. A dual active contour for head and boundary extraction. In *IEEE Colloquium on Image Processing for Biometric Measurement*, London, 1994.

[74] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[75] P. Hallinan, G. Gordon, A. Yuille, P. Giblin, and D. Mumford. *Two- and Three-Dimensional Patterns of the Face*. A K Peters Ltd., 1999.

[76] M. Hamouz, J. Kittler, J.K. Kamarainen, P. Paalanen, and H. Kälviäinen. Affine-invariant face detection and localization using GMM-based feature detector and enhanced appearance model. In *Proc. of the Sixth IEEE Intl. Conference on Automatic Face and Gesture Recognition (FGR'04)*, 2004.

[77] B. Han, C. Yang, R. Duraiswami, and L. Davis. Bayesian filtering and integral image for visual tracking. In *Worshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Montreux, Switzerland, 2005.

[78] C.-C. Han, H.-Y.M. Liao, K.-C. Yu, and L.-H. Chen. Fast face detection via morphology-based pre-processing. In *Proc. Ninth Intl. Conf. on Image Analysis and Processing*, pages 469–476, 1998.

[79] R.M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[80] R. Hietmeyer. Biometric identification promises fast and secure processing of airline passengers. *The International Civil Aviation Organization Journal*, 55(9):10–11, 2000.

[81] E. Hjelmas and B. Kee Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.

[82] D. Hond and L. Spacek. Distinctive descriptions for face processing. In *Proceedings of 8th British Machine Vision Conference*, pages 320–329, 1997. Base de caras disponible en: `http://cswww.essex.ac.uk/mv/allfaces`.

[83] R. Hoogenboom and M. Lew. Face detection using local maxima. In *IEEE Proc. of 2nd Int. Conf. onAutomatic Face and Gesture Recognition*, pages 334–339, 1996.

[84] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain. Face detection in color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

[85] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Systems, Man, and Cybernetics*, 34(3):334–352, 2004.

[86] J. Huang, D. Li, X. Shao, and H. Wechsler. Pose discriminiation and eye detection using support vector machines (SVM). In *Proceeding of NATO-ASI on Face Recognition: From Theory to Applications*, 1998.

[87] M. Isard and A. Blake. Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):2–28, 1998.

[88] T.S. Jebara and A. Pentland. Parameterized structure from motion for 3D adaptive feedback tracking of faces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 144–150, 1997.

[89] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the Hausdorff distance. *Lecture Notes in Computer Science*, 2091:90–95, 2001.

[90] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 274–280, 1999.

[91] K. Jonsson, J. Kittler, Y.P. Li, and J. Matas. Support vector machines for face authentication. *Image and Vision Computing*, 20(5-6):369–375, 2002.

[92] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME Journal Basic Engineering*, 82D:25–46, 1960.

[93] T. Kanade. *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, 1973.

[94] D.G. Kendall. Shape manifolds, procrustean metrics, and complex projective shapes. *Bull. London Math. Society*, 16:81–121, 1984.

[95] A. Khamene, R. Chisu, W. Wein, N. Navab, and F. Sauer. A novel projection based approach for medical image registration. In *Workshop on Biomedical Image Registration*, pages 247–256, Utrecht, The Netherlands, 2006.

[96] S.-H. Kim, N.-K. Kim, S.C. Ahn, and H.-G. Kim. Object oriented face detection using range and color information. In *Proc. Third Intl. Conf. Automatic Face and Gesture Recognition*, pages 76–81, 1998.

[97] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[98] K.J. Kirchberg, O. Jesorsky, and R.W. Frischholz. Genetic model optimization for Hausdorff distance-based face localization. In *Proc. International ECCV 2002, Workshop on Biometric Authentication*, volume LNCS-2359, pages 103–111, 2002.

[99] T. Kohonen. *Self-Organization and Associative Memory*. Springer, 1989.

[100] S.G. Kong, J. Heo, B.R. Abidi, J. Paik, and M.A. Abidi. Recent advances in visual and infrared face recognition–a review. *Computer Vision and Image Understanding*, 97((2005)):103–135, 2005.

[101] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. In *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, volume 4, pages 2537–2540, 1997.

[102] Y.H. Kwon and N. da Vitoria Lobo. Face detection using templates. In *Proc. Intl. Conf. Pattern Recognition*, pages 764–767, 1994.

[103] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers*, 42(3):300–311, 1993.

[104] K.M. Lam and H. Yan. Locating and extracting the eye in human face images. *Pattern Recognition*, 29:771–779, 1996.

[105] A. Lanitis, C.J. Taylor, and T.F. Cootes. An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.

[106] H.S. Lee, D. Kim, and S.Y. Lee. Robust face-tracking using color and facial shape. In *4th Intl. Conf. on Audio- and Video-Based Biometric Person Authentication*, volume LNCS 2688, pages 302–309, 2003.

[107] T.K. Leung, M.C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. Fifth IEEE Intl. Conf. Computer Vision*, pages 637–644, 1995.

[108] S.Z. Li and A.K. Jain. *Handbook of Face Recognition*. Springer, New York, 2005.

[109] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, Intel Labs, December 2002.

[110] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *IEEE Int. Conf. on Image Processing, ICIP 2002*, volume 1, pages 900–903, 2002.

[111] S. H. Lin, S. Y. Kung, and L. J. Linn. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. on Neural Networks*, 8:114–132, 1997.

[112] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh. Fast object detection with oclussions. In *Proc. of 8th Eurpean Conf. on Computer Vision*, volume 1, pages 402–413, 2004.

[113] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:570–582, 2000.

[114] X. Lu. Image analysis for face recognition, 2004. Disponible públicamente en la URL: `http://www.cse.msu.edu/~lvxiaogu/publications/publications.htm`.

[115] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[116] M.J. Lucena, J.M. Fuertes, and N.P. de la Blanca. Real-time tracking using multiple target models. In *IbPRIA 2005*, volume LNCS 3522, pages 20–27, 2005.

[117] K.V. Mardia and I.L. Dryden. Shape distributions for landmark data. *Advanced Applied Probability*, 21:742–755, 1989.

[118] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., New York, 1982.

[119] A.R. Martínez. Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:748–763, 2002.

[120] A.R. Martínez and R. Benavente. The AR face database. Technical Report 24, Centro de Visión por Computador (CVC), Barcelona, 1998. Disponible en la URL: `http://rv11.ecn.purdue.edu/~aleix/aleix_face_DB.html`.

[121] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[122] P.S. Maybeck. *Stochastic Models, Estimation, and Control*, volume I. Academic Press, New York, 1979.

[123] J. Meek. "Robo cop", UK Guardian newspaper, 2002-06-13. URL: `http://www.guardian.co.uk/Archive/Article/0,4273,4432506,00.htm`.

[124] J. Miao, B. Yin, K. Wang, L. Shen, and X. Chen. A hierarchical multiscale and multi-angle system for human face detection in a complex background using gravity-center template. *Pattern Recognition*, 32(7):1237–1248, 1999.

[125] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[126] A. Nefian and M.H. Hayes. Face recognition using an embedded HMM. In *Intl. Conf. on Audio- and Video-Based Biometric Person Authentication*, pages 19–24, 1999.

[127] A. Nefian and M.H. Hayes. Maximum likelihood training of the embedded HMM for face detection and recognition. In *International Conference on Image Processing*, 2000.

[128] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. 2D cascaded AdaBoost for eye localization. In *The 18th International Conference on Pattern Recognition (ICPR'06)*, 2006.

[129] K. Okada, K.J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. von der Malsburg. The Bochum/USC face recognition system. In H. Wechsler, P.J. Phillips, and otros, editors, *Face Recognition: From Theory to Applications*. Springer-Berlag, Berlin, 1998.

[130] N. Oliver, A. Pentland, and F. Berard. LAFTER: Lips and face real time tracker. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 123–129, 1997.

[131] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *IEEE Proc. of Intl. Conf. on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[132] V. Pahor and S. Carrato. A fuzzy approach to mouth corner detection. In *Proc. of ICIP-99*, volume I, pages 667–671, Kobe, Japan, 1999.

[133] I.S. Pandzic and R. Forchheimer, editors. *MPEG-4 Facial Animation: The Standard, Implementations, and Applications*. Wiley, Chichester, 2002.

[134] C. Papageorgiou and T. Poggio. A trainable system for object recognition. *Intl. Journal Computer Vision*, 38(1):15–33, 2000.

[135] P.S. Penev and J.J. Atick. Local feature analysis: a general statistical theory for object representation. *Netw. Comput. Neural Systems*, 7(3):477–500, 1996.

[136] A. Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):107–119, 2000.

[137] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[138] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition (CVPR)*, pages 947–954, 2005.

[139] P.J. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and J. Bone. Face Recognition Vendor Test 2002: Evaluating report. Technical Report NISTIR 6965, National Institute of Standards and Technology, 2003. Disponible en: `http://www.frvt.org`.

[140] P.J. Phillips, A. Martin, C.L. Wilson, and M. Przybocki. An introduction to evaluating biometric systems. *Computer*, 33:56–63, 2000.

[141] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[142] P.J. Phillips and E. M. Newton. Meta-analysis of face recognition algorithms. In *5th IEEE Conference on Automatic Face and Gesture Recognition*, Washington DC, 2002.

[143] P.J. Phillips, P.J. Rauss, and S.Z. Der. FERET (face recognition technology) recognition algorithm development and test results. Technical Report: ARL-TR-995, Army Research Laboratory, 1996.

[144] P.J. Phillips, H. Wechsler, J. Huang, and Patrick J. Rauss. The FERET database and evaluating procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[145] A. Pikaz and I. Dinstein. An algorithm for polygonal approximation based on iterative point elimination. *Pattern Recognition Letters*, 16(6):557–563, 1995.

[146] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte langs gewisser manigfaltigkeiten. *Ber. Ver. Sächs. Akad. Wiss. Leipzig, Math-Phys. Kl.*, 69:262–277, 1914. (En alemán, traducido al inglés en *S.R. Deans: The Radon Transform and Some of Its Applications*.).

[147] M.J.T. Reinders, R.W.C. Koch, and J.J. Gerbrands. Locating facial features in image sequences using neural networks. *Proc. of the 2nd Intl. Conference on Automatic Face and Gesture Recognition*, page 230, 1996.

[148] D. Reisfeld and Y. Yeshurun. Robust detection of facial features by generalised symmetry. In *Proc. of 11th Int. Conf. on Pattern Recognition*, pages 117–120, The Hague, The Netherlands, 1992.

[149] D. Robinson and P. Milanfar. Fast local and global projection-based methods for affine motion estimation. *Journal of Mathematical Imaging and Vision, Kluwer Academic Publishers*, 18:35–54, 2003.

[150] S. Romdhani, P. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. In *IEEE Intl. Conf. on Computer Vision*, volume 2, pages 695–700, 2001.

[151] D. Roth. Learning to resolve natural language ambiguities: A unified approach. In *Proc. 15th Natl. Conf. Artificial Intelligence*, pages 806–813, 1998.

[152] H.A. Rowley. *Neural Network-Based Face Detection*. PhD thesis, Carnegie Mellon University, 1999.

[153] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–28, 1998.

[154] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 786–793, Kauai, Hawaii, 2001.

[155] A. Ruiz García, P.E. López de Teruel, and G. García Mateos. A note on principal point estimability. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 304–307, Quebec, Canada, 2002.

[156] Y.-S. Ryu and S.-Y. Oh. Automatic extraction of eye and mouth fields from a face image using eigenfeatures and multilayer perceptrons. *Pattern Recognition*, 34:2459–2466, 2001.

[157] E. Saber and A.M. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 17(8):669–680, 1998.

[158] T. Sakai, M. Nagao, and S. Fujibayashi. Line extraction and pattern detection in a photograph. *Pattern Recognition*, 1:233–248, 1969.

[159] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *2nd IEEE Workshop on Applications of Computer Vision*, Saratosa, FL, 1994. URL: `http://www.uk.research.att.com/facedatabase.html`.

[160] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proc. Second Intl. Conf. Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[161] B. Scassellati. Eye finding via face detection for a foevated, active vision system. In *Proc. 15th Natl. Conf. Artificial Intelligence*, 1998.

[162] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 746–751, 2000.

[163] K. Schwerdt and J.L. Crowley. Robust tracking and compression for video communication. In *Proc. of Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time (RATFG'99)*, pages 2–9, 1999.

[164] S. Shan, B. Cao, W. Gao, and D. Zhao. Extended fisherface for face recognition from a single example image per person. In *IEEE Symp. Circ. Syst.*, volume 2, pages 81–84, 2002.

[165] F. Simion, V.M. Cassia, C. Turati, and E. Valenza. The origins of face perception: Specific versus non-specific mechanisms. *Infant and Child Development*, 10(59), 2001.

[166] P. Sinha. *Processing and Recognizing 3D Forms*. PhD thesis, Massachusetts Inst. of Technology, 1995.

[167] S.A. Sirohey. Human face segmentation and identification. Technical report, CS-TR-3176, Univ. of Maryland, 1993.

[168] L. Sirowich and M. Kirby. Low-dimensional procedure for the characterization of human face. *Journal Opt. Soc. Am.*, 4:519–524, 1987.

[169] K. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *Proc. of 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, pages 236–241, Killington, Vermont, USA, 1996. IEEE Computer Society.

[170] K. Sobottka and I. Pitas. Looking for faces and facial features in color images. *PRIA: Advances in Mathematical Theory and Applications*, 7(1), 1997.

[171] M.B. Stegmann, B.K. Ersboll, and R. Larsen. FAME–a flexible appearance modeling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, 2003.

[172] J. Ström. Model-based real-time head tracking. *EURASIP Journal on Applied Signal Processing*, 10:1039–1052, 2002.

[173] K.-K. Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, Massachusetts Inst. of Technology, 1996.

[174] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[175] X. Tang, Z. Ou, T. Su, H. Sun, and P. Zhao. Robust precise eye location by Adaboost and SVM techniques. *Lecture Notes in Computer Science*, 3497/2005:93–98, 2005.

[176] J.C. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and invariant moments. In *Proc. Third Intl. Conf. Automatic Face and Gesture Recognition*, pages 112–117, 1998.

[177] P. Thompson. Margaret Thatcher: A new illusion. *Perception*, 9(4):483–484, 1980.

[178] P. Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Technical University of Denmark, 1996.

[179] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[180] K. Toyama. Prolegomena for robust face tracking. In *Workshop on Automatic Facial Image Analysis and Recognition Technology (ECCV '98)*, 1998.

[181] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, New Jersey, 1998. páginas 198-203.

[182] A. Tsukamoto, C.-W. Lee, and S. Tsuji. Detection and pose estimation of human face with synthesized image models. In *Proc. Intl. Conf. Pattern Recognition*, pages 754–757, 1994.

[183] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, pages 586–590, Maui, Hawaii, 1991.

[184] S. Ullman. *High-level Vision: Object Recognition and Visual Cognition*. The MIT Press, 1996.

[185] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[186] M. Venkatraman and V. Govindaraju. Zero crossings of a non-orthogonal wavelet transform for object location. In *Proc. IEEE Intl. Conf. Image Processing*, volume 3, pages 57–60, 1995.

[187] C. Vicente Chicote, G. García Mateos, and A. García Meroño. Seguimiento de caras humanas mediante búsqueda logarítmica en rejilla. In *III Jornadas Regionales de Informática Gráfica*, pages 75–86, Jaén, 2002. DISTEC S.L.

[188] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Intl. Conf. on Computer Vision and Pattern Recognition, CVPR 2001*, pages 12–14, Kauai, Hawaii, 2001.

[189] P. Walsh. *Advanced 3D Game Programming with DirectX 9.0*. Wordware, 2003.

[190] J. Wilder. Face recognition using transform coding of gray scale projection projections and the neural tree network. In R. J. Mammone, editor, *Artificial Neural Networks with Applications in Speech and Vision*, pages 520–536. Chapman Hall, 1994.

[191] R. Willing. "Airport anti-terror systems flub tests face-recognition technology fails to flag suspects", USA Today, 2003-09-02. URL: `http://www.usatoday.com/usatonline/20030902/5460651s.htm`.

[192] L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[193] C. Wong, D. Kortenkamp, and M. Speich. A mobile robot that recognises people. In *IEEE Int. Conf. on Tools with Artificial Intelligence*, 1995.

[194] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multiview face detection based on real AdaBoost. In *Proc. of 6th Intl. Conf. on Automatic Face and Gesture Recognition*, pages 79–84, 2004.

[195] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(6):557–563, 1999.

[196] J. Xiao, T. Moriyama, T. Kanade, and J.F. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85–94, 2003.

[197] F. Xue and X. Ding. 3D+2D face localization using boosting in multi-modal feature space. In *The 18th International Conference on Pattern Recognition (ICPR'06)*, 2006.

[198] G. Yang and T. S. Huang. Human face detection in complex background. *Pattern Recognition*, 27(1):53–63, 1994.

[199] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Real-time face and facial feature tracking and applications. In *Proceedings of AVSP'98*, pages 79–84, Terrigal, Australia, 1998.

[200] M.-H. Yang. Recent advances in face detection. In *IEEE ICPR 2004 Tutorial*, Cambridge, U.K. URL: `http://vision.ai.uiuc.edu/mhyang/papers/icpr04_tutorial.pdf`.

[201] M.-H. Yang and N. Ahuja. Detecting human faces in color images. In *Proc. IEEE Intl. Conf. Image Processing*, volume 1, pages 127–130, 1998.

[202] M.-H. Yang and N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *Proc. SPIE: Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 458–466, 1999.

[203] M.-H. Yang, N. Ahuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *Proc. Fourth Intl. Conf. Automatic Face and Gesture Recognition*, pages 70–76, 2000.

[204] M.-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[205] M.-H. Yang, D. Roth, and N. Ahuja. A SNoW-based face detector. In S.A. Solla, T. K. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems*, pages 855–861. MIT Press, 2000.

[206] H. Yao, W. Gao, J. Li, Y. Lv, and R. Wang. Real-time lip locating method for lip-movement recognition. *Chinese Journal of Software*, 11(8):1126–1132, 2000.

[207] H. Yao, W. Gao, W. Shan, and M.H. Xu. Visual features extracting and selecting for lipreading. In *4th Intl. Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA)*, volume LNCS 2688, pages 251–259, 2003.

[208] T. Yokoyama, Y. Yagi, and M. Yachida. Facial contour extraction model. In *IEEE Proc. of 3rd Int. Conf. on Automatic Face and Gesture Recognition*, 1998.

[209] K.C. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15(9):713–735, 1997.

[210] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *Intl. Journal Computer Vision*, 8(2):99–111, 1992.

[211] S. Zhao and R.R. Grigat. An automatic face recognition system in the near infrared spectrum. In *Proceedings of MLDM*, pages 437–444, Leipzig, Germany, 2005.

[212] W. Zhao, R. Chellapa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[213] Z.-H. Zhou and X. Geng. Projection functions for eye detection. *Pattern Recognition*, 37:1049–1056, 2004.

[214] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98:124–154, 2004.

Índice alfabético