Human Face Processing with 1.5D Models

Ginés García-Mateos[1], Alberto Ruiz[1], and Pedro E. López-de-Teruel[2]

[1] Dept. de Informática y Sistemas
[2] Dept. Ing. y Tecn. de Computadores
Universidad de Murcia, 30.100 Espinardo, Murcia, Spain
{ginesgm,aruiz}@um.es, pedroe@ditec.um.es

Abstract. Integral projections reduce the size of input data by transforming 2D images into significantly simpler 1D signals, while retaining useful information to solve important computer vision problems like object detection, location, and tracking. However, previous attempts typically rely on simple heuristic analysis such as searching for minima or maxima in the resulting projections. We introduce a more rigorous and formal modeling framework based on a small set of integral projections –thus, we will call them *1.5D models*– and show that this model-based analysis overcomes many of the difficulties and limitations of alternative projection methods. The proposed approach proves to be particularly adequate for the specific domain of human face processing. The problems of face detection, facial feature location, and tracking in video sequences are studied under the unifying proposed framework.

Keywords: 1.5D object models, integral projections, face detection, facial feature location, face tracking.

1 Introduction

Dimensionality reduction is a required stage in many computer vision applications. This task is usually carried out with techniques like principal components analysis (PCA) [1], linear discriminant analysis (LDA), independent component analysis (ICA), or other feature extraction methods, such as edge or segment detection. Integral projections are among the most frequently used methods to reduce the huge volume of data contained in images, specially in the human face domain [2]. However, projections are often used just in heuristic and *ad hoc* algorithms [2,3,4,5]. A much more theoretically sound basis can be developed to take full advantage of the intrinsic power of the technique. Two aspects will be crucial to define this framework: first, a simple but powerful *modeling* framework for projections, which is generic and trainable; and second, an efficient and robust technique for the *alignment* of resulting 1D signals.

The rest of the paper is structured as follows. Section 2 describes the concept and properties of integral projections, and tackles the problems of modeling and alignment. A feasible face model using projections is presented in section 3. Then, the problems of human face detection, facial feature location, and face tracking in video are studied in sections 4, 5 and 6, always working with projections.

S.K. Zhou et al. (Eds.): AMFG 2007, LNCS 4778, pp. 220–234, 2007.

Experiments and references to related work are included in each section. Finally, the main contributions of the paper are summarized in section 7.

2 Integral Projections and 1.5D Models

Radon transform [6], Hough transform [7], and integral projections are closely related concepts. Let $f(x, y)$ be a continuous 2D function; its Radon transform is another 2D function, $\mathcal{R}[f](\theta, s)$, defined as:

$$\mathcal{R}[f](\theta, s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - s)\ dx\ dy \tag{1}$$

where δ is a Dirac's delta. Curiously enough, 58 years after being firstly formulated by Johann Radon in 1914, equation 1 was renamed as "Hough transform to detect lines" [7], in this case applied on discrete images.

Moreover, integral projections are a specific case of equation 1, where θ is a fixed projection angle. For example, let $i(x, y)$ be an image; $\mathcal{R}[i](0, y), \forall y$ is called the **vertical integral projection** of i, and $\mathcal{R}[i](\pi/2, x), \forall x$ the **horizontal projection**[1]. For simplicity, we will denote them with $PV_i(y)$ and $PH_i(x)$.

In a pioneering work by Kanade in the human face domain [2], integral projections were applied on edge images. More recently, some alternative methods have been proposed, such as the **variance projection functions** [5], where the variance of each row or column of pixels –instead of just the sum– is computed. Though these also have some interesting advantages, by definition they are not linear transforms; thus, many of the following properties do not hold.

2.1 Properties and Advantages

Compared to most other dimensionality reduction techniques –specially linear subspace methods, like PCA, LDA and ICA–, integral projections are simpler both in definition and computation. Nevertheless, they offer a very interesting set of properties, which make them preferable in many image analysis applications:

- **Invariance and noise filtering.** Projections are invariant to a number of image transformations such as mirroring, scale, shear and translation along the projection angle. It is also well-known that integral projections are highly robust to white noise [8]. Figure 1 shows a sample facial image under several instances of these transformations, and the corresponding vertical projections; it can be easily seen that, while the former are severely distorted, the latter remain greatly unaffected.
- **Locality.** Integral projections preserve the principle of locality of pixels: two neighbor points in a projection correspond to two neighbor regions in the image. This makes it possible to apply alignment processes *after* projection, whereas in PCA, LDA and ICA, images have to be aligned *before* projection.

[1] Some authors call "vertical projection" what we define as "horizontal projection" and *vice versa*, while many others adopt our same definition.

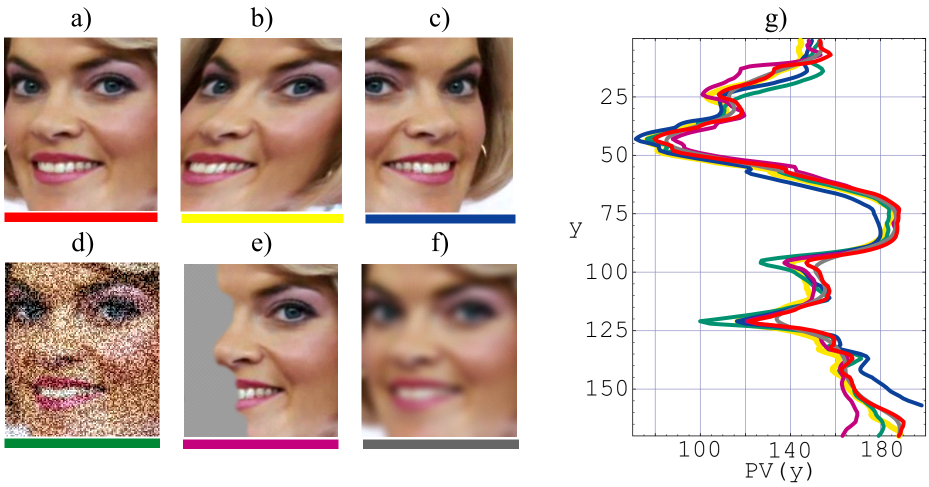

Fig. 1. Invariance of integral projections to various image transforms. a) Original image. b-f) The same image after: b) shear; c) mirroring and translation; d) random noise; e) partial occlusion; f) smoothing. g) The corresponding vertical projections.

In other words, integral projections are less sensitive to miss-alignment in the projected regions.

– **Invertibility.** No information is lost in the process of projection. Using an adequate number of projections at different angles, it is possible to reconstruct original images up to a desired precision level. This result derives from the *central slice theorem* [6], which is the base of computed tomography.
– **Characterization.** According to the previous property, integral projections are preferred when the number of projections needed to represent an object is small. Due to symmetry, this is precisely the case for human faces. Furthermore, projections from very different people share a common underlying structure, as can be seen in figure 2, where images of more than 1000 distinct individuals are projected.
– **Efficiency.** Finally, working with projections is obviously much faster than with full images. Moreover, *integral images* [10] can be used to reduce even more the cost of computing projections, as described below.

2.2 Gaussian Projection Models

Many computer vision systems have benefited from the good properties of integral projections [2,3,4,7]. But, in general, most of them merely analyze the projections heuristically, by searching for local maxima, minima, areas of high variation (maxima of the derivative), or any other similar *ad hoc* techniques. For example, a face is said to be present if the vertical projection presents a minimum in the eyes, a maximum in the nose, and another minimum in the mouth [4]. Considering the highly regular structure of human faces –see figure 2c)– it is clear that a lot of information is being thrown away by these simple heuristic methods.

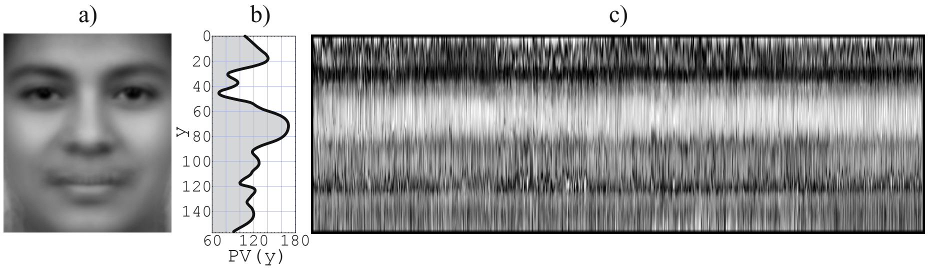

Fig. 2. Characterization of objects with projections. a) Mean male face. b) Vertical integral projection of the mean face (observe the –non-casual– similitude to a profile face). c) 3818 vertical projections of 1196 individuals from the FERET database [9] (each projection is represented as a single column).

To avoid this undesirable loss of information, we propose a mechanism which takes advantage of the whole structure of the projection signal. In particular, let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of training projections from a certain class of objects. We define a *projection model* as a pair of 1D signals (M, V), where:

– $M(i)$ is the mean of the set $\{P_1(i), P_2(i), \ldots, P_n(i)\}$.
– $V(i)$ is the variance of the set $\{P_1(i), P_2(i), \ldots, P_n(i)\}$.

This way, a set $\mathcal{P}$ of integral projections is modeled as m independent 1D gaussians, being m the domain of $\mathcal{P}$ –that is, the size of each input projection vector–. Figure 4a) shows an example of one of these models, corresponding to the vertical projections of a set of images of human faces.

2.3 Projection Alignment

Alignment (both in domain and codomain) between two different 1D signals is a key problem when working with integral projections. Let us define a family of alignment transformations on 1D signals, t_{abcde}. For any projection P, each transformed value is given by:

$$\begin{array}{l} t_{abcde}(P)(i) := a + b \cdot i + c \cdot P(d + e \cdot i) \\ \forall i \in \{(P_{min} - d)/e, \ldots, (P_{max} - d)/e\} \end{array} \tag{2}$$

This is an *affine transformation* of 1D signals in the XY plane. A visual interpretation of the free parameters (a, b, c, d, e) is shown in figure 3.

Alignment of an input projection P to a given projection model (M, V) can be formulated as the optimization of the following expression:

$$\{a^*, b^*, c^*, d^*, e^*\} = \arg \min_{a,b,c,d,e} \frac{1}{||r||} \sum_{i \in r} \frac{(M(i) - t_{abcde}(P)(i))^2}{V(i)} \tag{3}$$

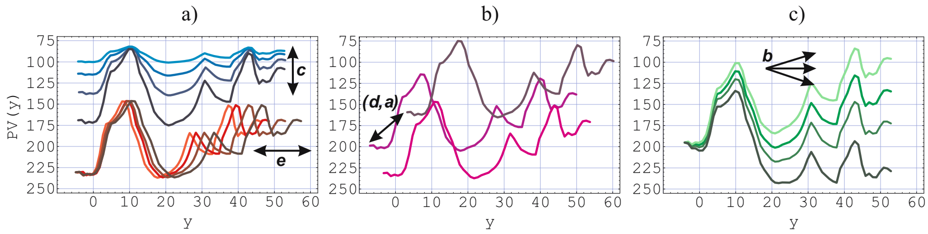

Fig. 3. Interpretation of parameters (a, b, c, d, e) in the alignment function defined in equation 2. a) Scale in signal's value (c) and domain (e). b) Translation in value (a) and domain (d). c) Shear (b), that accounts for non-uniform illumination.

where r is the intersection of domains of M and $t_{abcde}(P)$. For fixed $\{d, e\}$, the minimum squared error solution for the parameters $\{a, b, c\}$ can be obtained in closed form. Then, we can define the following function *mindist*:

$$mindist(d, e) := \min_{a,b,c} dist((M, V), t_{abcde}(P)) \tag{4}$$

where *dist* is the summation term in equation 3. Unfortunately, $\{d, e\}$ cannot be solved analytically. But, by definition, the range of possible values for both parameters is bounded by a maximum and minimum translation, d, and scale, e. Thus, we propose a simplified version of the Nelder-Mead simplex optimization algorithm [11] based on successive sampling and reduction of the search space in the plane $mindist(d, e)$. The algorithm is described in figure 4.

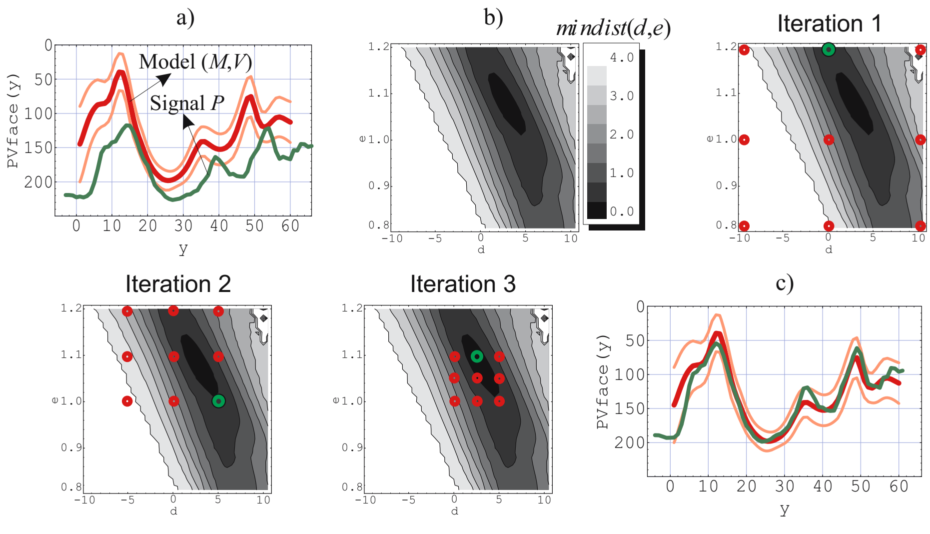

Fig. 4. Illustration of the alignment algorithm. a) Model and signal before alignment. b) Alignment distance, *mindist*, as a function of e and d; and the search space in the first iterations of the algorithm. c) Resulting signal and model after alignment.

3 A 1.5D Face Model

In the rest of this paper we will describe the application of gaussian projection models and the proposed alignment algorithm in the specific domain of face processing. As mentioned before, integral projections are specially adequate for representing human faces, because a very small number of projections is able to retain most information of typical faces. For example, in terms of variance of the gray value of a mean face –see figure 2a)–, the vertical projection alone describes more than 75% of the variance of the original image.

In the following, we will assume that input faces are normalized according to these three rules:

1. Faces are extracted with a predefined resolution of $W \times H$ pixels –typically 24×30– using a *similarity* transform, i.e. a scale/translation/rotation.
2. Faces are horizontally centered, with both eyes at the same height.
3. We set the height of the eyes $h_{eyes} = 0.2H$, and the height of the mouth $h_{mouth} = 0.8H$.

Our face model consists of two integral projection models –thus, we call it *1.5D model*–, which are computed on normalized faces. These models are:

– (MV_{face}, VV_{face}): model of the vertical integral projections of the extracted facial images, PV_{face}.
– (MH_{eyes}, VH_{eyes}): model of the horizontal projections of the eyes' region, PH_{eyes}, approximately between height $0.1H$ and $0.3H$ in extracted images.

Figure 5 shows a sample model computed on 374 faces. Observe the typical patterns of both models, corresponding to dark and light areas of human faces.

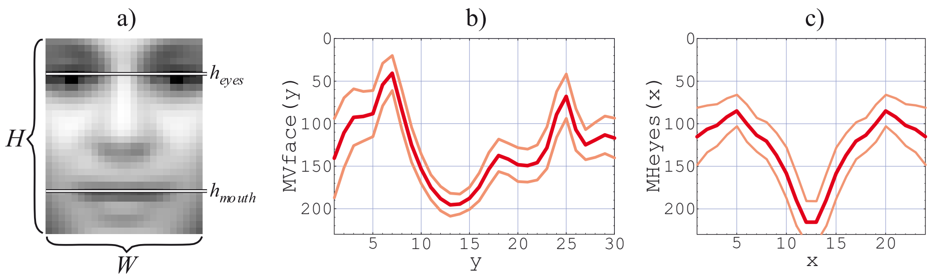

Fig. 5. Face model trained with a set of 374 faces not included in the tests. a) Mean face and parameters of the normalized model: H= height of the extracted faces; W= width; h_{eyes}= height of the eyes; and h_{mouth}= height of the mouth. b) Gaussian projection model of the vertical integral projection of the face, MV_{face}. c) Gaussian projection model of the horizontal projection of the eyes, MH_{eyes}.

4 Human Face Detection

Most face detectors –such as the popular method by Viola and Jones [10]– are based on multi-scale exhaustive search. Sung and Poggio were the first to develop this idea [12]. In essence, a binary (face/non-face) classifier is applied over all possible locations (or *windows*), and at all possible resolutions. These methods are known as *appearance-based* detectors.

4.1 Face Detection Using 1.5D Models

Our face detection technique follows the appearance-based approach, where the binary classifier is designed to work with integral projections and 1.5D models. The structure of the proposed detector is depicted in figure 6.

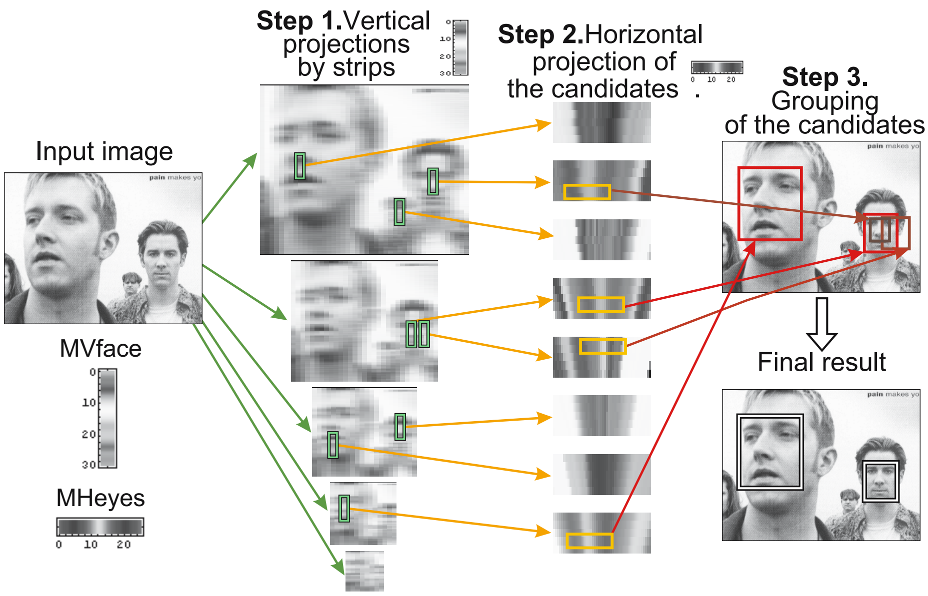

Fig. 6. Global structure of the face detection algorithm. In step 1, a pyramid of scaled vertical integral projections of the input image is computed, and the model MV_{face} is searched for at every position and scale. The resulting candidates are verified, in step 2, using horizontal projections and the corresponding model MH_{eyes}. Finally, tentative candidates are grouped in step 3 to eliminate multiple responses at the same location of the image.

The procedure can be summarized in the following 3 steps:

Step 1. First, the algorithm constructs a *pyramid* of vertical projections from the input image, using a typical scale factor reduction of 1.2. In accordance with the model, the width of the strips is W pixels, and these are computed in steps of $W/4$ pixels (thus, adjacent strips represent in fact partially overlapped regions). Then, the model of PV_{face} is searched for along the pyramid of projections.

This first step accounts for the uncertainty in the size of the faces, and represents the most expensive part of the process. However, using *integral images* [10], computing the whole pyramid of projections requires just $O(n)$ operations, where n is the number of pixels in the input image.

Step 2. The most relevant candidates obtained in step 1 are verified using PH_{eyes}. For each candidate, if the horizontal projection of the subregion approximately corresponding to the eyes –with slight variations in scale and horizontal position– does not fit with the model MH_{eyes}, the candidate is rejected; otherwise, it is accepted.

Step 3. Finally, the remaining candidates are analyzed in order to avoid multiple responses. Nearby and overlapping candidates are grouped together, and only the best candidate of each group is classified as a detected face.

4.2 Face Detection Results

Integral projections have already been applied to human face detection [3,4], mostly in *localization* scenarios –i.e., supposing that just one face is present in each image–. Figure 7 shows some sample results of the proposed detector on the public CMU/MIT face database [13], demonstrating that our method is able to detect an arbitrary number of faces in complex backgrounds.

Fig. 7. Sample results of the face detector based on integral projections. The images were taken from the CMU/MIT face database [13]. More results available at [14].

We have also evaluated the performance of our detector –both in quantitative and qualitative terms– on a face database developed by the authors' research group, which includes samples taken from TV, TDT, DVD, digital photographs, video-conference cameras, and additional samples taken from the CMU/MIT set –as shown in figure 7–. This database contains a total of 737 images with 853 faces. Our integral projection method (**IntProj**) was compared with two alternative –and publicly available– appearance-based techniques: Viola and Jones'

Table 1. Face detection results on a database with 737 images containing 853 faces. The percentage of correctly detected faces (*detection rate*) is shown for different false positive (FP) settings. FP ratio is relative to the number of images. The computer used was a Pentium IV at 2.6GHz, and the average image size is of 534 × 393 pixels.

Detection method	Detection rate				Time (ms)
	FP=5%	FP=10%	FP=20%	FP=50%	
IntProj	35.6	50.8	67.2	84.2	**85.2**
AdaBoost&Haar [10]	86.1	88.9	90.8	91.8	292.5
Neural Networks [13]	55.0	75.4	85.5	88.6	2337.7
AdaB&Haar + IntProj	**92.7**	**94.0**	**95.0**	**96.1**	295.6

detector [10], based on the AdaBoost algorithm and Haar-like filters, and Rowley's [13], based on neural networks. Additionally, a combination of our method and [10] was also studied. We present a summary of the results of this experiment in table 1, which shows the detection rates corresponding to several arbitrarily chosen false positive rates of the resulting ROC curves for each method.

Though more complex techniques certainly produce better detection rates, the proposed method exhibits a remarkable cost/benefit ratio. Moreover, **IntProj** and **NeuralNet** achieve similar maximum detection rates, while the former is 27 times faster. Considering only the images taken from webcams, **IntProj** reaches a 90% detection rate at 10% false positives per image.

But, clearly, the best performance is given by the combined method, which improves around a 5% the average detection rate of **AdaBoost&Haar**, at a negligible increment in the execution time.

5 Facial Feature Location

As shown in figure 7, our face detector simply outputs a rough location of the existing faces. This is a common characteristic of many methods available in the literature [15], where the faces are described with bounding rectangles. The purpose of facial feature location is to refine that description, providing –in our case– a precise location of the left and right eyes, and the mouth.

5.1 Face Feature Locator Using 1.5D Models

Basically, our facial feature locator performs a refined search of the 1.5D face model (i.e., MV_{face} and MH_{eyes}) on the previously detected faces. The input to this refinement process is the face rectangle generated by the detector. The proposed method consists of three steps (figure 8), all of them relying on the alignment algorithm described in section 2.3.

In the first step, we estimate face orientation, i.e. in-plane rotation[2]. This step makes use of face symmetry in a robust way. The vertical integral projection of

[2] The detector introduced in section 4 assumes that faces are approximately upright. In our experiments, an inclination of up to ±10° is allowed, with insignificant degradation in the detection rate.

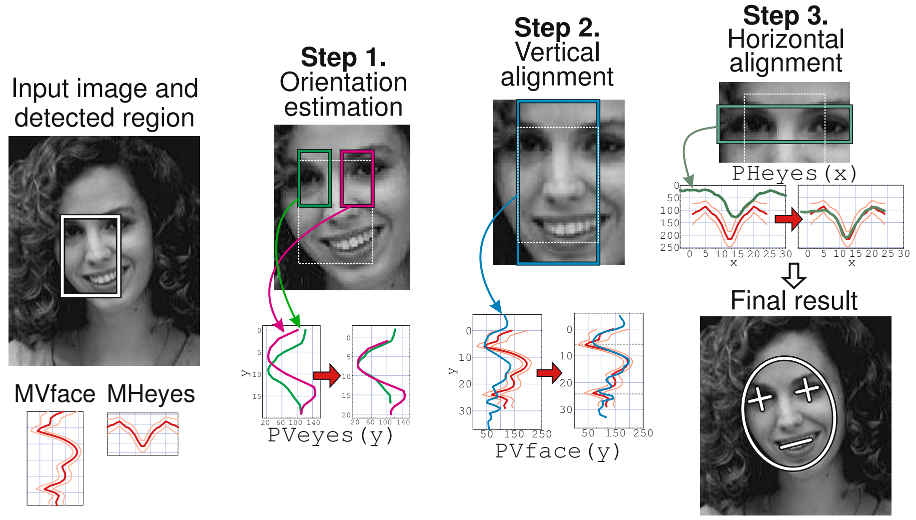

Fig. 8. Global structure of the facial feature location algorithm. In step 1, the orientation of the face is estimated using the vertical projections of both eyes. Then, in step 2, the vertical projection of the whole region is used to find the vertical position of the face. Similarly, the horizontal projection of eyes is computed and aligned in step 3.

the expected left and right eye regions are obtained. Then, both projections are aligned to each other, and the resulting displacement is easily transformed into an estimated inclination angle. This simple method can be very accurate up to angles of 20º.

The angle obtained in step 1 is used to rectify the input face. Then the accurate vertical and horizontal scale and location of the extracted face are determined in steps 2 and 3. The vertical integral projection of the face –along with an extra margin– is computed in PV_{face}, and this signal is aligned with respect to MV_{face}. Parameters $\{d, e\}$, resulting from the alignment algorithm (see equation 3) indicate the vertical translation and scale of the face, respectively.

In a similar way, in step 3 we compute the horizontal integral projection of the eyes' region PH_{eyes}, which is aligned with respect to MH_{eyes} to accurately locate the face horizontally. In this case, parameters $\{d, e\}$ indicate horizontal position and scale. Finally, the coordinates of the eyes and the mouth in the rectified image are mapped back into the original image.

5.2 Facial Feature Location Results

We present some sample results of the proposed method in figure 9. In all cases, the output of the combined face detector described in section 4 is used to feed the locator procedure based on integral projections, described in this section.

Though the proposed facial feature locator is based on face symmetry, it is notably robust even in situations where this symmetry is not so evident, as can

Fig. 9. Some sample results of the face locator based on integral projections. The four images on the left were taken from the CMU/MIT face database [13], and the rest from TV. More results available at [14].

be seen in figure 9. This is mainly due to two reasons: (1) changes in illumination are practically removed by the alignment process; and (2) integral projections remain invariant under small imprecisions in the projected regions –for example, if a part of the background is projected–. Furthermore, our method is able to work both with high and low resolution faces.

We have carried out extensive location experiments with more than 3700 manually labeled faces from the FERET database [9]. None of those faces were used to train the 1.5D face model[3]. The proposed method was compared with some alternative facial feature locators: a neural network-based eye locator by Rowley [13]; a simple template matching method (using correlation and mean eye and mouth patterns); and a modular eigenspace (eigen-eyes, eigen-mouth) technique [1]. The main results are summarized in table 2.

Again, the proposed method achieves a very good cost/benefit ratio. It is able to locate 99% of the faces with an error in the eyes position below 20% of the distance between eyes, taking just 3 ms of computing time per face. Moreover, in 96% of these cases the error is under 10%. Its accuracy is very similar or better than the neural networks locator, but it is about 100 times faster.

6 Face Tracking

Many different approaches have been proposed to deal with human face tracking, based on color, appearance, optical flow, predefined models, eigen-decompositions, and many other heterogeneous techniques. Not surprisingly, Ahlberg and

[3] In particular, the face model presented in figure 5 was used both in the detection and location experiments.

Table 2. Facial feature location results on 3743 faces from the FERET database [9]. For each method: **Location rate**: percentage of faces with both eyes with an error below 20%; **Angle diff.**: mean error in inclination angle (in degrees); **Dist. eyes**, **mouth**: mean distance error of eyes and mouth, respectively (error is always an Euclidean distance, expressed as % of the eye-to-eye distance); **Time**: average location time per face on a Pentium IV at 2.6GHz.

Location method	**Location rate (miss-locations)**	**Angle diff.**	**Dist. eyes**	**Dist. mouth**	**Time (ms)**
IntProj	**98.9% (41)**	**0.9°**	4.6%	**9.8%**	**3.1**
Neural Networks [13]	90.8% (343)	1.4°	**4.5%**	10.8%	346.0
Temp. Matching	91.1% (332)	2.0°	7.4%	10.5%	18.9
Eigen Features [1]	93.9% (272)	2.3°	6.2%	11.6%	45.1

Dornaika [16] use the expression "plethora of trackers" when talking about this topic. Here we prove that integral projections can also be applied successfully to this problem, producing a fast, stable and robust tracking method.

6.1 Face Tracking with 1.5D Models

Tracking methods are commonly based on two main components: a *prediction* mechanism, responsible for estimating tracking status in each new frame of a video sequence; and a *relocator*, which actually processes the current frame and computes the resulting position. If the observed motion is expected to be small, the first component can just be obviated; otherwise a more sophisticated predictor is required. In the human face domain, color based methods –such as the popular CamShift [17]– can be used to perform a suitable prediction. They are able to efficiently produce a robust but imprecise estimation of fast movements, that will be refined then by the relocator.

In this context, the problems of face relocation and facial feature location are closely related. Thus, our tracking algorithm shares a common structure with the technique described in section 5. However, there are two important differences:

1. The 1.5D face model –i.e., the projection models for PV_{face} and PH_{eyes}– is computed from the sequence itself. Recall that, in the case of facial feature location, a generic model was used.
2. The orientation estimation step is performed *after* vertical and horizontal alignments, instead of *before*. While in facial feature location the observed inclination can be relatively high, in tracking only a presumably slight variation of inclination needs to be considered.

Besides, an excessive alignment distance is used to detect the end of tracking. A more detailed description of the proposed face tracker can be found in [18].

6.2 Face Tracking Results

The method presented above was designed to perform robust and precise 2D face tracking under complex situations of facial expression, fast motion, low

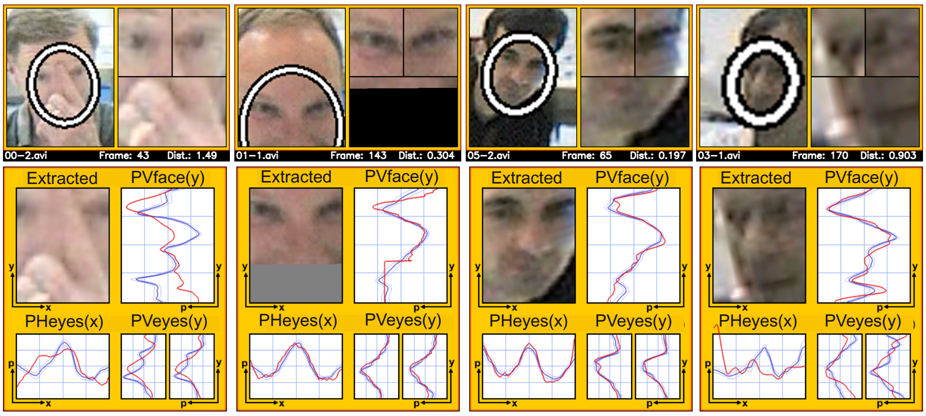

Fig. 10. Some sample results of the face tracker based on integral projections on videos from the NRC-IIT facial video database [19]. For each frame, we show the location of eyes and mouth (upper right), the bounding ellipse (upper left), and the computed projections (below).

Table 3. Face tracking results on 12 videos from the NRC-IIT facial video database [19] (00-1.avi, 00-2.avi, 01-1.avi, ..., 05-02.avi). The total number of faces (frame-by-frame) is 3635. **Tracked faces** and **false positives** are also counted frame-by-frame. Tracking is said to be correct if the obtained location is *close enough* to real eye positions (see caption of table 2); recall that all methods, except CamShift, involve a relocation of the eyes. The computer used was a Pentium IV at 2.6GHz.

Face Tracker	Tracked faces	False positives	Time (ms)
Detector [10]	2032 (55.8%)	368 (9.8%)	42.9
IntProj Null	2893 (79.6%)	493 (12.5%)	8.9
IntProj Color	**3050 (83.9%)**	**300 (7.3%)**	10.8
LK Tracker [20]	2828 (78.2%)	706 (17.1%)	**5.1**
Temp. Match	2387 (66.3%)	947 (23.9%)	11.3
CamShift [17]	1905 (51.5%)	1763 (47.0%)	5.8

resolution, partial occlusion, and poor illumination. The NRC-IIT facial video database [19], publicly available, is a good resource for experimentation under these circumstances. Figure 10 shows some difficult cases from this database.

Using 12 videos from the NRC-IIT set, we have compared the described tracker with three alternative approaches: a pyramidal implementation of Lucas and Kanade's method [20]; a template matching-based tracker; and the CamShift algorithm [17]. In addition, the result of applying Viola and Jones' face detector [10] to all frames is also reported. Table 3 summarizes the performance of these methods. The proposed technique (**IntProj**) was applied both without prediction (**Null**), and with a color based predictor (**Color**).

The low *detection rate* of **Detector** (below 56%) is a good indicator of the intrinsic complexity of the test. **IntProj Color** finds 50% more faces, while being

4 times faster. The high rate of false positives in **LK Tracker** is due to the well known *drift* problem of motion based trackers [16]. Our method attenuates its effects by using a robust model for the face. In **CamShift**, false positives are due to very imprecise locations of the face. In contrast, our tracker is able to provide an accurate location of eyes and mouth even in cases of low resolution.

We have carried out additional experiments using sample videos captured from TV, TDT, video-conference cameras and some DVD scenes. Several samples present great changes in out-of-plane rotation. In general, the range of allowed rotation is approximately $\pm 40^o$ in yaw, and $\pm 20^o$ in pitch. Many of these videos, and the obtained results, can be found at [14]. Basically, all the conclusions mentioned above still hold.

7 Discussion and Conclusions

In this paper we have tackled some of the main problems in face processing under a common framework based on integral projection models and alignment. Whilst projections are a classical and well-known technique in image analysis, little effort has been done to formalize their use. We have discussed the necessity to undertake this effort, by introducing the concept of a *probabilistic projection model* and a robust and general *alignment process*. Both aspects have been thoroughly studied, leading to a gaussian technique to model projections, and a fast iterative model-instance alignment algorithm. Using them in conjunction, we have proposed closely related solutions for several face processing problems, such as face detection on still images, facial feature location, and face tracking.

Our experiments prove that integral projections have a number of advantages with respect to other techniques: improved generalization, immunity to noise, and robustness against facial expressions and individual factors. The accuracy of the proposed algorithm is similar to that of the more complex state-of-the-art methods, with a considerable reduction of the computational cost.

Further applications of the proposed approach include the problems of person recognition, 3D pose estimation, and facial expression recognition [14]. Our future plans include using integral projections within the AdaBoost algorithm [10]. Instead of using Haar-like features, AdaBoost would take projections as the weak classifiers, giving rise to *not-so-weak* elementary classifiers.

Acknowledgements

This work has been supported by the Spanish MEC grant CSD2006-00046.

References

1. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: IEEE Computer Society Conf. on CVPR, pp. 84–91 (1994)
2. Kanade, T.: Picture Processing by Computer Complex and Recognition of Human Faces. PhD thesis, Kyoto University (1973)

3. Kotropoulos, C., Pitas, I.: Rule-based face detection in frontal views. In: Proc. I.C. Acoustics, Speech and Signal Processing, vol. 4, pp. 2537–2540 (1997)
4. Sobottka, K., Pitas, I.: Looking for faces and facial features in color images. PRIA: Advances in Mathematical Theory and Applications 7(1) (1997)
5. Feng, G.C., Yuen, P.C.: Variance projection function and its application to eye detection for human face recognition. Pattern Rec. Letters 19, 899–906 (1998)
6. Dean, S.R.: The Radon Transform and Some of Its Applications. John Wiley & Sons, New York (1983)
7. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Comm. ACM 15, 11–15 (1972)
8. Robinson, D., Milanfar, P.: Fast local and global projection-based methods for affine motion estimation. J. of Math. Imaging and Vision 18, 35–54 (2003)
9. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
10. Viola, P., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: IEEE Intl. Conf. on Comp. Vision and Pattern Recogn., pp. 12–14 (2001)
11. Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal 7, 308–313 (1964)
12. Sung, K.-K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. on PAMI 20(1), 39–51 (1998)
13. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 23–28 (1998)
14. García-Mateos, G.: Face processing with IP: `http://dis.um.es/~ginesgm/fip/`
15. Yang, M.-H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
16. Li, S.Z., Jain, A.K.: Handbook of Face Recognition. Springer, New York (2005)
17. Bradsky, G.D.: Computer vision face tracking as a component of a perceptual user interface. In: Workshop on Appl. of Comp. Vision, pp. 214–219. Princeton University Press, Princeton, NJ (1998)
18. García-Mateos, G.: Refining face tracking with integral projections. In: Luck, M., Mařík, V., Štěpánková, O., Trappl, R. (eds.) ACAI 2001 and EASSS 2001. LNCS (LNAI), vol. 2086, pp. 222–229. Springer, Heidelberg (2001)
19. Gorodnichy, D.O.: Video-based framework for face recognition in video. In: Second Workshop on FPiV 2005, Victoria, BC, Canada, pp. 330–338 (2005)
20. Bouguet, J.-Y.: Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Intel Corporation, Microprocessor Research Labs (2000)
21. Stegmann, M.B., Ersboll, B.K., Larsen, R.: FAME–a flexible appearance modeling environment. IEEE Transactions on Medical Imaging 22(10), 1319–1331 (2003)
22. Ma, Y., Ding, X.: Robust precise eye localization under probabilistic framework. In: Proc. of IEEE Conf. on Automatic Face and Gesture Recogn., pp. 339–344. IEEE Computer Society Press, Los Alamitos (2004)