

PROCESAMIENTO DE IMÁGENES

Programa de teoría

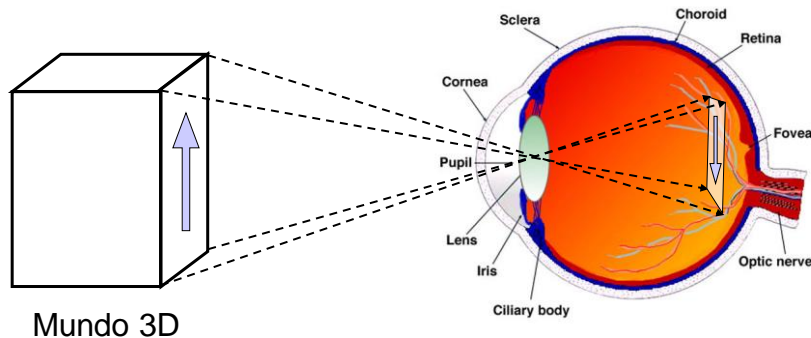
1. Adquisición y representación de imágenes.
2. Procesamiento global de imágenes.
3. Filtros y transformaciones locales.
4. Transformaciones geométricas.
5. Espacios de color y el dominio frecuencial.
6. Análisis de imágenes.

Tema 1. Adquisición y representación de imágenes.

- 1.1. El proceso de formación de imágenes.
- 1.2. Representación de imágenes digitales.
- 1.3. Dispositivos de captura.
- 1.4. Formatos de almacenamiento.
- A.1. Tipos de datos en OpenCV.

1.1. El proceso de formación de imágenes.

- La formación de imágenes es un proceso mediante el cual una información luminosa 3D (**la escena**) es **proyectada** en un plano 2D (**la imagen**).
- Las **cámaras** imitan el proceso que tiene lugar en el **ojo humano**.



<http://webvision.med.utah.edu/anatomy.html>

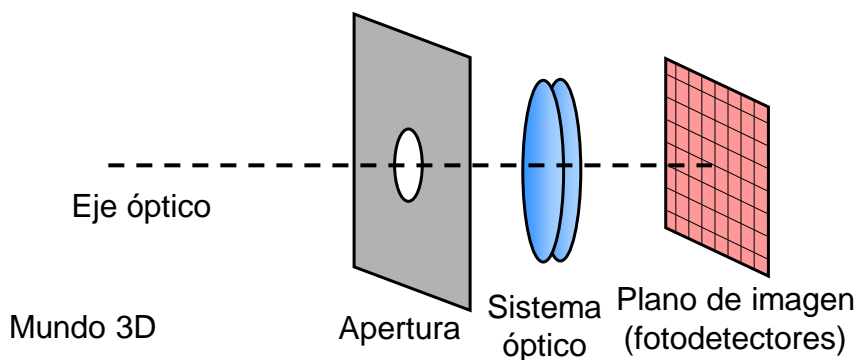
Mundo 3D

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

3

1.1. El proceso de formación de imágenes.

- **Modelo de cámara simplificado.**



Mundo 3D

Apertura

Sistema
óptico

Plano de imagen
(fotodetectores)

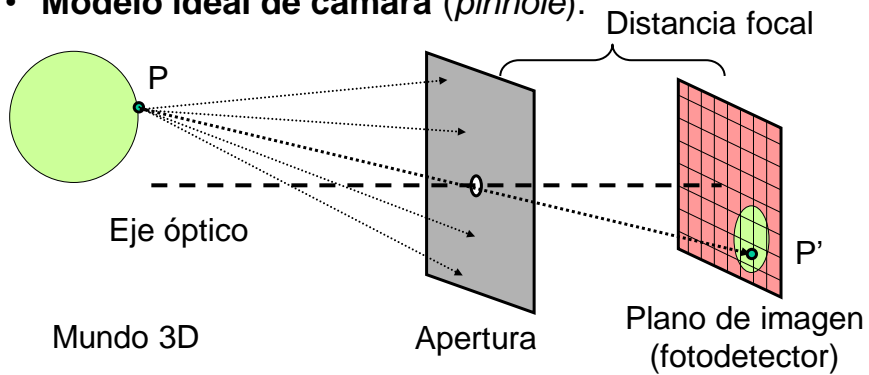
- El objetivo del modelo es que cada punto de la escena sea proyectado en un solo punto del plano de imagen.
- De esa manera la imagen estará **enfocada**.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

4

1.1. El proceso de formación de imágenes.

- Modelo ideal de cámara (*pinhole*).



- Una **superficie mate** emite luz en todas las direcciones.
- Cuando la **apertura es muy pequeña**, desde cualquier punto sólo pasa luz con una dirección.
- Todos los puntos están **bien definidos**: imagen enfocada.

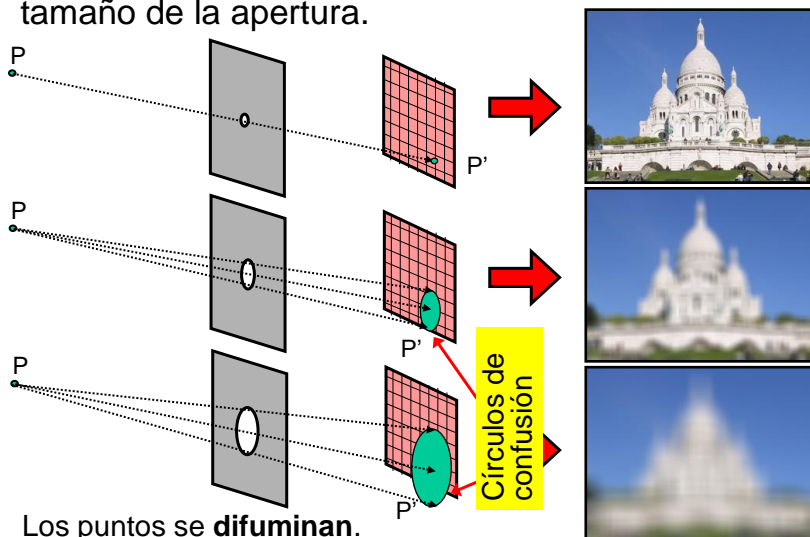
Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

5



1.1. El proceso de formación de imágenes.

- Comparar con lo que ocurre a medida que aumenta el tamaño de la apertura.



- Los puntos se **difuminan**.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

6

1.1. El proceso de formación de imágenes.

- La imagen estará enfocada cuando el tamaño del **círculo de confusión** sea menor que el tamaño de cada **celda del fotodetector**.



<http://www.pinholeaday.org/>

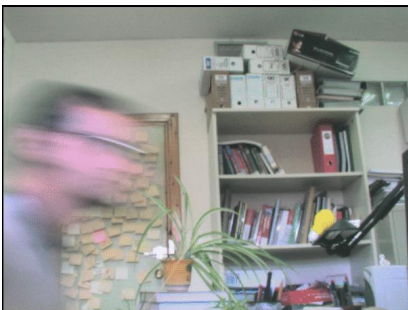
- Sin embargo, el modelo pinhole tiene muchas limitaciones y es poco usado.
- Apertura muy pequeña → Entra muy poca luz → La imagen sale muy oscura.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

7

1.1. El proceso de formación de imágenes.

- **Solución 1:** aumentar el **tiempo de exposición**, manteniendo el tamaño de la apertura.
 - No funciona bien si hay **movimiento**.
 - Aunque, se puede usar para *acumular* movimiento.



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

8

1.1. El proceso de formación de imágenes.

- **Solución 2:** aumentar el tamaño de la apertura y utilizar unas **lentes** que realicen el enfoque.
 - Por las limitaciones físicas de las lentes, sólo se pueden enfocar los objetos en cierta distancia.
 - **Profundidad de campo:** rango de distancias (en la escena) en la que los objetos aparecen enfocados.



Prof. campo (aprox.)
de unos pocos cm.



Prof. campo (aprox.)
desde 3 m. a infinito

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

9

1.1. El proceso de formación de imágenes.

- **Solución 3:** manteniendo reducidos el tiempo de exposición y el tamaño de la apertura, aumentar la **sensibilidad** de los fotodetectores.
 - Usando películas más sensibles o ajustando la sensibilidad (o ganancia) en la cámara.
 - **Problema:** con más sensibilidad aumenta el ruido.



Mucha luz, baja sensibilidad



Poca luz, alta sensibilidad

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

10



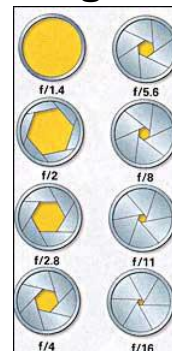
1.1. El proceso de formación de imágenes.

- Realmente, los **cuatro factores** no son contradictorios, sino que están presentes a la vez en todos los sistemas fotográficos:
 - **Tamaño de apertura** (o **apertura del diafragma**).
 - **Tiempo de exposición** (o **velocidad de obturación**).
 - **Sensibilidad de los fotodetectores** (o **valor ISO**).
 - **Óptica utilizada** (sistema de lentes). Dos funciones:
 - **Enfoque**: su ajuste, junto con la apertura, determina la **profundidad de campo**.
 - **Aumento (zoom)**: establece el **ángulo de visión** y la **distancia focal**.
- Los distintos elementos se ajustan para conseguir **dos objetivos**:
 - Que entre la cantidad de **luz suficiente**.
 - Que los objetos de interés estén **enfocados**.

1.1. El proceso de formación de imágenes.

Abertura del diafragma

- El **diafragma** es una imitación del **iris** de un ojo humano.
- La apertura se expresa en **relación a la constante f**. Típico entre **f/1,4** y **f/64**.
- Cuanto menor es la apertura, más **profundidad de campo** pero entra menos luz. El **pinhole** sería **f/GRANDE**.



<http://www.totonostra.com/fotografia/>

f/32



f/5



<http://en.wikipedia.org/wiki/F-number>

1.1. El proceso de formación de imágenes.

Velocidad de obturación

- Es el **tiempo** durante el cual se deja pasar la luz al fotodetector.
- Se mide en **segundos**. Normalmente entre 1/8000 y 30 segundos. Lo habitual en fotografía es $\sim 1/125$ s.
- Junto con la apertura, determina la **cantidad de luz que entra**.
- Otro problema es el **movimiento**. Si el tiempo es muy grande, la imagen puede aparecer movida.
 - Objetos que se mueven rápido en condiciones normales.
 - Movimiento involuntario en escenas nocturnas u oscuras.

1.1. El proceso de formación de imágenes.

Sensibilidad de los fotodetectores

- En fotografía **analógica**, está relacionada con la **composición y grosor de la película: estándares ISO**.
- Clasificados según el nivel de sensibilidad. Desde 3200 ISO (muy sensible) hasta 50 ISO (poco sensible).
- Cuanto más sensible, más ruido (efecto de **granularidad**).
- En fotografía **digital**, la sensibilidad está relacionada con la **ganancia** (voltios por fotón). Se hace un equivalente ISO.



ISO 80

ISO 400

ISO 800

ISO 1600

1.1. El proceso de formación de imágenes.

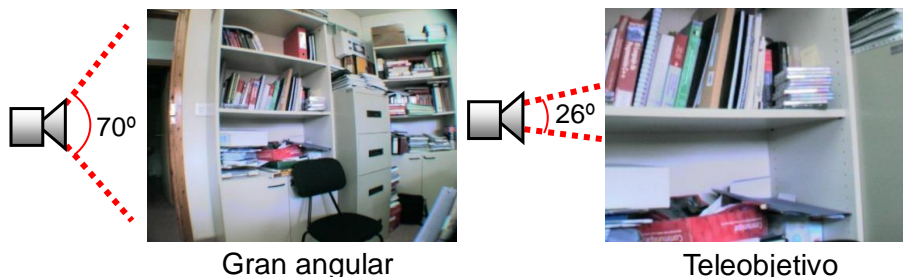
Óptica de enfoque

- Junto con el tamaño de apertura determina la **profundidad de campo**. Cuanto más amplia mejor.
- La p.c. es un **rango** definido por dos valores: la distancia más próxima enfocada y la más lejana.
- Suele estar entre unos pocos centímetros (**modo macro**) e infinito (paisajes, astronomía, etc.).
- **Tipos de sistemas** de enfoque:
 - **Enfoque fijo** (sin lentes de enfoque): cámaras pinhole, video-vigilancia,..., no muy habitual.
 - **Enfoque manual**: controlado por el usuario.
 - **Enfoque automático**: requiere un motor y una lógica de control. Normalmente basado en el punto central.

1.1. El proceso de formación de imágenes.

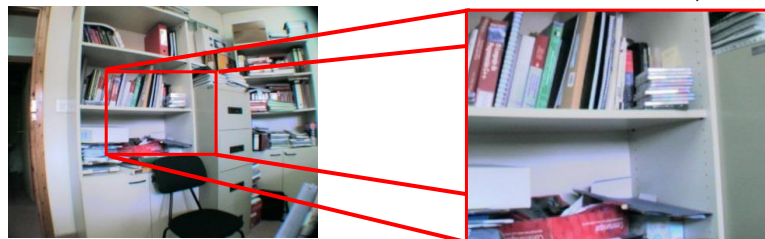
Óptica de aumento (zoom)

- Los conceptos de **aumento**, **zoom**, **campo visual** y **distancia focal** están estrechamente relacionados entre sí.
- **Campo visual**: cantidad (angular) de una escena que aparece visible en la imagen.
- Se distingue entre **gran angular** (ángulo grande, $>60^\circ$) y **teleobjetivo** (ángulo pequeño, $<30^\circ$).



1.1. El proceso de formación de imágenes.

- **Zoom (aumento):** relación (cociente) entre el máximo y el mínimo ángulo de visión.
- **Ejemplo.** Máximo 70° , mínimo 26° .
Zoom = $70/26 = 2,7x$



- **Tipos:** zoom analógico y digital.
- El **zoom analógico** se consigue modificando (desplazando) el sistema de lentes, haciendo que disminuya el campo visual.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

17

1.1. El proceso de formación de imágenes.

- El **zoom digital** es un simple proceso, a posteriori, de **interpolación**. El zoom digital no mejora la información ni la calidad de las imágenes. Más bien al revés.
- **Ejemplos.**



Con zoom digital

Con zoom óptico

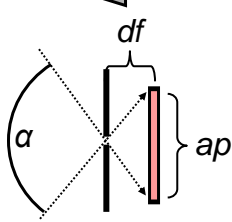
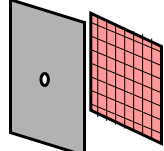
Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

18

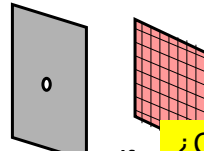
1.1. El proceso de formación de imágenes.

- El **ángulo de visión** y el **zoom** están estrechamente relacionados con la distancia focal.
- **Distancia focal:** distancia entre la apertura y el plano de imagen. Cuanto mayor distancia focal, menor ángulo de visión y más zoom, y viceversa.

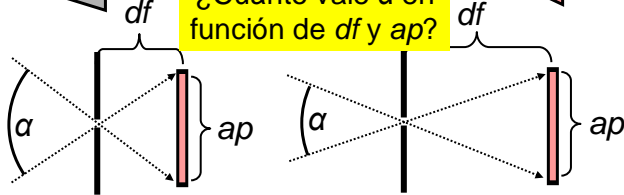
Gran angular



Teleobjetivo



¿Cuánto vale α en función de df y ap ?



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

19 +

1.1. El proceso de formación de imágenes.

$$\alpha = 2 \cdot \arctan (ap/2df)$$

- En analógico, la distancia focal se mide en **milímetros**. El plano de imagen suele ser película de 35 mm ($ap = 35$ mm).
- Ángulos de visión según la distancia focal:
 - **Gran angular** $df < 35$ mm $\rightarrow \alpha > 2 \arctan(35/70) = 53,1^\circ$
 - **Teleobjetivo** $df > 70$ mm $\rightarrow \alpha < 2 \arctan(35/140) = 28,1^\circ$
 - El equivalente del **ojo humano** es de unos 45° .
- En fotografía digital, se debería medir en **píxeles**... pero se mide también en mm. Por lo tanto, es necesario conocer el ancho del chip CCD.
- El zoom sirve para **acercar los objetos lejanos**, pero...
- ¿Qué ocurre si hacemos un zoom al mismo tiempo que alejamos la cámara? \rightarrow El efecto **zoom de Hitchcock**.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

20

1.1. El proceso de formación de imágenes.

- El efecto zoom de Hitchcock. Tiburón (1975).



<http://www.youtube.com/watch?v=Y48R6-iYHs>

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

21

1.1. El proceso de formación de imágenes.

- El efecto zoom de Hitchcock. ESDLA 1 (2001).

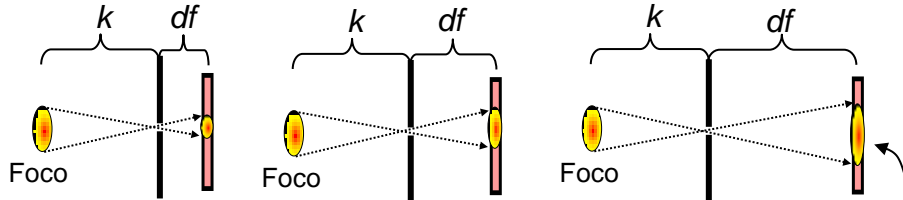


Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

22

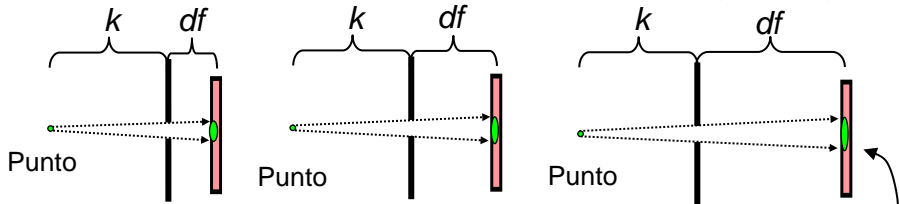
1.1. El proceso de formación de imágenes.

- **Ojo:** la distancia focal también influye en la cantidad de luz entrante y en la profundidad de campo.
- **Cantidad de luz:** a menor distancia focal, entra más luz.



Razón: la misma cantidad de luz se distribuye en un espacio mayor

- **Profundidad de campo:** mejor enfoque con un gran angular.

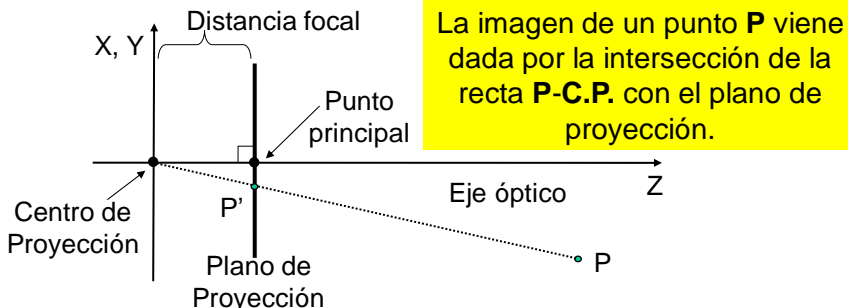


Razón: a igual distancia aumenta el tamaño del círculo de confusión

Tema 1. Adquisición y representación de imágenes.

1.1. El proceso de formación de imágenes.

- Matemáticamente, el proceso de formación de imágenes es modelado como una **proyección perspectiva**.
- **Elementos del modelo** de proyección perspectiva:
 - **Centro de proyección** (equivale a la apertura del *pinhole*).
 - **Plano de proyección** (plano de la imagen).



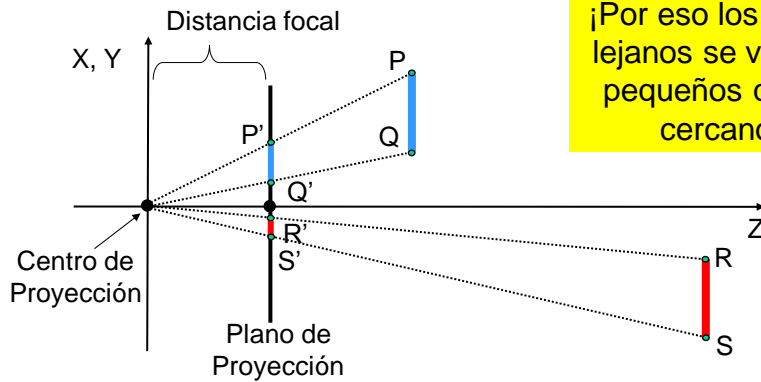
Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

24

1.1. El proceso de formación de imágenes.

- Si el centro de proyección es el punto $(0, 0, 0)$ y la distancia focal es 1, y el punto principal $(0, 0, 1)$, la proyección en la imagen de un punto $P = (x, y, z)$ será:

$$(u, v) = (x/z, y/z)$$



¡Por eso los objetos lejanos se ven más pequeños que los cercanos!

1.1. El proceso de formación de imágenes.

- Pero el modelo proyectivo no es completo. No explica algunos fenómenos como el **desenfoque**, la **distorsión radial** y las **aberraciones cromáticas**.

Conclusiones:

- El proceso de formación está en la “**parte analógica**” del ámbito de procesamiento de imágenes.
- Para nosotros las imágenes serán simples **matrices de números**, pero...
- Es importante conocer los **elementos, factores y parámetros** que intervienen en los dispositivos de captura.
- Por suerte (o por desgracia), muchas cámaras **no permiten ajustar** los parámetros, lo hacen automáticamente.

1.2. Representación de imágenes digitales.

- Una **imagen digital** es una matriz, o array bi-dimensional, de números.
- Cada celda de la matriz es un **píxel**.
- **Ejemplo.** Imagen de 20x15.

90	67	68	75	78	98	185	180	153	139	132	106	70	80	81	69	69	67	35	34
92	87	73	78	82	132	180	152	134	120	102	106	95	75	72	63	75	42	19	29
63	102	89	76	98	163	166	164	175	159	120	103	132	96	68	42	49	46	17	22
45	83	109	80	130	158	166	174	158	134	105	71	82	121	80	51	12	50	31	17
39	69	92	115	154	122	144	173	155	105	98	86	82	106	83	76	17	29	41	19
34	80	73	132	144	110	142	181	173	122	100	88	141	142	111	87	33	18	46	36
37	93	88	136	171	164	137	171	190	149	110	137	168	161	132	96	56	23	48	49
66	117	106	147	188	202	198	187	187	159	124	151	167	158	138	105	80	55	59	54
127	136	107	144	188	197	188	184	192	172	124	151	138	108	116	114	84	46	67	54
143	134	99	143	188	172	129	127	179	167	106	118	111	54	70	95	90	46	69	52
141	137	96	146	167	123	91	90	151	156	121	93	78	82	97	91	87	45	66	39
139	137	80	131	162	145	131	129	154	161	158	149	134	122	115	99	84	35	52	30
137	133	56	104	165	167	174	181	175	169	165	162	158	142	124	103	67	19	31	23
135	132	65	86	173	186	200	198	181	171	162	153	145	135	121	104	53	14	15	33
132	132	88	50	149	182	189	191	186	178	166	157	148	131	106	78	28	10	15	44

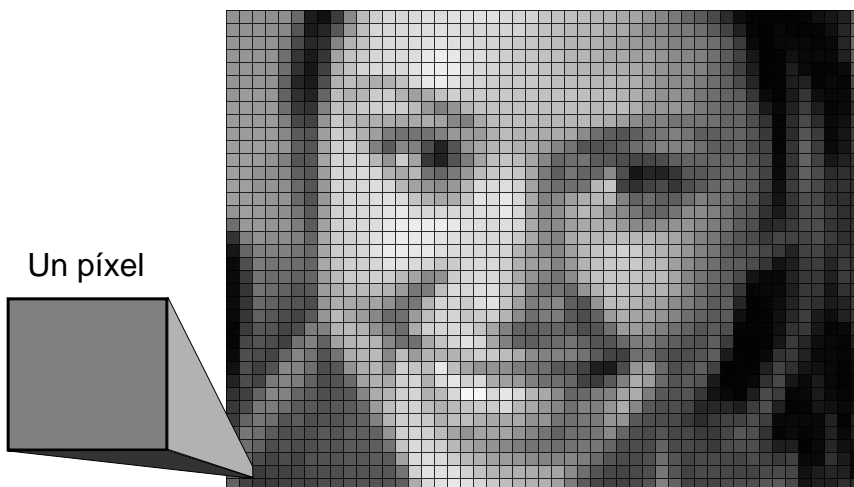
Un píxel

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

27

1.2. Representación de imágenes digitales.

- Una forma más común de visualizar una imagen...



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

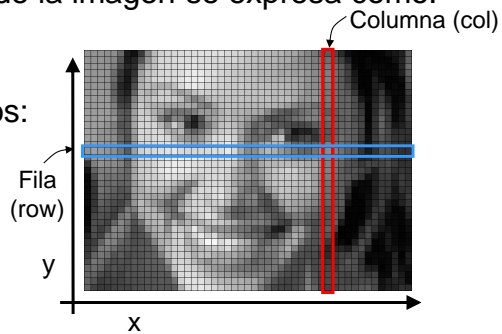
28

1.2. Representación de imágenes digitales.

Un poco de nomenclatura

- N° de columnas de la matriz: **ancho** de la imagen (*width*).
- N° de filas de la matriz: **alto** de la imagen (*height*).
- Eje horizontal: **eje x**.
- Eje vertical: **eje y**.
- Normalmente el tamaño de la imagen se expresa como: **ancho x alto**

- **Ejemplo.** Tamaños típicos:
320x240, 640x480,
800x600, 1024x768, ...

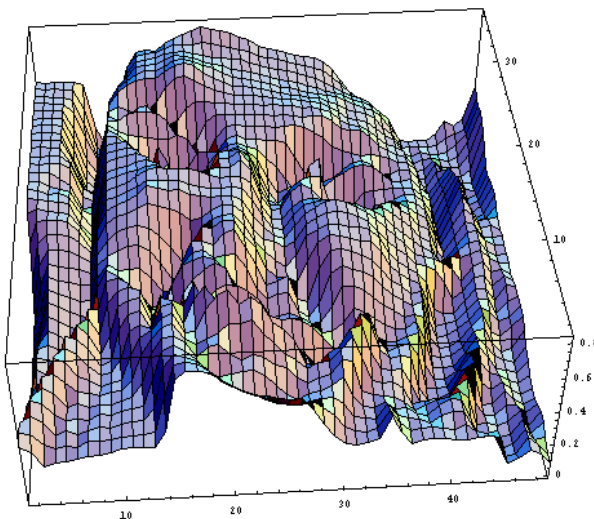


Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

29

1.2. Representación de imágenes digitales.

- Otra posible forma de visualizar la imagen...



Una imagen se puede interpretar como una superficie bidimensional.

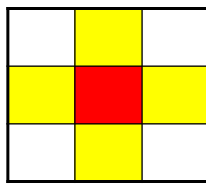
Una imagen digital es un muestreo discreto de la señal continua.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

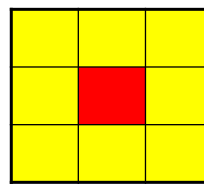
30

1.2. Representación de imágenes digitales.

- **Vecindad, proximidad:** en una imagen, los píxeles próximos tienen una “*relación más estrecha*” entre sí que los lejanos. → Diferencia respecto a una matriz en sentido genérico.
- Por ejemplo, se espera que los valores de dos píxeles próximos sean más o menos *parecidos*.
- Tiene sentido definir la **vecindad** de un píxel y la **distancia** entre dos píxeles.



Vecindad a 4



Vecindad a 8

1.2. Representación de imágenes digitales.

- Supondremos un acceso indexado a los píxeles: si i es una imagen, $i(x, y)$ será el valor del píxel en la columna x , fila y .
- Pero, ¿qué indica el valor de un píxel?
- Cada píxel representa el valor de una **magnitud física**.
 - Cantidad de **luz** en un punto de una escena.
 - Valor de **color** (cantidad de radiación en la frecuencia del rojo, verde y azul).
 - Nivel de **radiación** infrarroja, rayos X, etc. En general, cualquier radiación electromagnética.
 - **Profundidad** (distancia) de una escena en una dirección.
 - Cantidad de **presión** ejercida en un punto.
 - Nivel de **absorción** de determinada radiación.
 - Etcétera, etcétera.

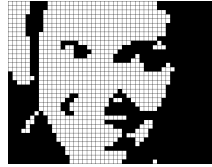
1.2. Representación de imágenes digitales.

¿De qué tipo de datos es cada celda de la matriz?

- **Imagen binaria:**

1 píxel = 1 bit

– 0 = negro; 1 = blanco

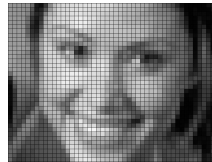


- **Imagen en escala de grises:**

1 píxel = 1 byte

– Permite 256 niveles de gris

– 0 = negro; 255 = blanco



- **Imagen en color:**

1 píxel = 3 bytes

– Cada píxel consta de 3 valores:
(Rojo, Verde, Azul)

– Un byte por color

– 16,7 millones de colores posibles



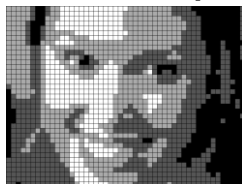
Procesamiento de Imágenes

Tema 1. Adquisición y representación de imágenes.

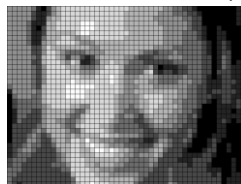
33

1.2. Representación de imágenes digitales.

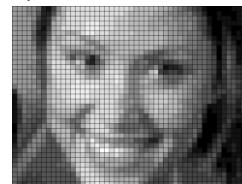
- Pero no limitado a... existen *infinitos* tipos posibles.
- Un nivel de gris, o un color, se puede representar con más o menos bits: **profundidad** de color (*depth*).



2 bits por píxel



3 bits por píxel



4 bits por píxel

- **Hi-color:** método reducido para representar colores
1 píxel = 2 bytes
– 5 bits por cada color (Rojo, Verde, Azul)
- **Imágenes en punto flotante:** útiles en procesos intermedios
1 píxel = 1 float o un double

Procesamiento de Imágenes

Tema 1. Adquisición y representación de imágenes.

34

1.2. Representación de imágenes digitales.

Imágenes multicanal:

- Cuando los píxeles representan magnitudes en distintos dominios físicos, decimos que la imagen es **multicanal**.
- **Ejemplo.** Imagen en color → Imagen con 3 canales: canal R (rojo), canal G (verde), canal B (azul).



Canal R



Canal G



Canal B

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

35

1.2. Representación de imágenes digitales.

Imágenes multicanal:

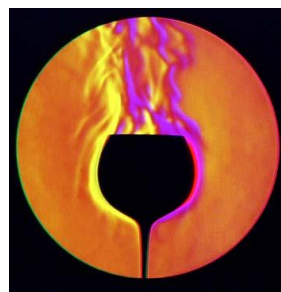
- **Ejemplo.** RGBA → Imagen RGB más canal Alfa. El canal Alfa representa el nivel de transparencia del píxel.
- **Ejemplo.** En algunas aplicaciones (p.ej., imágenes de satélite, visión nocturna) suelen usarse canales para frecuencias no visibles, infrarrojo, ultravioleta, etc.



as it looks to the eye

in ultraviolet (apurious-color)

proper B&W appearance in ultraviolet



<http://www.rit.edu/~andpp/h/>

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

36

1.2. Representación de imágenes digitales.

Almacenamiento de imágenes digitales

- **Cuestión 1:** ¿Cuál es el origen de coordenadas y el orden de las filas?
 - **Top-left:** el píxel $i(0, 0)$ es la esquina superior izquierda.
→ Suele ser el más habitual.
 - **Bottom-left:** el píxel $i(0, 0)$ es la esquina inferior izquierda. → Usado en algunos formatos (p.ej. BMP).
- **Cuestión 2:** ¿Cómo se almacenan los distintos canales?
 - **Entrelazado** (*interleaved, pixel order*): $R_0, G_0, B_0, R_1, G_1, B_1, R_2, G_2, B_2, \dots, R_n, G_n, B_n$ | $B_0, R_0, G_0, 0, B_1, R_1, G_1, 0, \dots$
 - **No entrelazado** (*non-interleaved, plane order*): $R_0, R_1, R_2, \dots, R_n, G_0, G_1, G_2, \dots, G_n, B_0, B_1, B_2, \dots, B_n$.

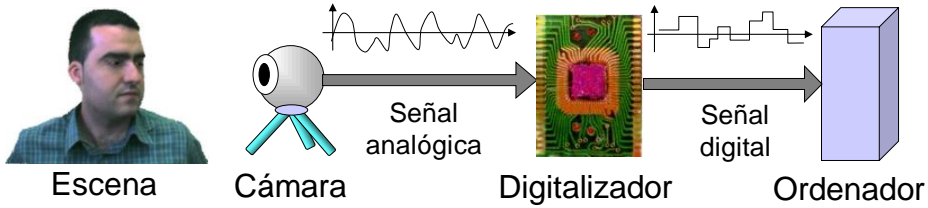
1.2. Representación de imágenes digitales.

Resumen

- **Parámetros de una imagen digital:**
 - Ancho y alto.
 - Número de canales y significado de cada uno.
 - Número de bits por píxel y canal (*depth*).
 - Origen de coordenadas y modo de almacenamiento multicanal.
- **Resolución espacial:** tamaño de la imagen.
- **Resolución fotométrica:** profundidad de color.
- **Resolución temporal:** aplicable en vídeos.

1.3. Dispositivos de captura.

- Podemos distinguir entre **captura** y **digitalización**.



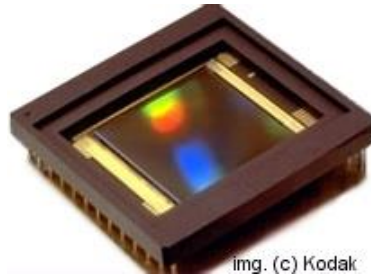
- **Señal analógica:** señal de vídeo, foto impresa, diapositiva, etc.
- **Digitalizadores:** digitalizador de vídeo, escáner, etc.
- Actualmente, la **distinción** es cada vez más **difusa**. Captura y digitalización van incorporadas en los mismos dispositivos (cámaras y escáneres).

1.3. Dispositivos de captura.

- **Características de un digitalizador:**
 - **Tamaño de imagen.** Ancho y alto de las imágenes tomadas. Depende del número de píxeles de fotodetector.
 - En cámaras fotográficas se mide en megapíxeles. Por ejemplo, resolución máxima: 2048x1536 → 3,34 megapíxeles.
 - En cámaras de vídeo suele ser mucho menor. Un tamaño típico puede ser de 800x600 → 0,48 megapíxeles.
 - **Tamaño del píxel.** Determina la densidad de píxeles. Es más relevante, por ejemplo, en escáneres.
 - **Propiedad física medida.** Luz, infrarrojo, ultravioleta, etc.
 - **Linealidad.** El nivel de gris debería ser proporcional al brillo de la imagen. Tb. es importante el número de niveles de gris.
 - **Nivel de ruido.** Ante una escena de color uniforme todos los píxeles deberían ser iguales. Pero nunca lo son. El ruido se mide en relación al nivel de contraste en la imagen.

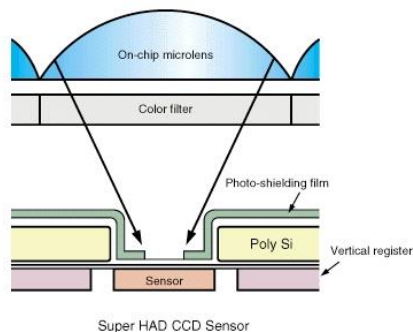
1.3. Dispositivos de captura.

- Existen muchos tipos de dispositivos de captura, según el tipo de iluminación, sensores y mecanismo de escaneado de la imagen.
- Los más populares son los basados en **CCD: Charge-Coupled Devices**.
 - Se han *impuesto* en **muchos ámbitos**: fotografía digital, vídeo digital, cámaras de TV, astronomía, microscopía, escáneres, etc.
 - Utilizan **sensores de silicio**.
 - El CCD es un chip que integra una **matriz de fotodetectores**.



1.3. Dispositivos de captura.

- Esquema de una **celda del CCD** (o píxel).

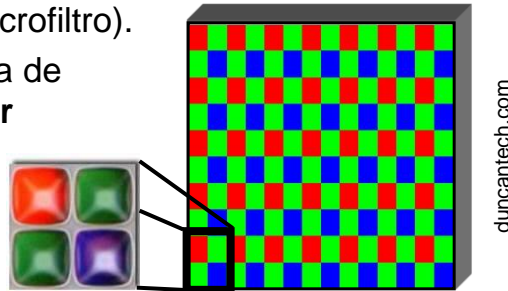


Sony Semiconductors

- El CCD está construido en un semiconductor de **silicio**.
- Cuando llega un **fotón**, el semiconductor libera **electrones**.
- Cada celda es un **pozo**, que **acumula** los electrones que han saltado (similar a un condensador).
- El n° de electrones es proporcional a la **intensidad de luz**.

1.3. Dispositivos de captura.

- El detector es independiente del color → Se usa un **filtro de color** (microfiltro).
- **Distribución** típica de los **filtros de color** en el CCD (**patrón de Bayer**).



duncantech.com

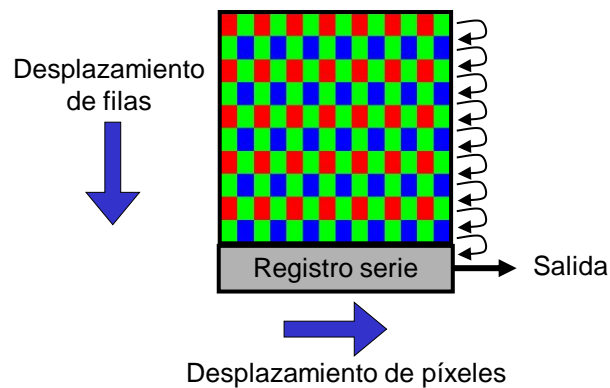
- **Ojo:** existen el **doble** de detectores **de verde** que de rojo y de azul. **Razón:** el ojo humano es mucho más sensible al verde que a los otros colores.
- Cada fotodetector es un píxel. Los colores no presentes se **interpolan** usando los 2 ó 4 píxeles vecinos de ese color.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

43

1.3. Dispositivos de captura.

- ¿Cómo se leen los valores de los píxeles?
- Hay un **desplazamiento de la carga** de los pozos, hasta salir por un extremo.



- Esto es el llamado "*full frame CCD*".

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

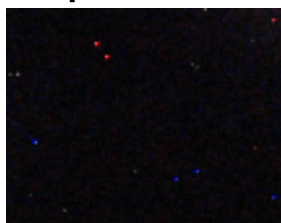
44

1.3. Dispositivos de captura.

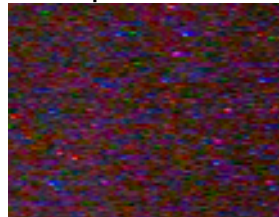
- Por sus buenas características, los CCD son muy usados en muchos ámbitos.
- Pero también tienen sus **limitaciones**:
 - **Corriente oscura (*dark current*)**: los electrones saltan al llegar un fotón, pero también pueden hacerlo por el calor.
 - **Campo de estrellas**: las imperfecciones provocan algunos píxeles con alta corriente oscura. Aunque no llegue luz, aparecen iluminados.
 - **Ruido fotónico**: debido a la naturaleza cuántica de la luz. Es mayor con escasa iluminación.
 - **Rebosamiento (*blooming*)**: cuando un pozo se llena de electrones, se *desparrama* su contenido a los píxeles cercanos.

1.3. Dispositivos de captura.

- **Campo de estrellas**. Mayor cuanto peor es la cámara.

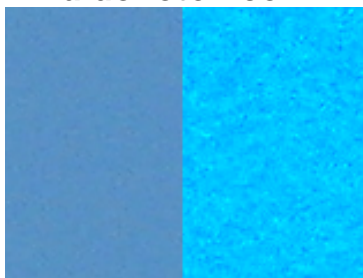


SONY DFW500



QUICKCAM PRO

- **Ruido fotónico**

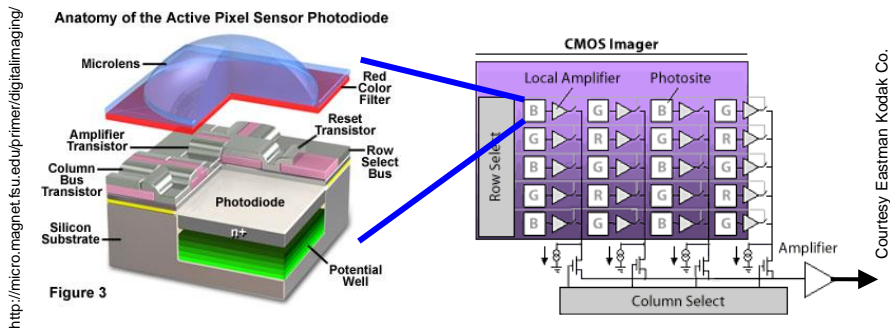


- **Rebosamiento (*blooming*)**



1.3. Dispositivos de captura.

- Otro tipo muy popular son los chips **CMOS: Complementary Metal-Oxide-Semiconductor**.
 - También basados en **semiconductores de silicio**.
 - **Diferencia** con CCD: cada píxel incorpora su propia circuitería, se pueden leer y seleccionar independientemente (sin necesidad de desplazamientos).



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

47

1.3. Dispositivos de captura.

- **Ventajas:** suelen ser más rápidos, tienen mejor integración (necesitan menos circuitería) y disminuyen el *blooming*.
- **Inconvenientes:** hay menos espacio de captura en el chip (menos luz), son menos uniformes (hay más ruido) y necesitan *búfferes*.

Todos los píxeles deberían leerse al mismo tiempo



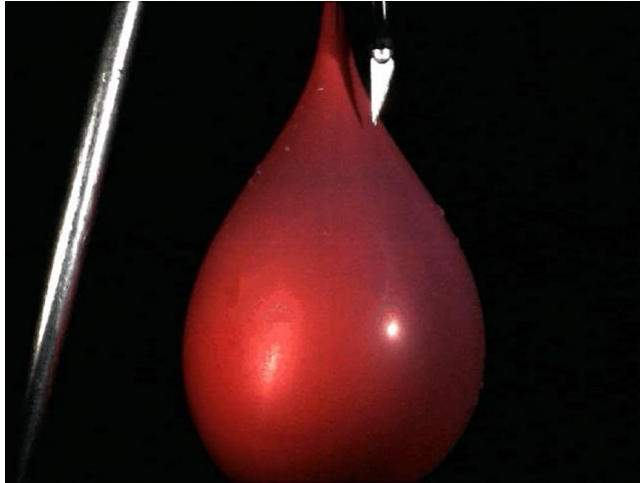
- Las cámaras fotográficas digitales suelen usar CCD.
- Las cámaras de videoconferencia suelen usar CMOS, aunque las de más calidad usan CCD.
- Las diferencias entre unas y otras son cada vez menores.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

48

1.3. Dispositivos de captura.

- **Ejemplo.** Los chips CMOS suelen usarse en aplicaciones que necesitan una **velocidad de muestreo muy alta.**



Capturado a 2.000 imágenes por segundo.

<http://www.visionresearch.com/index.cfm?sector=htm/app&page=Gallery>

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

49

1.3. Dispositivos de captura.

- También son útiles, por ejemplo, en **cámaras lineales.**
- **Ejemplo.** Aplicación en **foto-finish.**



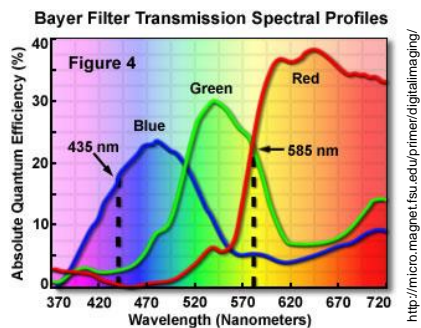
Capturado aprox. a 1.000 columnas por segundo.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

50

1.3. Dispositivos de captura.

- ¡OJO! la imágenes capturadas por una cámara no siempre corresponden a lo que ve el ojo humano.
- **Lo que el ojo no ve:** tanto los chips CCD como los CMOS son sensibles a la **radiación infrarroja...**
- ... como la emitida por un mando a distancia.



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

51 +

1.3. Dispositivos de captura.

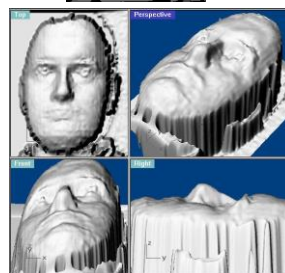
- Existen otros muchos tipos de dispositivos de captura, usados con imágenes de información no luminosa.
- **Ejemplos.**



Imagen de ultrasonidos (ecografía)



TAC (Tomografía axial computerizada)



Imágenes de profundidad

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

52

1.3. Dispositivos de captura.

Conclusiones

- La tecnología predominante son los dispositivos **basados en CCD y CMOS**.
- Ambos consisten en **arrays de fotodetectores** contruidos sobre un semiconductor de silicio.
- **Parámetros más relevantes:** número de píxeles del array y nivel de ruido.
- Otros parámetros (zoom, enfoque, distancia focal, etc.) dependen de la óptima de la cámara.
- En cada aplicación la mejor opción puede ser diferente.

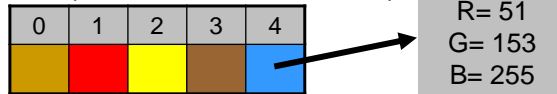
1.4. Formatos de almacenamiento.

- Existen muchos formatos. Podemos destacar: BMP, GIF, PNG, JPG, TIFF, etc.
- **Diferencias** entre los formatos:
 - Niveles de profundidad admitidos:
 - 1 bit → Imágenes en blanco y negro
 - 1 byte → Escala de grises o paleta de 256 colores
 - 3 bytes → Modelo RGB
 - Tipo de compresión:
 - Sin pérdida: RLE, LZW, Huffman
 - Con pérdida: mediante FFT, DCT, wavelets
 - Otras características:
 - Posibilidad de definir de transparencias
 - Diferentes imágenes en un mismo archivo (animaciones)
- Como resultado, **según la aplicación** será más adecuado uno u otro formato. ±

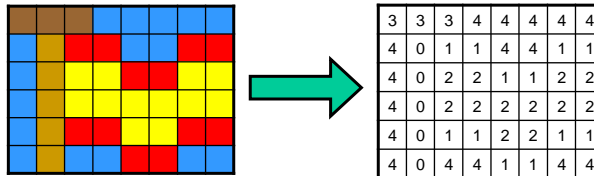
1.4. Formatos de almacenamiento.

Almacenamiento de imágenes mediante paleta

- **Paleta de colores:** es una tabla de tamaño n , donde cada posición es un color (normalmente en RGB).



- El valor de un píxel de la imagen hace referencia a la paleta.



- Número de bits/píxel \Leftrightarrow Tamaño de la paleta.
2 bits = 4 colores; 3 bits = 8 colores; 4 bits = 16 colores; ...

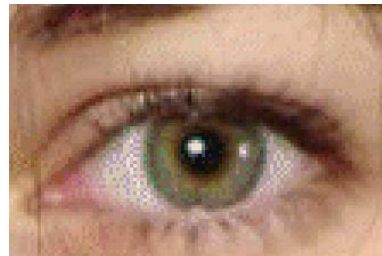
1.4. Formatos de almacenamiento.

Almacenamiento de imágenes mediante paleta

- Normalmente, las paletas no suelen ser de más de 256 colores (1 byte por píxel).
- Si la imagen originalmente tiene más colores, es necesario reducir los colores \rightarrow Seleccionar los **más usados**.
- **Resultado:** hay una pérdida de información de color.



Sin paleta



Con paleta (256 colores)

1.4. Formatos de almacenamiento.

- **Tipos de compresión**

- **Compresión sin pérdida:** si se comprime y luego se descomprime se obtiene la misma imagen.
- **Compresión con pérdida:** no se obtiene la misma imagen, hay una pérdida de calidad en la imagen.

- **Compresión RLE (*Run Length Encoding*):** sin pérdida.

Se basa en detectar la repetición de un mismo valor.

- Un valor no repetido se almacena directamente.
- Un valor repetido se almacena de forma especial, mediante un par (*Valor, N^o repeticiones*).
- **Ejemplo.** Icono bandera: 3, 3, 3, 4, 4, 4, 4, 4, 4, 0, 1, 1, ...
Imagen comprimida: (3, 3), (4, 6), 0, (1, 2), ...

1.4. Formatos de almacenamiento.

- **Compresión RLE (*Run Length Encoding*)**

- La compresión/descompresión es muy sencilla y rápida.
- Pero, ¿funcionará bien?



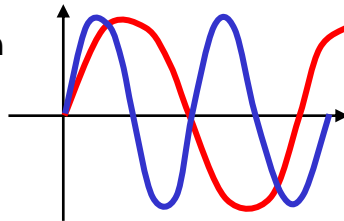
- En imágenes con muchas **regiones uniformes** la compresión será alta.
- Ocurrirá en dibujos “pintados a mano”.
- En **imágenes fotográficas**, con RGB, difícilmente se repetirá un valor.
- La compresión será escasa o nula.

1.4. Formatos de almacenamiento.

- **Compresión LZW (Lempel Ziv Welch):** sin pérdida.
 - Es un método sustitucional o basado en diccionario.
 - **Idea:** si una misma secuencia de valores se repite varias veces, hacer referencia al sitio donde se repite.
 - **Ejemplo.** Supongamos que queremos comprimir un texto.
Entrada: "Pablito clavó un clavito ¡Qué clavito clavó Pablito!"
Diccionario: #1 = Pablito; #2 = clavó; #3 = clavito
Comprimido: "#1 #2 un #3 ¡Qué #3 #2 #1!"
- ¿Funcionará bien?
 - Igual que el anterior, el funcionamiento óptimo será con dibujos (más que con fotos), y especialmente usando paletas de colores. Compresión en torno al 50%.

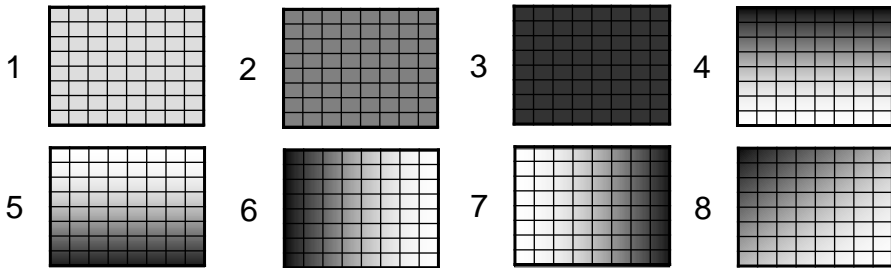
1.4. Formatos de almacenamiento.

- **Compresión con pérdida:** mediante FFT (Transformada Rápida de Fourier), DCT (Transformada Discreta del Coseno), wavelets, etc.
 - **Idea:** si se permite cierta pérdida en la calidad de las imágenes es posible alcanzar cotas más altas de compresión.
 - Cuanta más compresión, más pérdida de calidad.
 - La mayoría de las técnicas están basadas en **análisis frecuencial** de las imágenes.
 - **Recordatorio. Descomposición en series de Fourier:** cualquier señal continua se puede expresar como una suma de señales sinusoidales.



1.4. Formatos de almacenamiento.

- **Compresión mediante DCT en JPG:**
 - Idea parecida a la FFT, pero usando cosenos y en dos dimensiones.
 - **Explicación intuitiva:**
 - a) Las imágenes se dividen en bloques de 8x8 píxeles.
 - b) Existe un **catálogo estándar** de bloques de 8x8 píxeles (similar la paleta, pero con bloques y predefinida).



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

61

1.4. Formatos de almacenamiento.

- c) Cada bloque de 8x8 de la imagen es sustituido por el número del catálogo que sea más parecido.

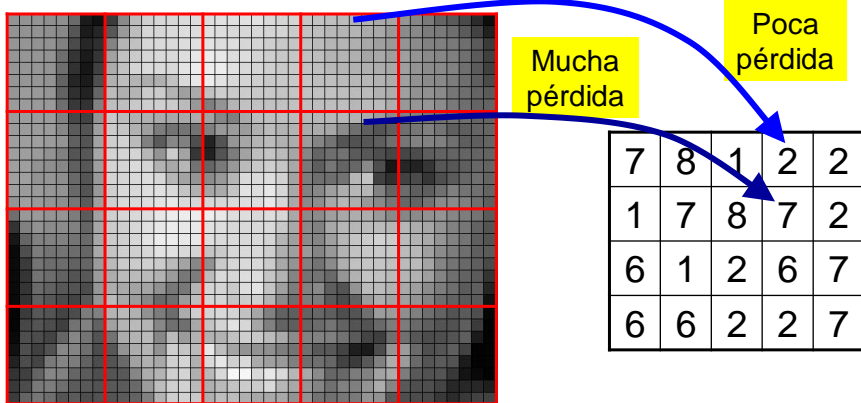


Imagen de entrada

Imagen comprimida

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

62

1.4. Formatos de almacenamiento.

- **Compresión mediante DCT en JPG:**
 - Es posible ajustar el nivel de compresión modificando el tamaño del catálogo de bloques.
 - **Catálogo pequeño** → Mucha compresión, pocos bits por cada gloque de 8x8, pero poca calidad.
 - **Catálogo grande** → Poca compresión, muchos bits por cada bloque, pero alta calidad.

Ojo, es una imagen pequeña



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

63

1.4. Formatos de almacenamiento.

Formato BMP (*Windows Bitmap*)

- Fue desarrollado por Microsoft para la permitir una rápida entrada/salida por disco/pantalla.
- **Características:**
 - Permite muchos niveles de profundidad: 1 bit por píxel (2 colores), 4 bits (16 colores), 8 bits (escala de grises o paleta), 16 bits (Hi-color) y 24 bits = 3 bytes (True-color).
 - Utiliza compresión sin pérdida: RLE o sin comprimir.
 - Almacenamiento bottom-left y entrelazado de canales.
- **Ventajas:**
 - No hay pérdida de calidad en las imágenes.
 - La lectura y escritura son muy rápidas.
 - Formato muy sencillo: cabecera + datos.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

64

1.4. Formatos de almacenamiento.

Formato BMP (*Windows Bitmap*)

- **Inconvenientes:**
 - El tamaño de las imágenes es excesivamente grande, sobre todo en imágenes fotográficas. Tamaño de imagen = (aprox.) ancho*alto*bits_por_pixel
 - No adecuado para transmisión por red.
 - Poco popular fuera de los entornos de MS Windows (aunque está libre de patentes).
- **Aplicaciones:**
 - Aplicaciones que requieran una rápida salida por pantalla.
 - Aplicaciones donde no deba haber pérdida de calidad, aun a costa del tamaño.

1.4. Formatos de almacenamiento.

Formato BMP (*Windows Bitmap*)

- **Ejemplos:**



Resolución: 512x384
Profundidad: 24 bits/píxel
Tamaño: 576 Kbytes



Resolución: 400x308
Profundidad: 24 bits/píxel
Tamaño: 167 Kbytes

1.4. Formatos de almacenamiento.

Formato GIF (*Graphics Interchange Format*)

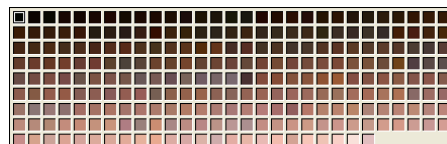
- Desarrollado por CompuServe en 1987 para la rápida transmisión de imágenes en color por las redes.
- **Características:**
 - Mucho más restringido que TIFF y que BMP.
 - Basado en uso de **paletas**, de hasta 256 colores.
 - Usa el algoritmo de **compresión LZW**.
 - **Ojo:** LZW es compresión sin pérdida, pero el uso de paletas implica una pérdida de información de color.
 - Permite definir **transparencias**. Se puede definir una entrada de la paleta como “transparente”.
 - Un fichero puede contener **múltiples imágenes**. Esto permite crear animaciones sencillas.

1.4. Formatos de almacenamiento. Formato GIF (*Graphics Interchange Format*)

Paleta: 32 colores
Tamaño: 33 Kbytes



Paleta: 256 colores
Tamaño: 87 Kbytes



1.4. Formatos de almacenamiento.

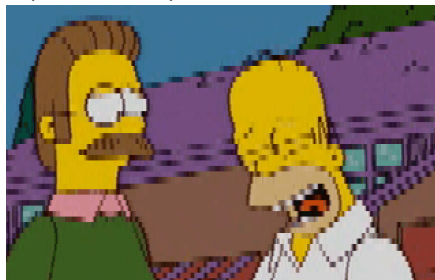
Formato GIF (*Graphics Interchange Format*)

- **Ventajas:**
 - Características no admitidas por otros formatos, como transparencias, animaciones y entrelazado.
 - Adecuado para transmisión en redes.
 - Muy popular.
- **Inconvenientes:**
 - Poco adecuado para imágenes fotográficas: pérdida de color y tamaños muy grandes.
 - Formato poco flexible.
 - Problemas de patentes hicieron que apareciera el formato PNG como alternativa al GIF. Hoy día, las patentes existentes sobre GIF han expirado.

1.4. Formatos de almacenamiento.

Formato GIF (*Graphics Interchange Format*)

- **Entrelazado:** las filas no se almacenan en posiciones consecutivas, sino salteadas (de 4 en 4).
- Esto permite hacerse una idea de la imagen cuando sólo se ha cargado una cuarta parte de la misma.
- **Aplicaciones:**
 - Compresión y almacenamiento de **dibujos e imágenes esquemáticas** con un número reducido de colores distintos.
 - Transmisión de imágenes por **red**: imágenes de tamaño reducido (iconos, símbolos, etc.), animaciones sencillas.



1.4. Formatos de almacenamiento.

Formato PNG (*PNG's NOT GIF*)

- Diseñado para reemplazar a GIF, está libre de patentes.
- Utiliza compresión **sin pérdida**, con el algoritmo **DEFLATE** (el mismo que gzip), basado en predicción: se espera que cada línea se parezca mucho a la anterior.
- **Profundidades** admitidas: 1, 2, 4, 8 bits/píxel (paleta o gris), 8, 16 bits (gris, RGB, o RGBA).
- Transparencias mediante **canal alfa**.
- **Desventajas:**
 - No adecuado para fotografías.
 - No permite animaciones.
- **Aplicaciones:** las mismas que GIF.



320x240 píxeles
(57,2 Kbytes)

http://en.wikipedia.org/wiki/Portable_Network_Graphics

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

71

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

- Es el formato más elaborado de los cuatro y orientado al almacenamiento de imágenes fotográficas.
- **Características:**
 - Admite imágenes en escala de grises (1 byte por píxel) y RGB (3 bytes por píxel).
 - Incluye un **mecanismo avanzado de compresión**, que puede ajustarse a distintos ratios de compresión.
 - La principal característica es la **compresión con pérdida**, mediante DCT.
 - El fichero puede incluir una versión reducida, para **previsualizar** la imagen antes de leerla entera.
 - Está libre de patentes.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

72

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

- **Mecanismo de compresión JPEG:**

- 1) Conversión del espacio de color, de RGB a YUV (Y= iluminación, UV= crominancia).
- 2) Reducción de resolución (a la mitad) en los canales UV.
→ El ojo humano es más sensible a la intensidad que al color.
- 3) Compresión mediante DCT de los grupos de 8x8 píxeles en cada canal.
→ El tamaño del “catálogo” depende del nivel de compresión.
- 4) Compresión sin pérdida del resultado mediante códigos de Huffman.

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

Tamaño: 31 Kbytes



Tamaño: 86 Kbytes



- Comprimiendo al mismo tamaño que GIF, la calidad es mejor, sobre todo en imágenes fotográficas. Pero...

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

- Con mayor compresión se producen distintos problemas.

Pérdida de color

Efecto de cuadrículado



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

75

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

- **Ventajas:**
 - En la mayoría de los casos, consigue un ratio compresión/calidad mucho mejor que los otros formatos.
 - Nivel de compresión ajustable. Típicamente entre 1:10 y 1:100
 - Formato muy popular y casi exclusivo en muchos ámbitos.
- **Inconvenientes:**
 - Compresión/descompresión complejas y costosas.
 - No incluye transparencias ni animaciones.
 - Genera artefactos o artificios (*artifacts*).
 - La información perdida no se recupera. Si trabajamos con un JPEG guardando en disco tras cada operación, la imagen se va degradando.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

76

1.4. Formatos de almacenamiento.

Formato JPEG (*Joint Photographic Experts Group*)

- **Aplicaciones:**
 - Prácticamente, todas las aplicaciones de fotografía digital: captura, almacenamiento, transmisión, impresión, etc.
 - No usar si no se permite pérdida de calidad o si se trabaja con dibujos.
- Los artefactos pueden ser inadmisibles en ciertas aplicaciones que requieren alta calidad.
- Existe un nuevo estándar, **JPEG2000** que evita los artefactos. En lugar de DCT, usa una transformación basada en **wavelets**.
 - Mejora la compresión sobre un 20%, pero es más costoso.

1.4. Formatos de almacenamiento. Comparación JPEG / JPEG 2000

JPG 3.12 Kbytes

JPG 1.27 Kbytes

PNG 135 Kbytes



JP2 3.08 Kbytes

JP2 1.13 Kbytes

JP2 832 bytes



1.4. Formatos de almacenamiento.

- **Otros: Formato TIFF (*Tagged Image File Format*)**
 - Diseñado para trabajos de impresión profesional de alta resolución y calidad (impresión industrial).
 - Es muy flexible, basado en **tags** (bloques de datos de formato predefinido).
 - El formato es **muy abierto**: admite hasta 64.000 canales, n^o arbitrario de bits por píxel (hasta enteros o reales de 64 bits), distintos espacios de color, múltiples imágenes por fichero, cualquier tipo de compresión existente, etc.
- **Otros: Formato RAW (*o negativo digital*)**
 - No existe un único estándar RAW, cada empresa usa el suyo.
 - Algunas características comunes: se almacenan los datos sin procesar; la profundidad suele ser 12 ó 14 bits/píxel; no son RGB, sino los resultados del patrón de Bayer; normalmente no hay compresión o es sin pérdida.

1.4. Formatos de almacenamiento.

Conclusiones:

- Buscar el **formato y nivel de compresión** más adecuado para cada aplicación particular.
- Cuidado con los **formatos con pérdida**. → Cada vez que se guarda hay una pérdida de calidad. → Guardar una copia sin pérdida del original.
- **Ojo**: son formatos de **almacenamiento**. Para procesar las imágenes en memoria no se usan estos formatos, sino imágenes descomprimidas (*en crudo*): matrices de píxeles.

1. Adquisición y representación de imágenes.

Conclusiones:

- Una **imagen digital** no es más que una matriz de números.
- Las imágenes digitales son **muestreos discretos** de señales continuas bidimensionales.
 - Discretización en el **espacio**: ancho y alto.
 - Discretización en el **valor**: profundidad de píxel.
- El **procesamiento de imágenes** recibe imágenes como entrada y produce imágenes en la salida (mejora, restauración, etc.).
- Pero... de donde no hay no se puede sacar. → La **adquisición** de imágenes sigue siendo fundamental.

Anexo A.1. Tipos de datos en OpenCV.

- Tipos de datos auxiliares
- El tipo Mat
- Almacenamiento de las imágenes
- Operaciones de entrada/salida
- Otras cuestiones
- Ejemplos

A.1. Tipos de datos en OpenCV.

Tipos de datos auxiliares:

- **Point**: coordenadas de un punto (un píxel) en una imagen. La numeración de filas y columnas empieza en 0. En esencia es una clase con dos enteros: **x**, **y**.

```
int x0= 100, y0= 200;
Point punto1(x0, y0);
Point punto2;
punto2= Point(x0, y0);
punto1.x= punto2.y+1;
...
```

- Existen otras variantes. Puntos con coordenadas reales: **Point2f** (float), **Point2d** (double); y puntos 3D con (x,y,z): **Point3i** (int), **Point3f** (float), **Point3d** (double).

A.1. Tipos de datos en OpenCV.

Tipos de datos auxiliares:

- **Size**: sirve para representar el tamaño de una región rectangular, en píxeles. Es una clase con dos atributos: width (anchura) y height (altura):

```
Size tam1(640, 480);
Size tam2;
tam2= Size(ancho, alto);
tam1.width= 200;
qDebug("%d", tam2.height);
...
```

- La variante **Size2f** se usa cuando el tamaño puede ser un valor real (no entero).

A.1. Tipos de datos en OpenCV.

- **Rect**: rectángulo en la imagen, dado por el píxel superior izquierdo (x, y), anchura (width) y altura (height).

```
Rect rec(0, 0, 640, 480);  
Rect r2;  
r2= Rect(x, y, ancho, alto);  
rec.widht= 200;  
qDebug("%d", r2.height);  
...
```

- **Scalar**: representa un color, consta de un array con 4 números double, que se pueden acceder con [].

```
Scalar color;  
color[0] -> Azul   color[1] -> Verde   color[2] -> Rojo
```

A.1. Tipos de datos en OpenCV.

- Estos tipos básicos aparecerán en los parámetros de las llamadas a muchas de las funciones.
- Existen **funciones *inline*** para crear e inicializar variables de estos tipos de datos: **Point**, **Size**, **Rect**, **Scalar**, **Vec3b**, etc.

```
Point pt1= Point(3,4), pt2;  
pt2.x= 5; pt2.y= 8;  
qDebug("%d", Size(100, 200).width);  
... Rect(2, 2, 50, 50) ...  
... Scalar::all(3) ... Scalar(5, 7, 9) ...  
line(imagen, pt1, pt2, color);  
line(imagen, Point(1,1), Point(10,10), color);
```

A.1. Tipos de datos en OpenCV.

- En las funciones para dibujar (puntos, líneas, círculos, elipses, etc.) el color se representa mediante un **Scalar**, que almacena los canales RGB.
- Ejemplo, función para dibujar un círculo:
void circle(... Point centro, int radio, **Scalar color**, ...)
- Para crearlo se puede usar la macro **CV_RGB(r, g, b)**:

```
#define CV_RGB(r, g, b)      Scalar((b), (g), (r))  
...  
circle(img, Point(100,100), 50, CV_RGB(0,0,255));  
...
```

↓
Color azul

A.1. El tipo Mat.

El tipo de datos para representar imágenes es el tipo **Mat**.

- Las imágenes en memoria están **descomprimidas**, como una matriz de píxeles que podemos consultar y modificar.
- Supongamos definida la variable: **Mat img**;
- Principales **propiedades de una imagen**:
 - **Tamaño**: img.size(); **Anchura**: img.size().width == img.cols;
Altura: img.size().height == img.rows
 - **Profundidad de píxeles**: img.depth() = enteros de 8, 16 y 32 bits, reales de 32 y 64 bits.
 - **Número de canales**. img.channels() = 1, 2, 3 ó 4.
 - El **origen** de coordenadas es **top-left** y el **orden** de los canales es **entrelazado de píxeles** con orden (B,G,R).

A.1. El tipo Mat.

- La memoria de Mat se reserva y se libera automáticamente.
- **Creación de una imagen:**

Mat imagen(int alto, int ancho, int tipo);

- **alto, ancho:** indican el tamaño de la imagen
- **tipo:** indica la profundidad de píxel y el número de canales:
 - CV_8UC1, CV_8UC2, CV_8UC3 - unsigned 8-bit integers
 - CV_16SC1, CV_16SC2, CV_16SC3 - signed 16-bit integers
 - CV_16UC1, CV_16UC2, CV_16UC3 - unsigned 16-bit integers
 - CV_32SC1, CV_32SC2, CV_32SC3 - signed 32-bit integers
 - CV_32FC1, CV_32FC2, CV_32FC3 - floating-point 32-bit
 - CV_64FC1, CV_64FC2, CV_64FC3 - floating-point 64-bit

- Otras formas alternativas de crear/redimensionar la imagen:

Mat imagen(Size size, int tipo [, Scalar color]);

imagen.create(Size size, int tipo [, Scalar color]);

imagen.create(int ancho, int alto, int tipo [, Scalar color]);

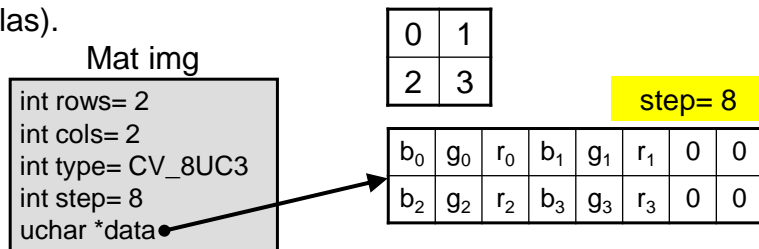
Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

89

A.1. Almacenamiento de imágenes.

Almacenamiento en memoria de los datos

- Los píxeles (en crudo) de la imagen están referenciados en: uchar ***data**;
- Los píxeles se almacenan por filas, de izquierda a derecha, empezando por la fila superior.
- Entre una fila y la siguiente hay **step** bytes (al final de la fila pueden sobrar algunos bytes, con el propósito de alinear las filas).



Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

90

A.1. Almacenamiento de imágenes.

Acceso a un píxel de una imagen

- El acceso a píxeles concretos es posible, aunque **es preferible usar funciones ya implementadas** sobre imágenes (que nos evitan tener que acceder a los píxeles).
- **Acceso al píxel (y, y) de una imagen: Mat img:**
 - **Leer un píxel:** TIPO valor= img.at<TIPO>(y, x);
 - **Escribir un píxel:** img.at<TIPO>(y, x)= valor;
 - El TIPO del píxel depende del tipo de la imagen:

img	TIPO	img	TIPO	img	TIPO	img	TIPO	img	TIPO	img	TIPO
8UC1	uchar	16SC1	short	16UC1	ushort	32SC1	int	32FC1	float	64FC1	double
8UC2	Vec2b	16SC2	Vec2s	16UC2	Vec2w	32SC2	Vec2i	32FC2	Vec2f	64FC2	Vec2d
8UC3	Vec3b	16SC3	Vec3s	16UC3	Vec3w	32SC3	Vec3i	32FC3	Vec3f	64FC3	Vec3d
8UC4	Vec4b	16SC4	Vec4s	16UC4	Vec4w	32SC4	Vec4i	32FC4	Vec4f	64FC4	Vec4d

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

91

A.1. Operaciones de entrada/salida.

Entrada/salida de imágenes a disco

- Las funciones para la **entrada/salida** de imágenes se encuentran en el módulo **HighGUI**, de OpenCV.
- **Leer una imagen** de un fichero:

Mat imagen= **imread** (string nombre, int enColor=1);

- **nombre:** nombre del fichero a leer.
- **enColor:** número de canales en la imagen resultante:
 - >0 : la imagen resultante tendrá siempre 3 canales (RGB)
 - =0 : la imagen tendrá 1 canal (escala de grises)
 - <0 : el número de canales depende del propio fichero
- **Formatos** admitidos: BMP, DIB, JPEG, PNG, PBM, PGM, PPM, RAS, TIFF y GIF.
- Si no se puede leer el fichero, **imagen.empty()** es true.

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

92

A.1. Operaciones de entrada/salida. Entrada/salida de imágenes a disco

- Guardar una imagen en un fichero:

bool **imwrite** (string nombre, Mat imagen);

- **nombre**: nombre del fichero a escribir.
- **imagen**: imagen a guardar (debe ser 8U).
- **Formatos** admitidos: BMP, DIB, JPEG, PNG, PBM, PGM, PPM, RAS, TIFF y GIF.
- Devuelve false si no se puede escribir.

- Guardar una imagen JPEG seleccionando el nivel de calidad/compresión:

vector<int> params;

valor= 0, máxima compresión
valor= 100, máxima calidad

params.push_back(CV_IMWRITE_JPEG_QUALITY);

params.push_back(valor); // *valor indica la calidad/compresión*

imwrite("salida.jpg", img, params);

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

93

A.1. Operaciones de entrada/salida. Salida de imágenes por pantalla

- HighGUI ofrece también una forma sencilla de visualizar las imágenes. Permite crear ventanas, que son referenciadas mediante cadenas de texto (string).

- **Crear una ventana:**

void **namedWindow** (string nombre, int flags);

- **nombre**: nombre de la ventana a crear.
- **flags**: admite distintas opciones: CV_WINDOW_NORMAL o CV_WINDOW_AUTOSIZE (redimensionar o no), CV_WINDOW_FREERATIO o CV_WINDOW_KEEPRATIO (relación de aspecto variable o fija), CV_GUI_NORMAL o CV_GUI_EXPANDED (ventana normal o expandida).

- Ejemplo típico:

namedWindow("Imagen", CV_WINDOW_NORMAL);

Procesamiento de Imágenes
Tema 1. Adquisición y representación de imágenes.

94

A.1. Operaciones de entrada/salida.

Salida de imágenes por pantalla

- **Mostrar una imagen en una ventana:**
void **imshow** (string nombre, Mat imagen);
 - **nombre**: nombre de la ventana a usar.
 - **imagen**: imagen a mostrar.
- **Destruir una ventana / destruirlas todas:**
void **destroyWindow** (string nombre);
void **destroyAllWindows** ();
 - **nombre**: nombre de la ventana a destruir.
- **Cambiar el tamaño de una ventana:**
void **resizeWindow** (string nombre, int ancho, int alto);
 - **nombre**: nombre de la ventana a redimensionar.

A.1. Operaciones de entrada/salida.

- **Esperar pulsación de una tecla:**
int **waitKey** (int delay= 0);
 - **delay**: tiempo máximo de espera, en milisegundos. Si es menor o igual que 0, espera indefinidamente.
 - Devuelve el código de la tecla, o -1 si no se pulsó ninguna.
- **Clonar una imagen:**
Mat clon= **imagen.clone()**;
 - **imagen**: imagen a clonar, se devuelve una copia exacta.
- **Otras operaciones** (consultar en la documentación):
 - **imagen= escalar**: inicializar una imagen con valor constante.
 - **split, merge**: separar los canales de una imagen, y componer varios canales en una imagen, respectivamente.
 - **imagen.convertTo**: convertir una imagen, cambiando los valores de profundidad de píxel y escalando los valores.

A.1. Otras cuestiones.

Algunas cuestiones adicionales

- En las operaciones que devuelven una imagen, la propia operación se encarga de crearla con el tamaño y tipo.
- **Modo in-place:** si una función lo admite, significa que la imagen resultante puede ser la misma que la de entrada.
- **Ejemplo: add** (imagenA, imagenB, imagenA);
- **Regiones de interés (ROI):** podemos definir una imagen como un fragmento de otra (una ROI). Las operaciones se aplican solo sobre la región de interés.
- **Creación de una región de interés:**
Mat fragmento= imagen(Rect(x, y, ancho, alto));
- Algunas funciones permiten usar una máscara (**mask**, imagen booleana; 0: no aplicar la operación; ≠0: sí aplicar).

A.1. Ejemplos.

- **Ejemplo 1.** Leer una imagen y mostrarla en una ventana. Se espera que el usuario pulse una tecla para cerrarla.

```
Mat img;  
namedWindow("Imagen", CV_WINDOW_NORMAL);  
img= imread("fichero.jpg", 1); // ¿Y si ponemos un 0?  
imshow("Imagen", img);  
waitKey(0);  
destroyWindow("Imagen");
```

A.1. Ejemplos.

- **Ejemplo 2.** Leer una imagen en color, separar los canales R, G y B, y guardarlos en 3 ficheros distintos.

```
string nombres[3]= {"canalB.jpg", "canalG.jpg", "canalR.jpg"};
Mat img= imread("imagen.jpg", 1);
namedWindow("Imagen", 0);
imshow("Imagen", img);
vector<Mat> canales;
split(img, canales);
for (int i= 0; i<3; i++) {
    namedWindow(nombres[i], 0);
    imshow(nombres[i], canales[i]);
    imwrite(nombres[i], canales[i]);
}
```