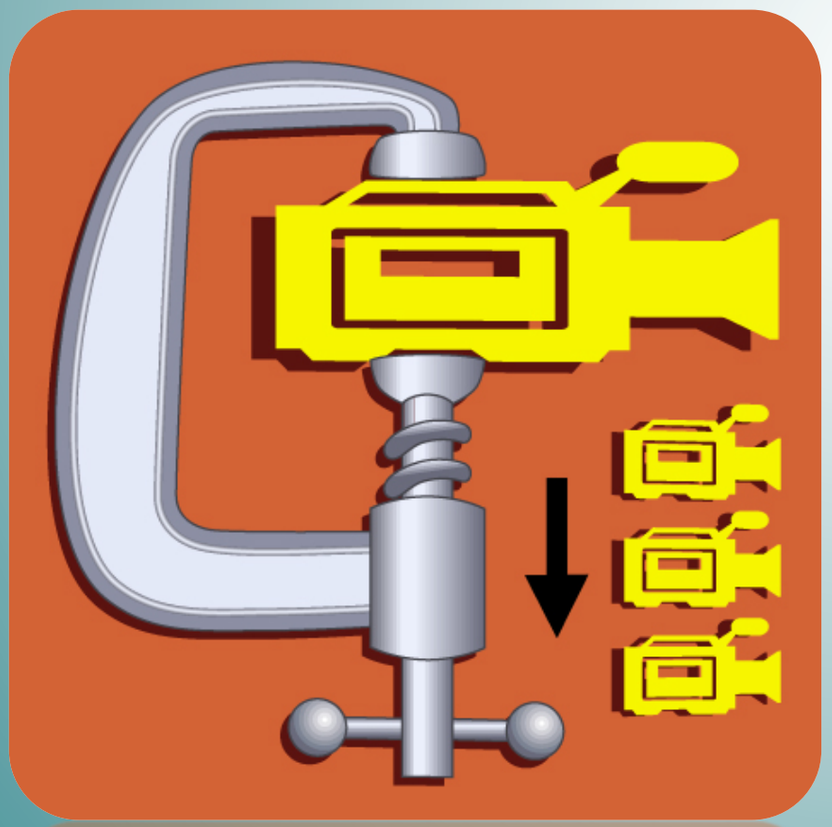
Compresión de Vídeo

Tema 1.1. Imagen digital



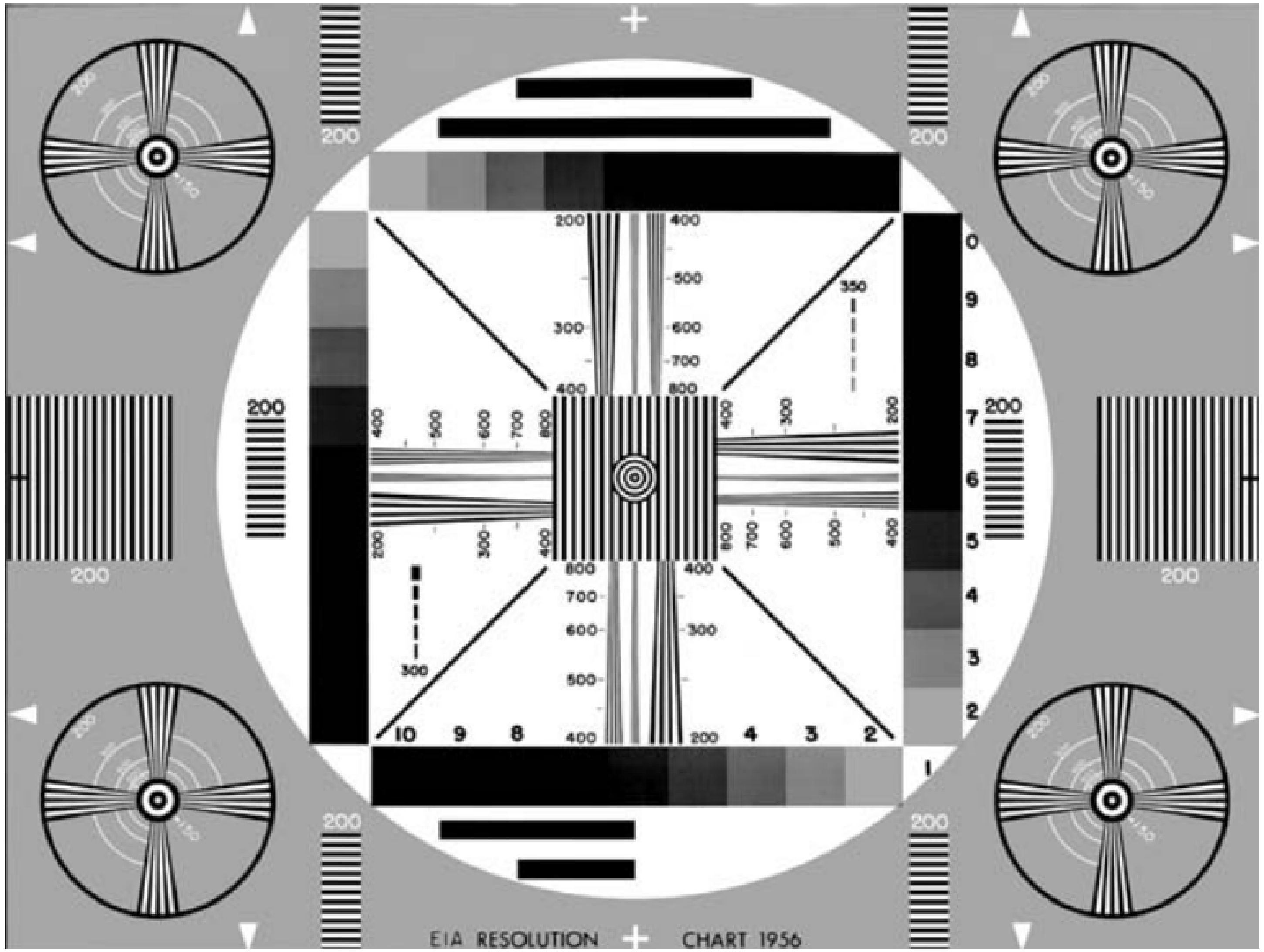
Juan A. Michell Martín
Gustavo A. Ruiz Robredo

Departamento de Electrónica y Computadores

Este tema se publica bajo Licencia:
Creative Commons BY-NC-SA 4.0

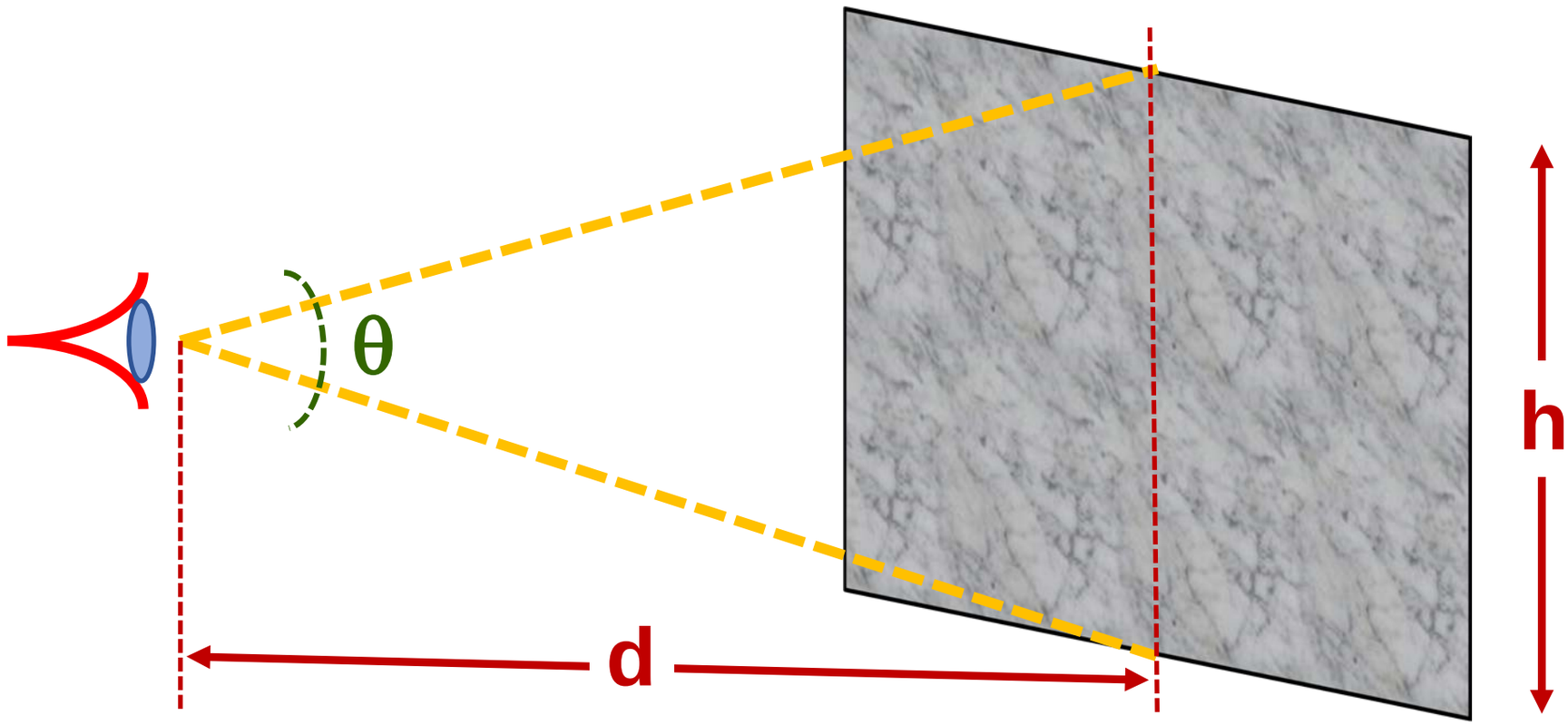
PERCEPCIÓN VISUAL HUMANA

- La **resolución** es la habilidad de separar dos pixels contiguos, es decir, identificar detalles a través de un test de gradientes.

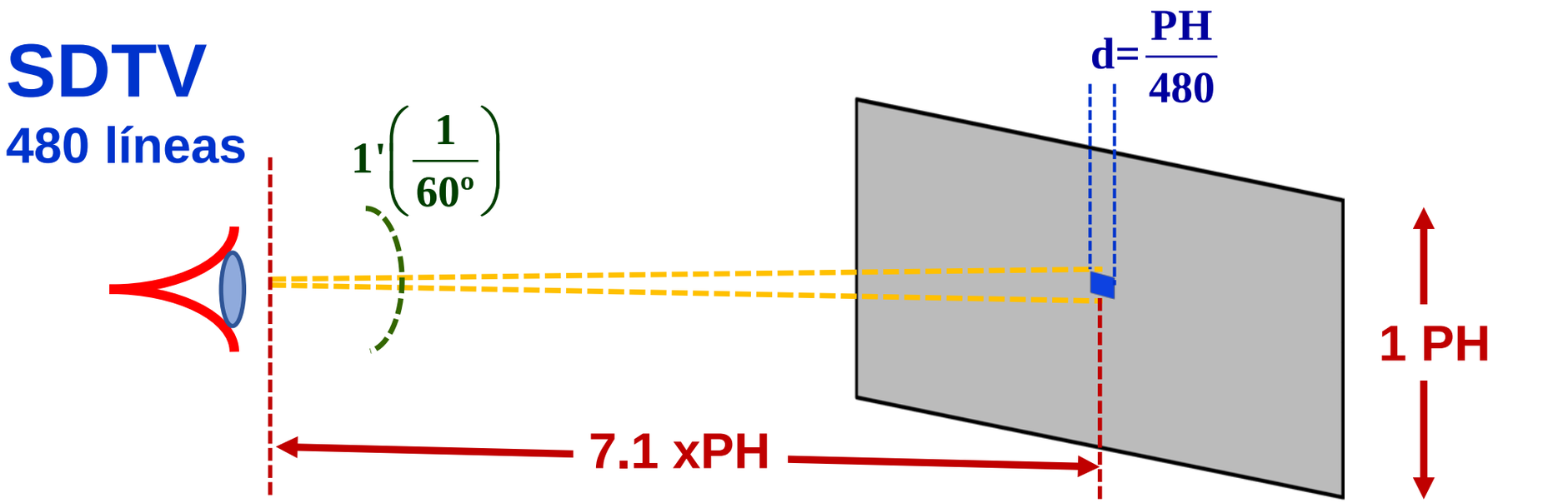


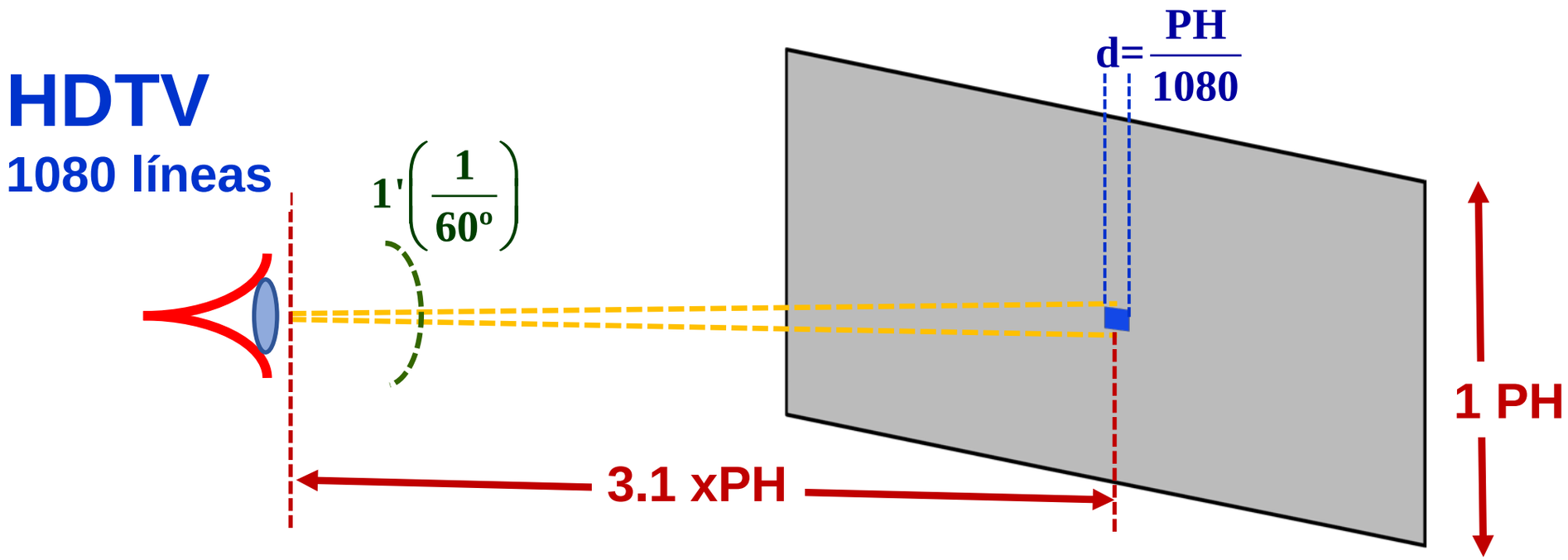
EIA 1956 standard test pattern. www.bealecorner.com

- **La resolución depende de varios factores:**
  - **Altura (h) del monitor**
  - **Distancia (d) entre el observador y el monitor.**
  - **Angulo de visión (θ)**



➢ La distancia óptima de visión para diferentes pantallas.

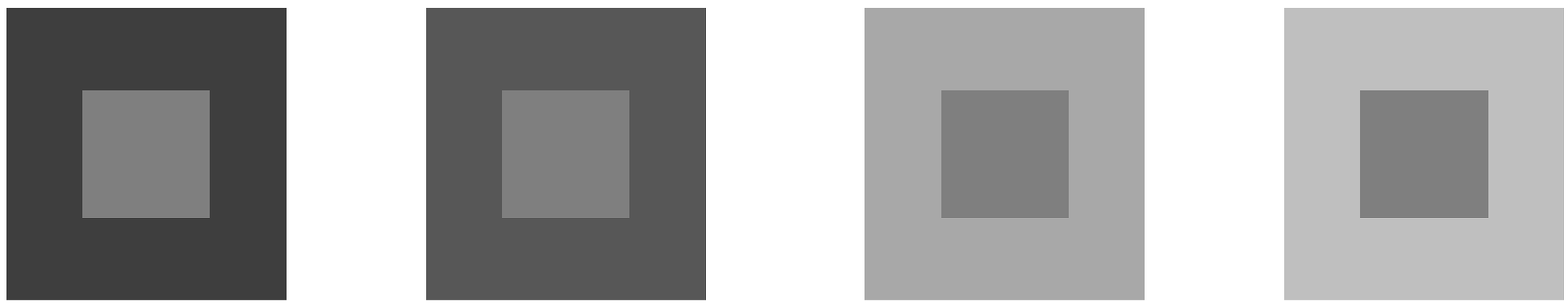




Percepción de la luminosidad

- **La luminosidad se define como el atributo de la sensación visual por el que un área aparece emitir más o menos luz.**

- **La luminosidad en el ojo humano es subjetiva adaptándose a determinados niveles de luminosidad de las zonas de contorno.**



- **En 1865, Ernst March detectó a través de las bandas de Ernst que la percepción de luminosidad del ojo humano no es una función simple.**

- **Explica como la reproducción de bordes no es un requerimiento crítico. Esta propiedad es aprovechada en el tratamiento de imágenes.**

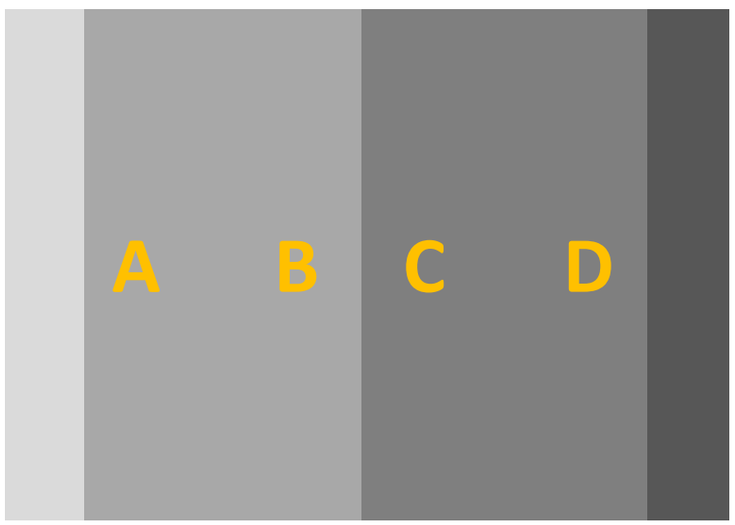
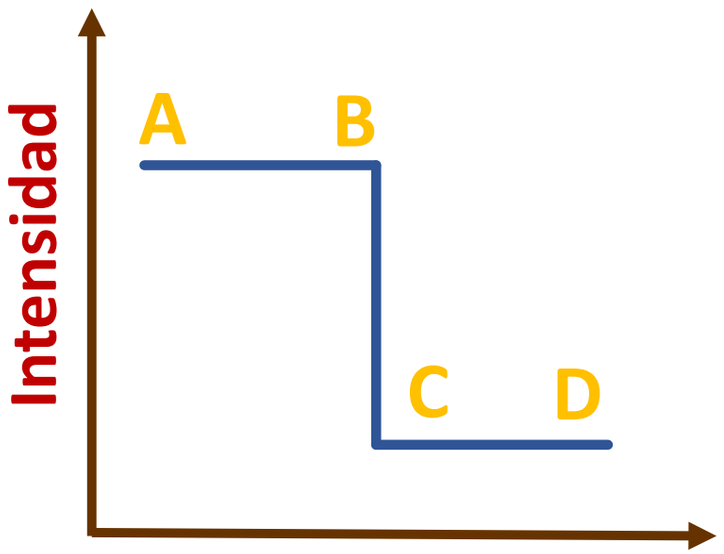


Imagen con niveles de grises



Distribución de luminosidad

Percepción de luminosidad

Sensibilidad de contraste

- La sensibilidad de contraste del ojo humano se define como la diferencia de brillo que puede ser detectado por un observador.

- Después de diferentes experimentos se ha comprobado que para distinguir dos niveles de luminosidad se requiere que

$$\frac{\Delta Y}{Y} = 1\%$$

Implicaciones del ojo humano

- **El ojo humano es más sensible a variaciones de alto contraste. Esto significa que altas variaciones de intensidad (por ejemplo bordes) son fácilmente detectables.**

- **El ojo humano es más sensible a cambios de luminosidad que afecta a grandes áreas que si esos cambios se producen en pequeñas áreas.**

- **El ojo humano es más sensible a imágenes persistentes de larga duración.**

IMAGEN DIGITAL

- Una *imagen* puede definirse como una función 2D $f(x,y)$, en la que x e y son las coordenadas espaciales y el valor de f en cualquier par de coordenadas se denomina *intensidad* de la imagen en ese punto.

- Una *imagen digital* es una imagen en la que tanto las coordenadas como los valores de la intensidad son cantidades discretas.

- **Digitalización** de una imagen: *muestreo espacial*, para seleccionar un número finito de puntos, y *cuantificación* de la intensidad en cada punto, para limitar sus valores a un número finito.

Una imagen digital es la representación de un *array* finito de puntos, denominados *píxeles*, que en general son números enteros.

Imagen en color de 1080×1920 puntos © www.nasa.gov

ADQUISICIÓN DE IMÁGENES DIGITALES: SENSORES

- **Un sensor de imagen convierte energía electromagnética (luz) en señales eléctricas que pueden ser procesadas, mostradas, interpretadas como imágenes.**

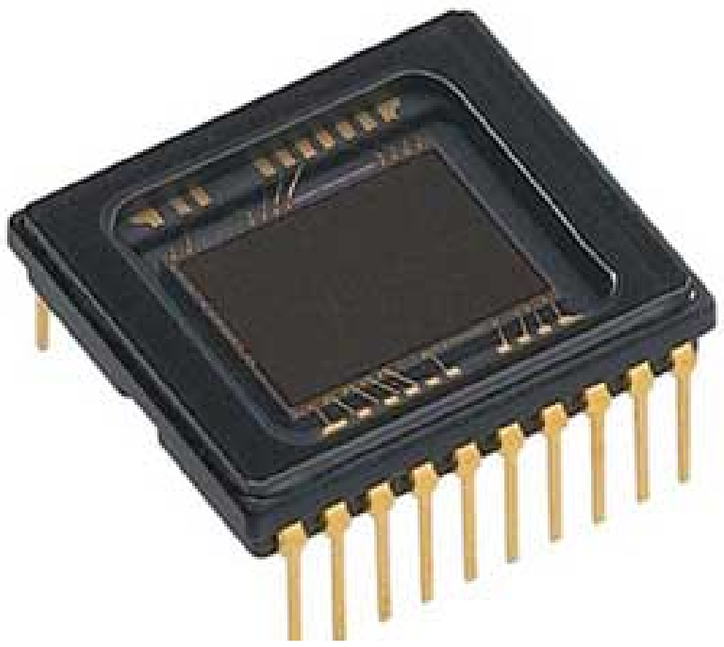
- **Existen actualmente dos tecnologías dominantes para sensores de imagen formados en esencia por semiconductores de metal-óxido (MOS) que están distribuidos en forma de matriz:**

  - ✓ **CCD (Charge Coupled Device)**

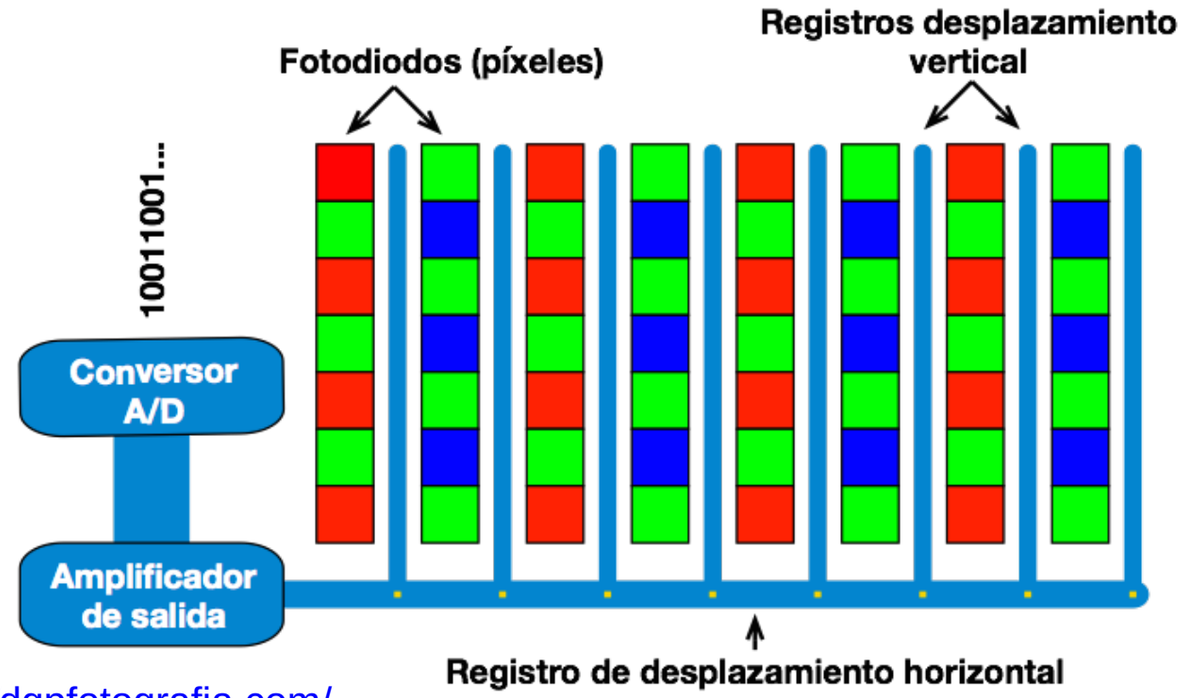
  - ✓ **CMOS (Complementary Metal Oxide Semiconductor)**

Sensor CCD

- **Boyle y Smith recibirían el Premio Nobel de la física por este invento en el año 2009.**
- **En el CCD, se convierte las cargas de las celdas de la matriz en voltajes y entrega una señal analógica en la salida, que será posteriormente digitalizada por la cámara.**
- **Necesidad de un chip adicional que se encargue del tratamiento de la información proporcionada por el sensor, lo que se traduce en un gasto mayor y equipos más grandes.**



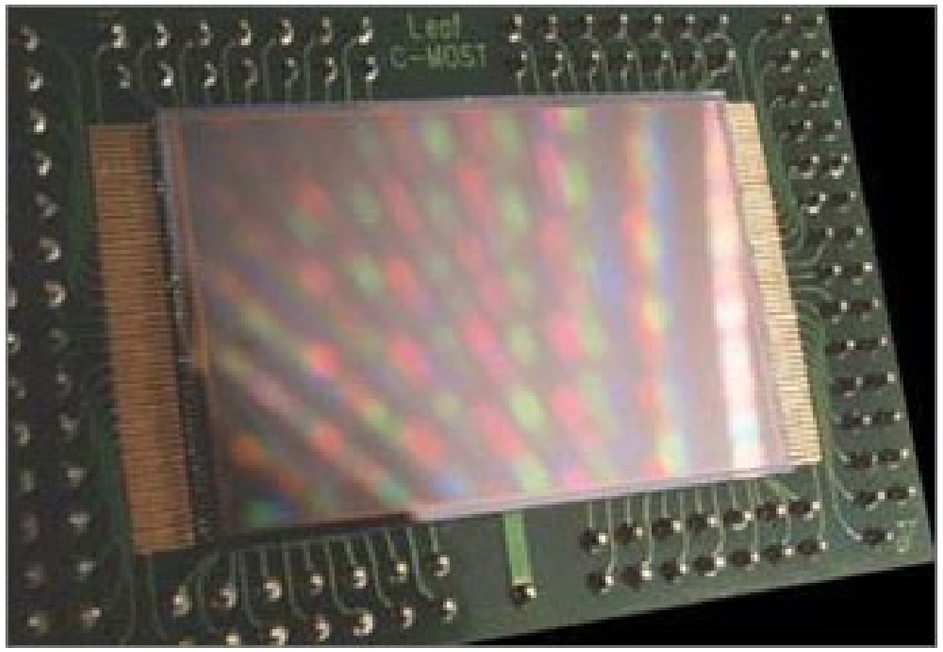
http://www.digitalfotored.com



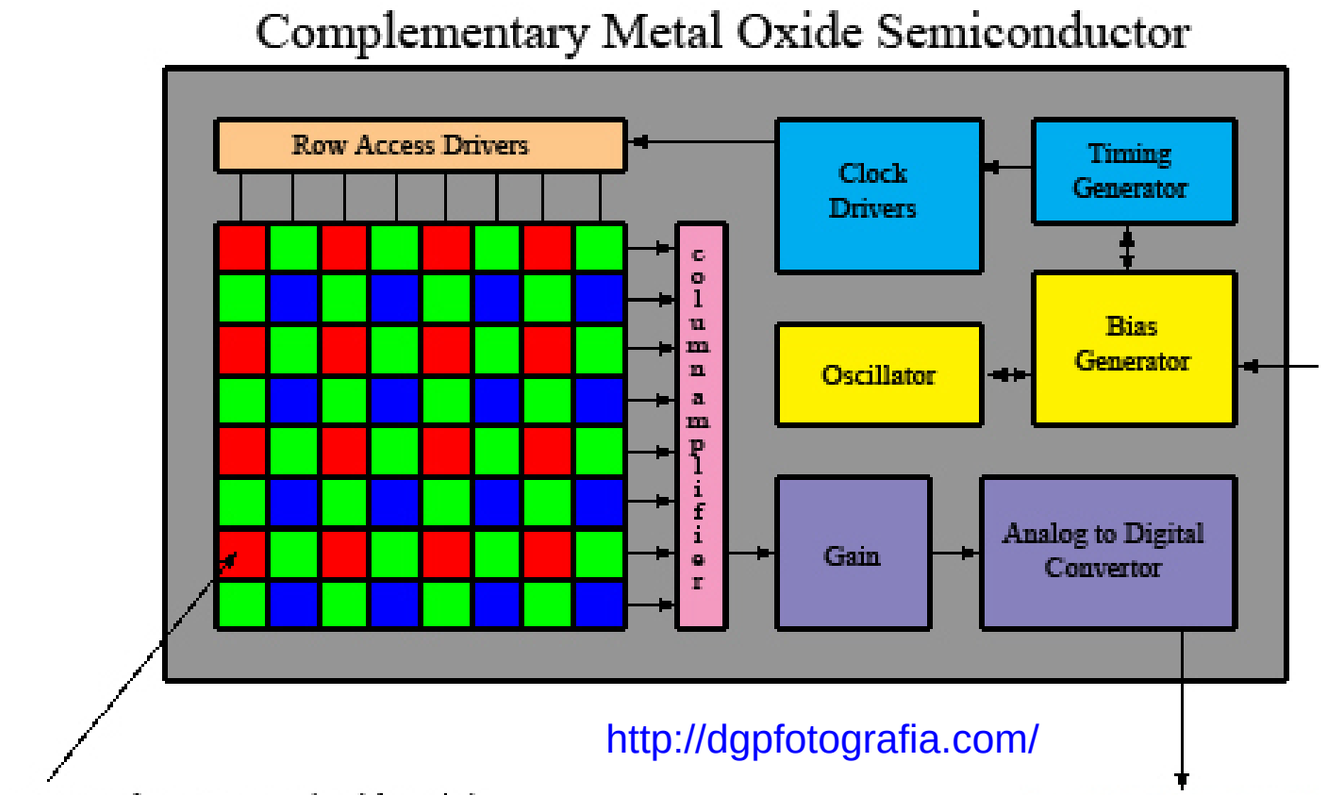
http://dgpfotografia.com/

Sensor CMOS

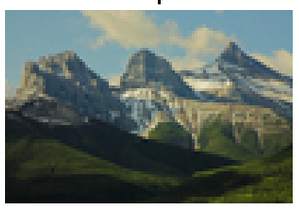
- Cada celda es independiente y no es necesaria la digitalización.
- Son más baratos, sensibles a la luz y rápidos.
- Los sensores CMOS se han impuesto a los CCD.



http://www.digitalfotored.com



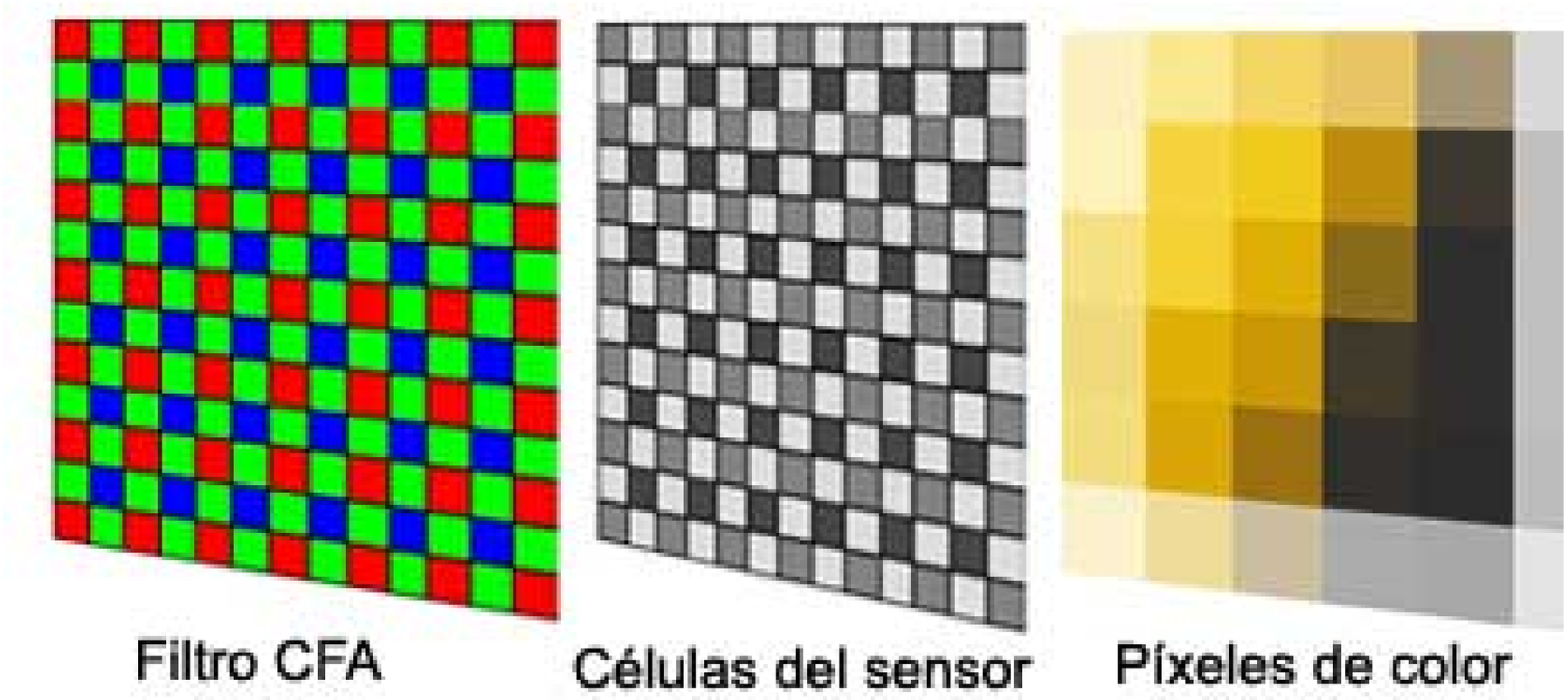
http://dgpfotografia.com/



El color en el sensor

- **Los sensores de imagen son monocromos con escala en grises desde el blanco al negro.**

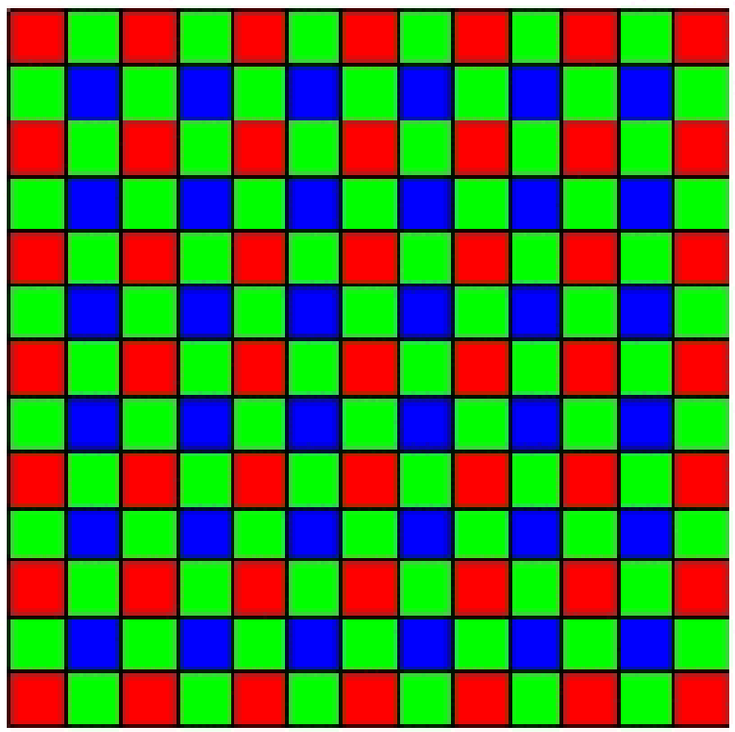
- **Para captar color, la luz atraviesa diferentes filtros de colores. El filtro CFA es uno de los más comunes.**



http://www.digitalfotored.com

Filtro Bayer

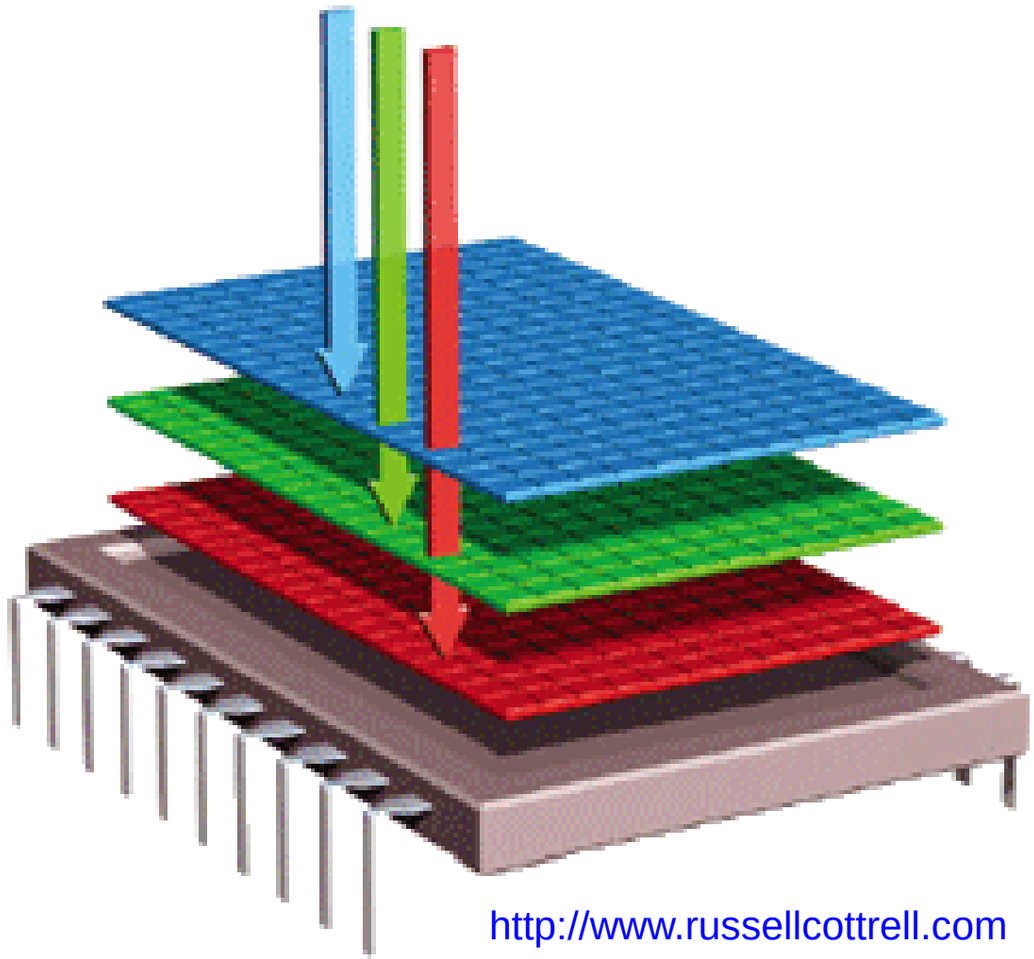
- El filtro, máscara o mosaico de Bayer es un tipo de matriz de filtros, rojos verdes y azules
- El mosaico de Bayer está formado por un 50% de filtros verdes, un 25% de rojos y un 25% de azules. Interpolando dos muestras verdes, una roja, y una azul se obtiene un pixel de color. La razón de que se use mayor cantidad de puntos verdes es que el ojo humano es más sensible a ese color.



Filtro Foveon X3

- El nuevo sensor Foveon X3 fue creado por (Foveon).

- Usa fotodetectores que se encuentran incorporados en cada píxel, de forma que cada uno absorbe una luz diferente: Azul, Verde y Roja.



http://www.russellcottrell.com

IMAGEN MONOCROMA

Una imagen monocroma se representa por una matriz bidimensional de números enteros F o mediante una función discreta $f(x, y)$.

$$\mathbf{F} = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,\mathrm{N}-1) \\ f(1,0) & f(1,1) & \dots & f(1,\mathrm{N}-1) \\ \dots & \dots & \dots & \dots \\ f(\mathrm{M}-1,0) & f(\mathrm{M}-1,1) & \dots & f(\mathrm{M}-1,\mathrm{N}-1) \end{bmatrix}$$

Tamaño: M×N, M y N enteros

x: índice de fila (1 a M en Matlab)

y: índice de columna (1 a N en Matlab)

$f(i, j)$: intensidad o nivel de gris del *píxel* (i, j)

- **Los elementos (*píxeles*) representan tonos grises. Generalmente los *píxeles* se codifican con 8 *bits* (8 *bpp*), lo que significa utilizar una paleta de 256 tonos grises que van desde el negro (0) al blanco (255).**

- **En Matlab se representan utilizando diferentes *tipos* (*class*) *de datos*:**

  `uint8` (entero: 0 a 255)
  `uint16` (entero: 0 a 65.535)
  `double` (doble: 0.0 to 1.0)

Imagen gris de 1080×1920 puntos y 8 *bpp*

IMÁGENES EN COLOR RGB

- ***Arrays* 3D de *píxeles* de color, formados por la *terna* de componentes, rojo (R), verde (G) y azul (B), correspondiente a su localización espacial.**

- **Profundidad de bits (*bit depth*): número de bits utilizados para representar cada pixel de color.**

- **Imagen RGB 24: cada componente de color se codifica con 8 bits, es decir, 3x8=24 bits cada pixel. (Imagen monocromática, 8 bits por cada pixel).**

http://calibraciontv.blogspot.com.es/

Imagen RGB

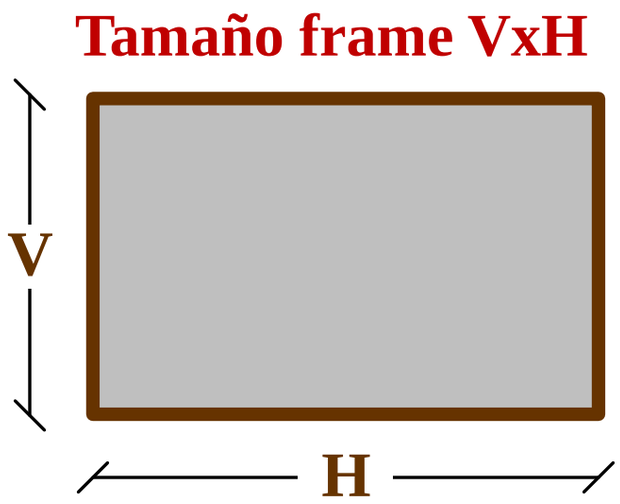
Componente R

Componente G

Componente B

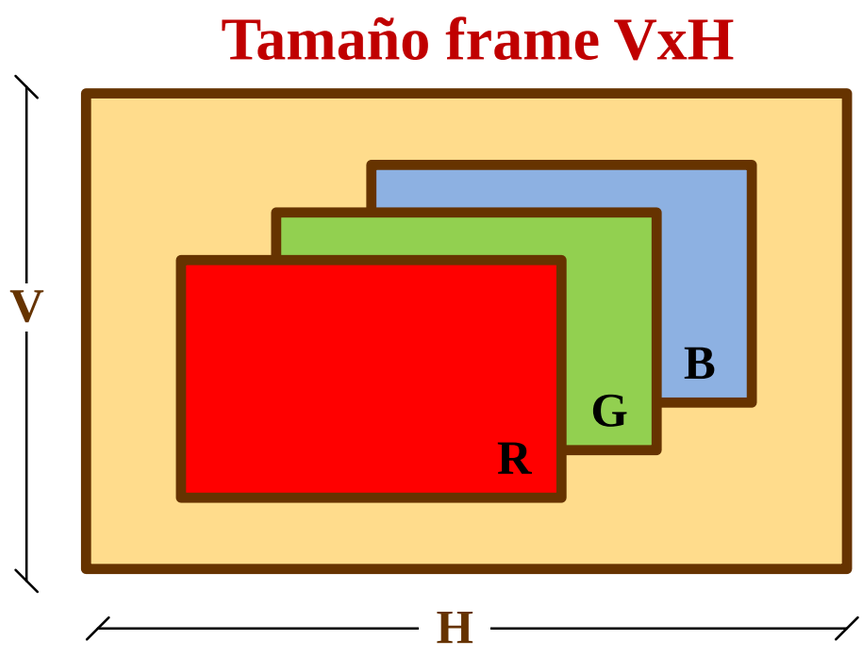
Almacenamiento en Matlab

➢ Imagen monocromática



Matlab 2D: I (1:V, 1:H)

➢ Imagen RGB



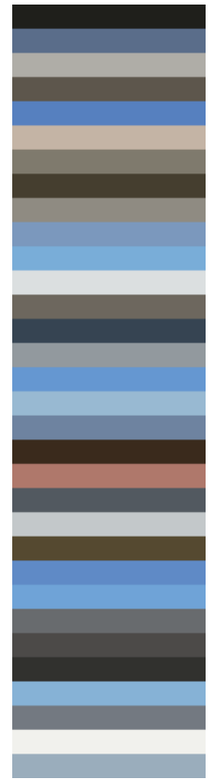
Matlab 3D: I (1:V, 1:H, 1:3)

R: I(1:V, 1:H, 1)
G: I(1:V, 1:H, 2)
B: I(1:V, 1:H, 3)

IMÁGENES INDEXADAS

***Arrays* 2D de la misma dimensión que la imagen, cuyos elementos son *índices* (*punteros*) que apuntan a las filas de un *array* 2D denominado *mapa de color* (*color map*), que contiene la lista de colores utilizados en la imagen.**

Imagen de color indexada



Mapa de color

IMÁGENES BINARIAS

***Arrays lógicos* 2D: 1 *bpp*; usualmente 0 es el color negro y 1 el blanco.**

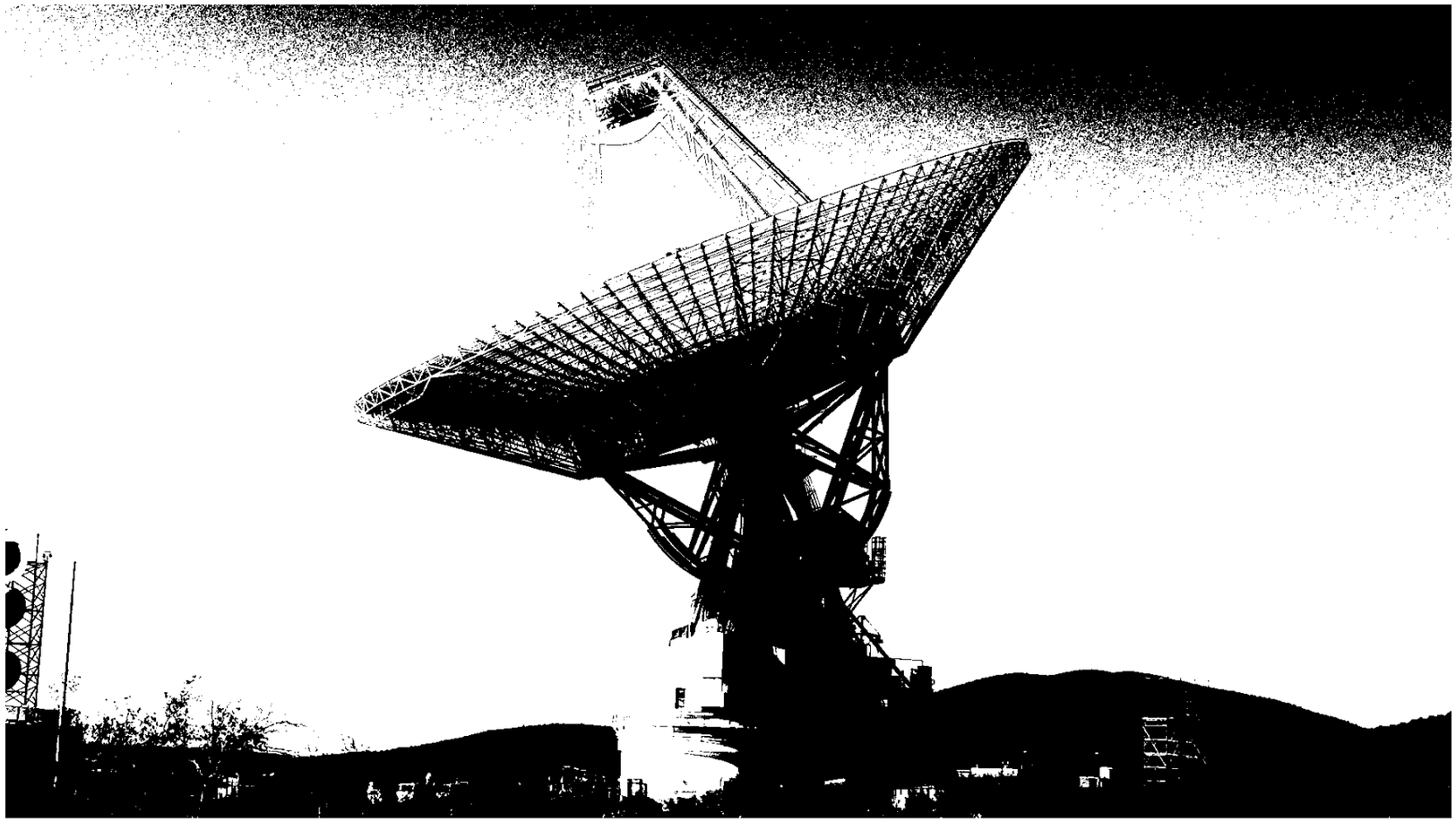


Imagen binaria (1 *bpp*) de 1080×1920 puntos

COMPRESIÓN DE IMAGEN

- **Codificación orientada a reducir el tamaño de almacenamiento y la velocidad de transmisión.**

  - **CON PÉRDIDAS:**
    Se admite un cierto deterioro de la calidad visual de la imagen resultante a cambio de reducir el tamaño de almacenamiento; ej. imágenes fotográficas de propósito general.

  - **SIN PÉRDIDAS:**
    La imagen se codifica conservando toda su calidad; se utiliza siempre que sea necesario conservar todos los detalles de la imagen; ej. imágenes médicas.

Ejemplo de compresión de imagen JPEG

Imagen original: color RGB, tamaño 1080×1928

IMAGEN	FORMATO	CALIDAD	TAMAÑO
PIA13277.tif	TIFF	TOTAL	6.084 KB
PIA13277_Max.jpg	JPEG	MÁXIMA	1.690 KB
PIA13277_High.jpg	JPEG	ALTA	475 KB
PIA13277_Med.jpg	JPEG	MEDIA	202 KB
PIA13277_Low.jpg	JPEG	BAJA	92 KB

FICHEROS DE IMAGEN

Una cabecera que contiene las propiedades, altura y anchura, bits por *píxel*, tipo de compresión, etc., seguida de los *píxeles*, generalmente comprimidos.

Formatos de fichero de imagen

JPEG: imagen con calidad fotográfica; elevada compresión con mínima pérdida de calidad perceptiva.

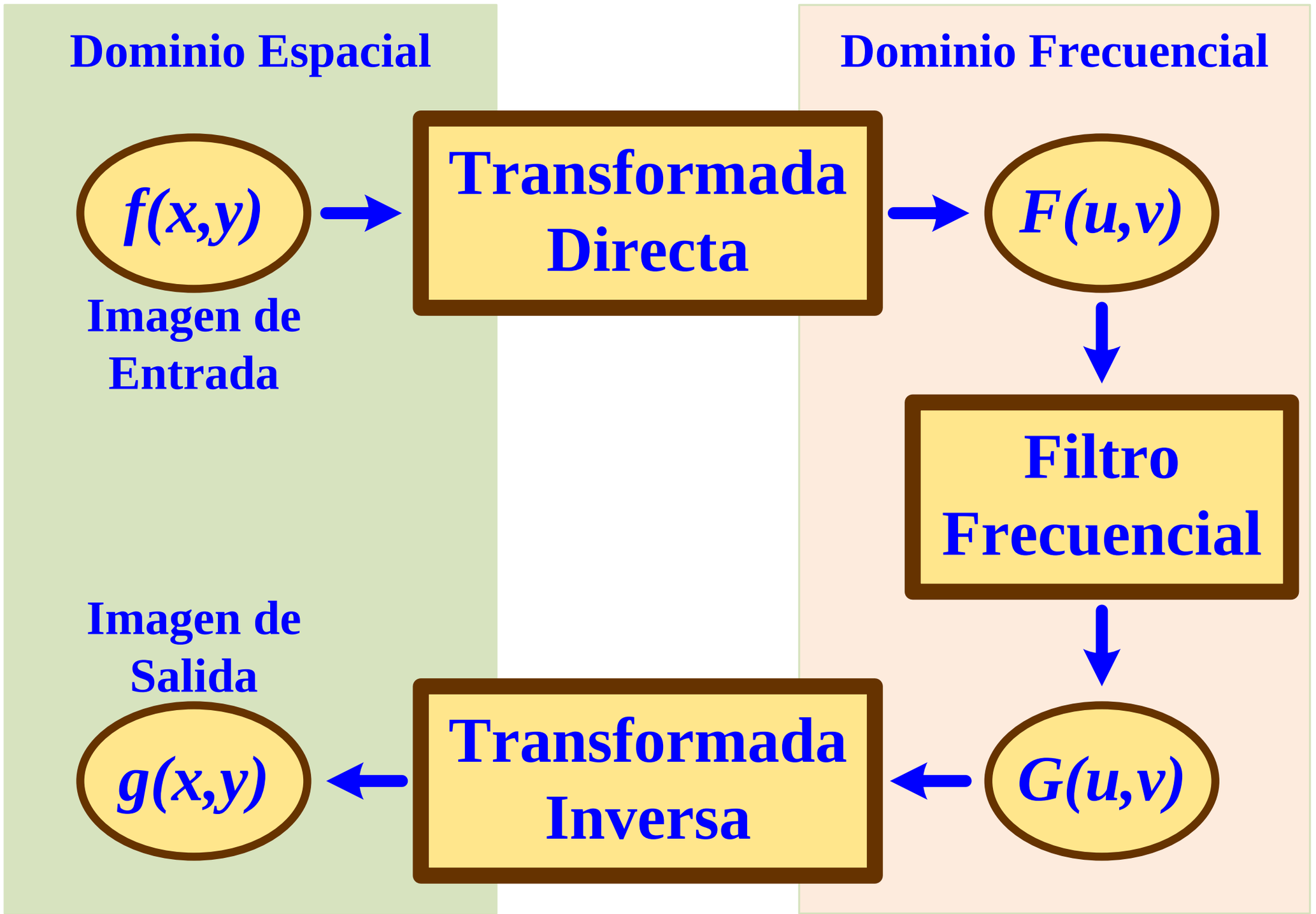
TIFF: formato sofisticado con muchas opciones y capacidades; soporta cinco esquemas de compresión diferentes.

Otros: RAW (cámaras fotográficas), BIN (*píxeles* en bruto, sin cabecera), PPM, BMP, GIF, PNG.

PROCESADO DE IMAGEN

- **Operaciones en el Dominio Espacial**
  - **Operaciones de Punto**
    - **Transformación de Intensidad**
  - **Operaciones de Entorno**
    - **Filtrado Espacial**
  - **Operaciones con Imágenes**
    - **Combinación de imágenes**
  - **Operaciones Geométricas**
    - **Redimensionamiento y Rotación**
- **Operaciones en el Dominio Frecuencial**
  - **Filtrado en el Dominio de la Frecuencia**

FILTRADO FRECUENCIAL



OPERACIONES DE PUNTO

Se aplica el mismo operador de transformación *T* a todos los *píxeles* de la imagen, con independencia de su localización y de los valores de su entorno.

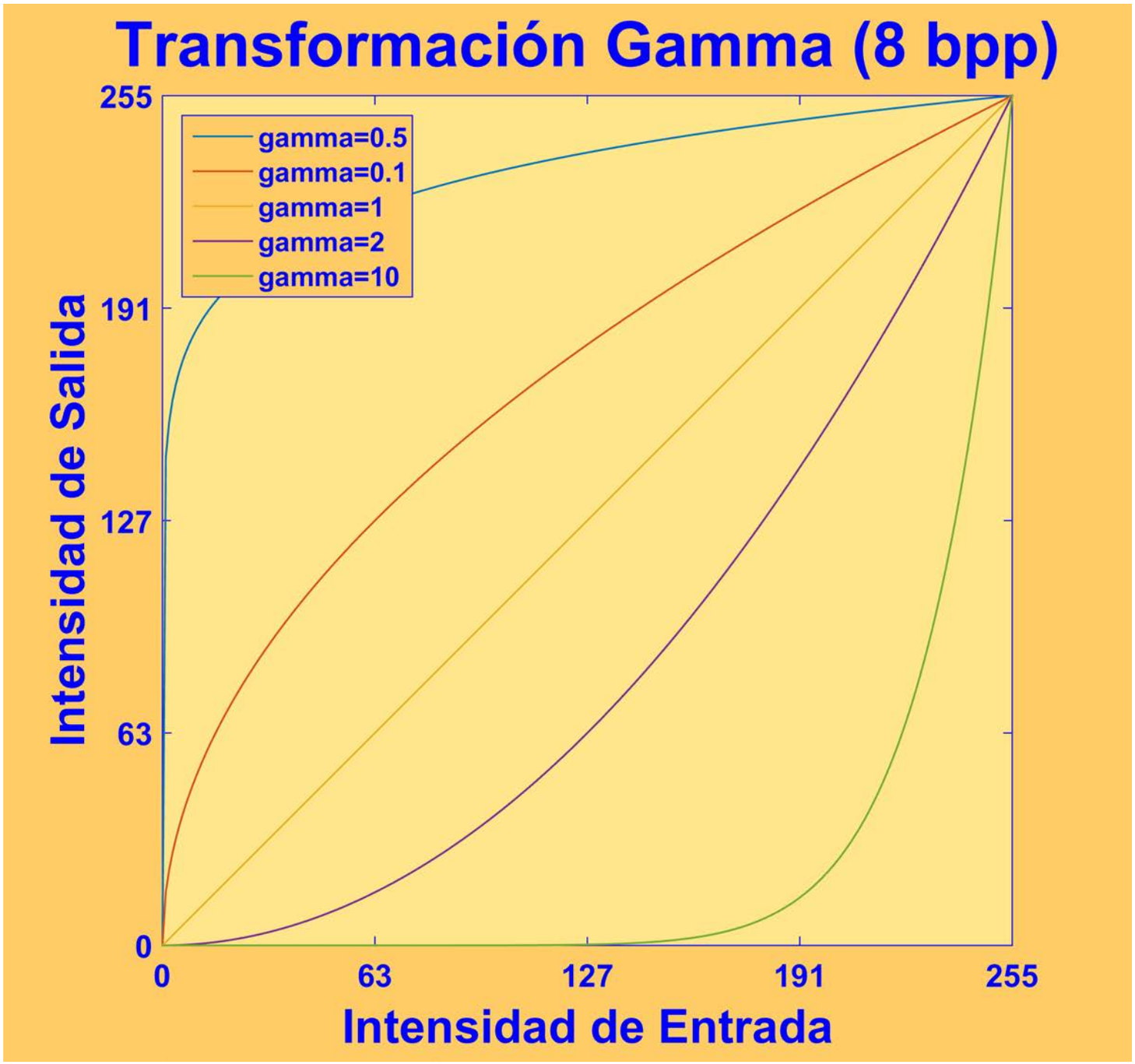
$$g(x,y) = T\left(f(x,y)\right)$$

$f(x,y)$: **imagen original** $\quad$ $g(x,y)$: **imagen procesada**

$$s = T[r]$$

r : **nivel de intensidad de un pixel en la posición** (i, j) **de** f

s : **nivel de intensidad resultante en la posición** (i, j) **de** g



$$s = c \cdot r^{\gamma}$$

$$c = 1$$

$$r, s \begin{cases} \textit{niveles de gris} \\ \textit{normalizados} \end{cases}$$

Transformación gamma: c=1, γ=2.

Imagen Original

Imagen Resultante

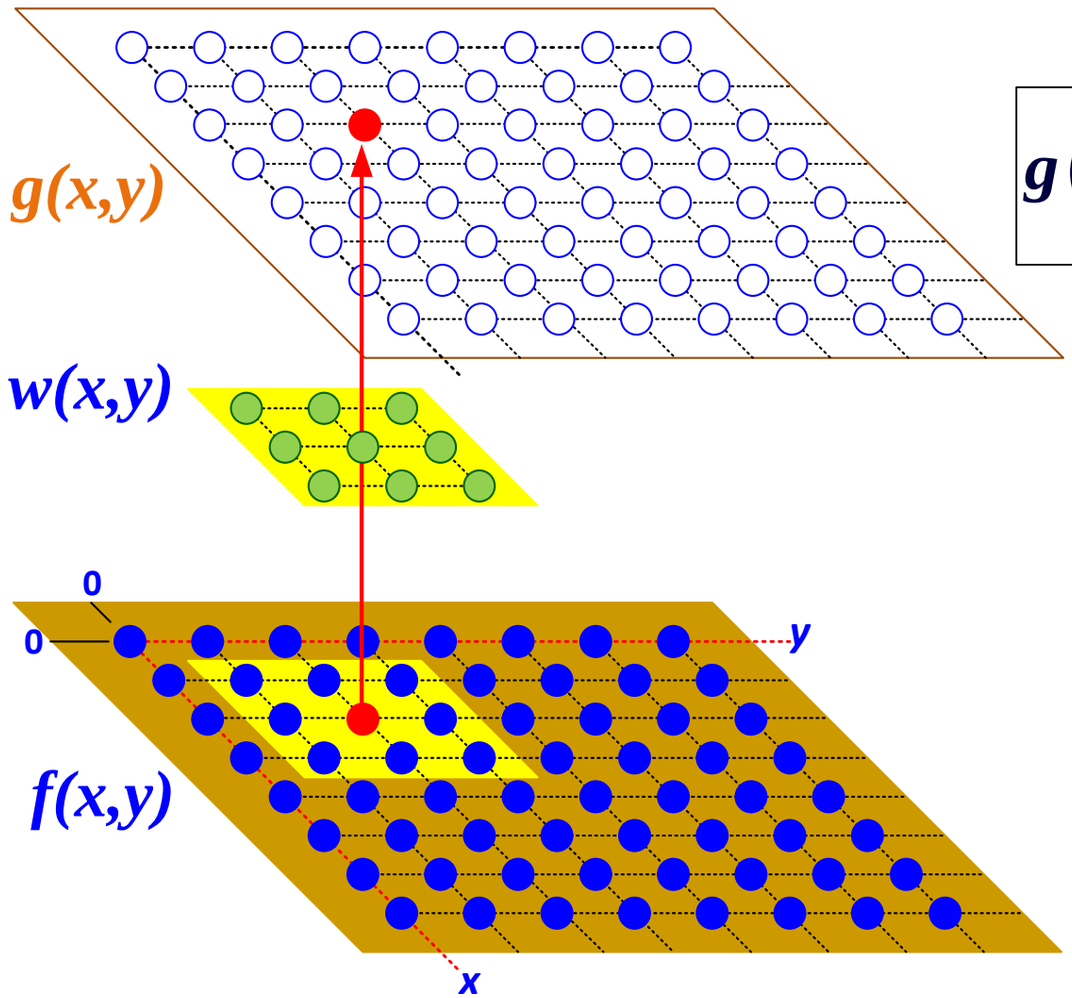
Ejemplo de otras operaciones:

- **Histogramas:** Matlab **Ejemplo 7**
- **Transformación espacial:** Matlab **Ejemplo 8**

OPERACIONES DE ENTORNO

Filtrado

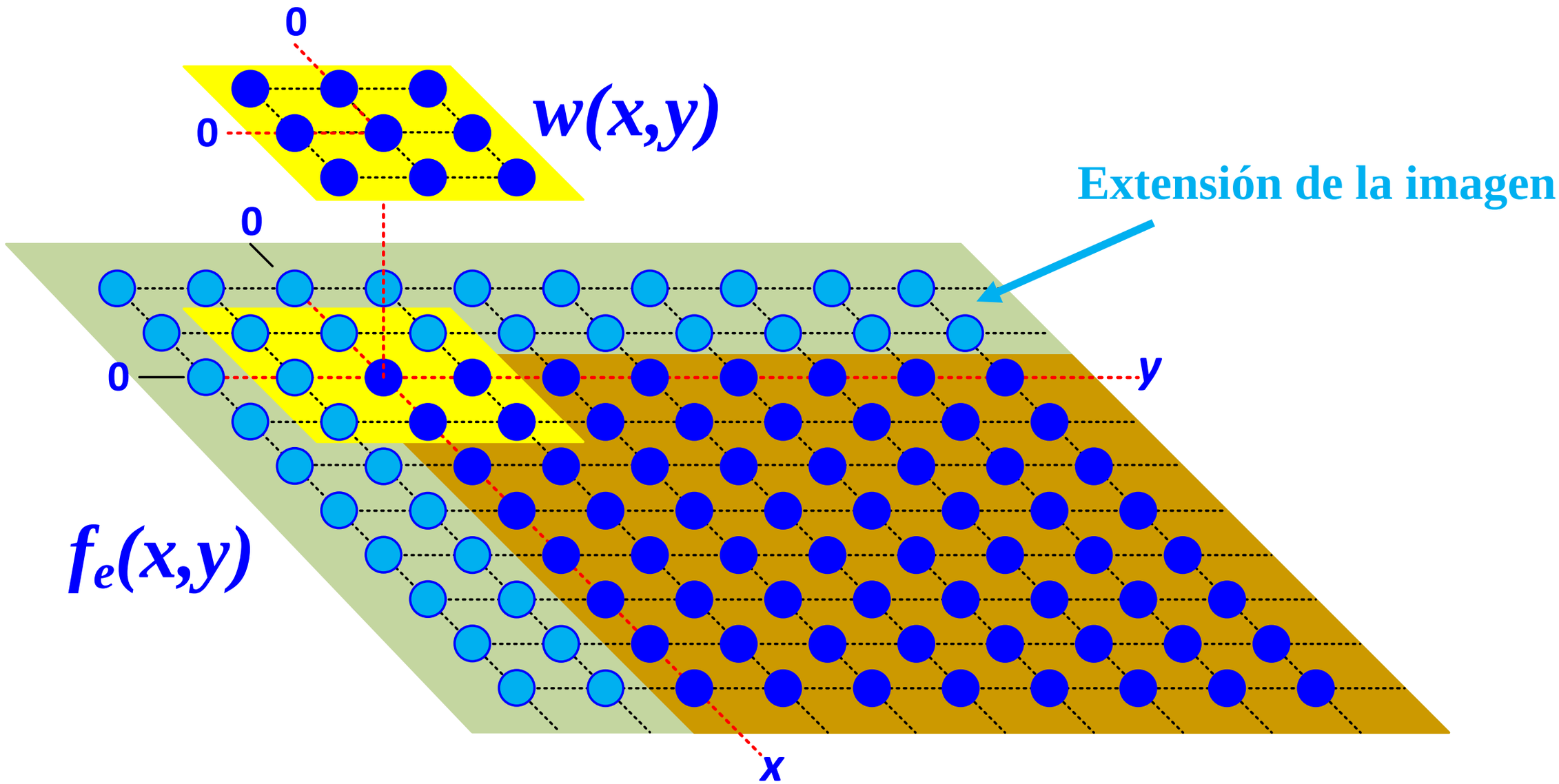
Cada *píxel* de salida, $g(x, y)$, se calcula en función del valor original en esa posición, $f(x, y)$, y de los valores vecinos en el entorno de $f(x, y)$, ponderados por una máscara $w(x, y)$ de forma correlacionada.



$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w(i,j)\, f(x+i, y+j)$$

Operaciones de contorno (boundary)

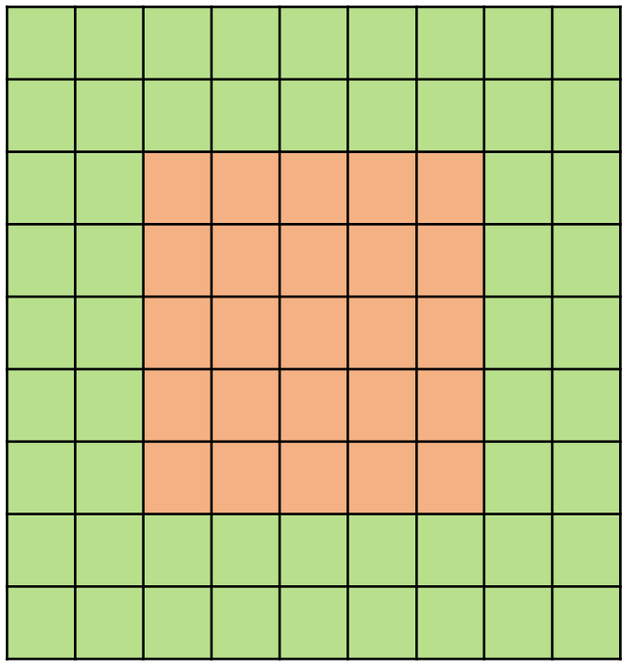
Las operaciones en los bordes exigen extender la imagen (*padding*):



Tipos de extensión de la imagen :

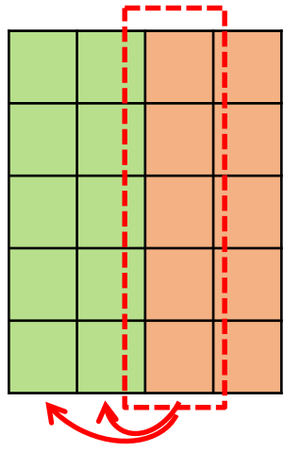
a) añadir ceros o un nivel constante (*Padding*).
b) repetir los bordes (*Replicate*).
c) simetría en los bordes (*Simmetry*).
d) extensión periódica de la imagen (*Circular*).

Padding

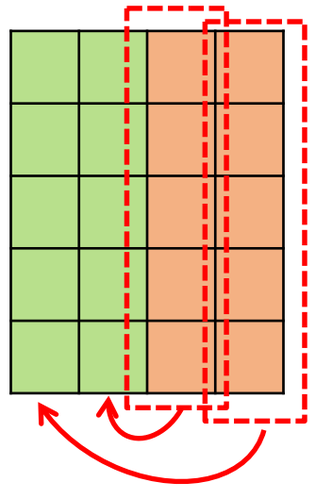


Relleno de un mismo valor X (defecto 0)

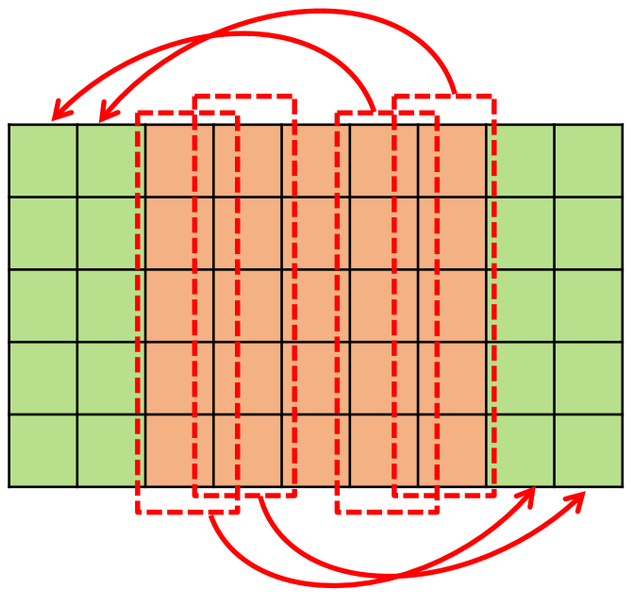
Replicate



Simmetry



circular



Convolución

Convolución entre una imagen *f* de tamaño *M*×*N* y una mascara *w* de tamaño *m*×*n*:

$$g(x,y) = w(x,y) \otimes f(x,y)$$

$$g(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i,j)\, f(x-i, y-j)$$

$$a,b = 1,2,\ldots \quad , \quad m = 2a-1, \quad n = 2b-1$$

$$m << M, \quad n << N$$

Extensión de *f*:

$$f \rightarrow f_e \begin{cases} M \rightarrow M_e = M + a \\ N \rightarrow N_e = N + b \end{cases}$$

Ejemplo de correlación

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Imagen original

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Imagen ampliada (boundary: padding 0)

1	2	3
4	5	6
7	8	9

w

Rotación 180º

9	8	7
6	5	4
3	2	1

w convolución

W

1	2	3
4	5	6
7	8	9

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Primera posición de w

W

1	2	3
4	5	6
7	8	9

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Segunda posición de w

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

unpadding

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

Resultado correlación

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

unpadding

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Resultado convolución

Ejemplos de matrices de filtrado

Dependiendo de la máscara, una misma operación básica puede difuminar (desenfocar), mejorar la nitidez (enfocar), detectar bordes o eliminar ruido.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Máscara de Desenfoque

0	-1	0
-1	5	-1
0	-1	0

Máscara de Enfoque

-1	0	1
-2	0	2
-1	0	1

Máscara de Bordes Vertical

Imagen Original

Imagen Desenfocada

Imagen Enfocada

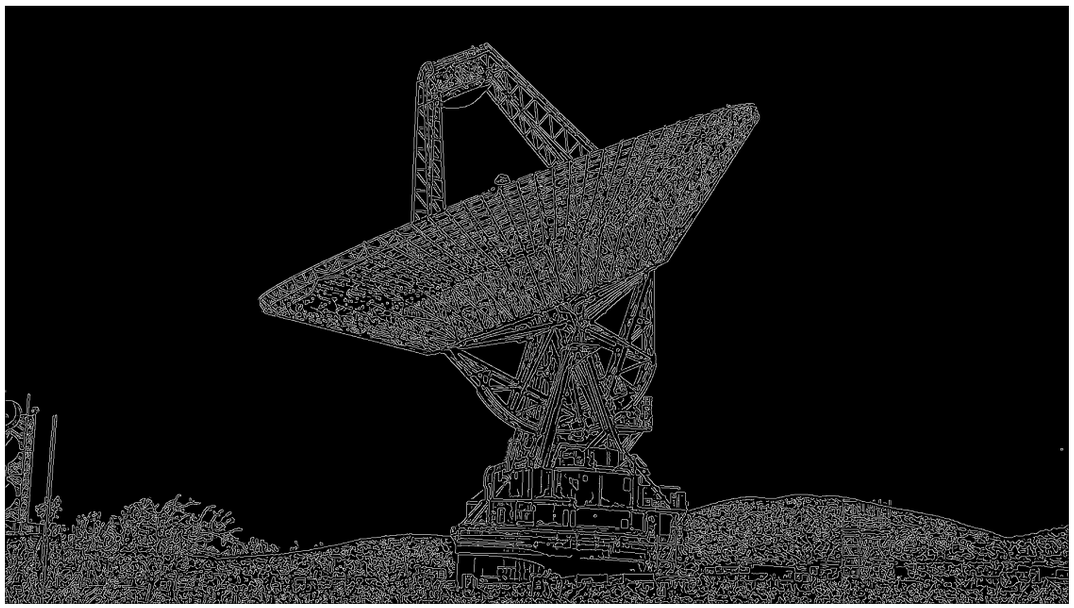


Imagen de Bordes

HERRAMIENTA DE FILTRADO INTERACTIVA

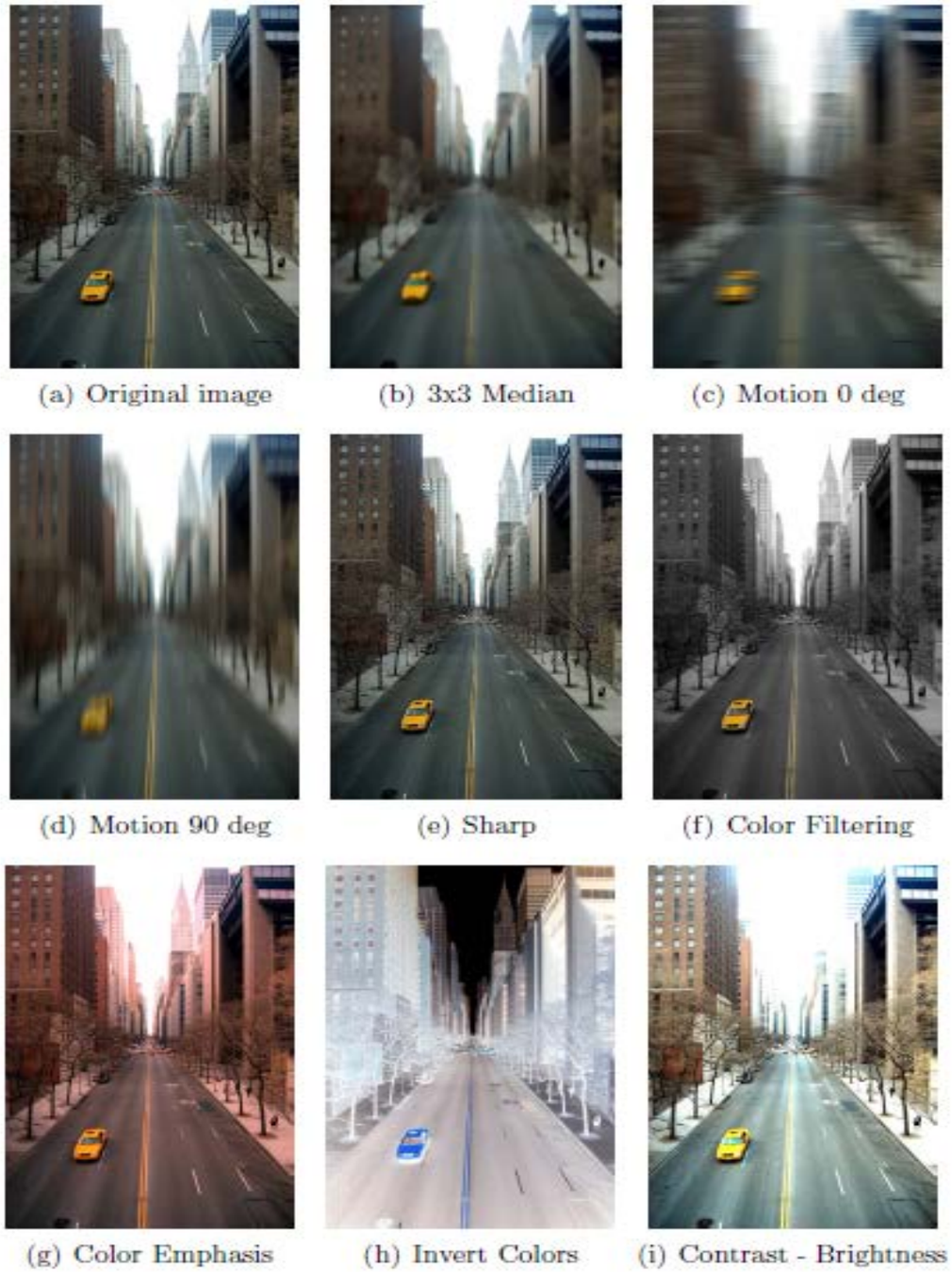
- **En la carpeta filtergui se encuentra una herramienta para realiza diferentes filtrados y procesado a una imagen. Desde Matlab abrir el fichero imageProc.m y ejecutar.**

- **Esta aplicación ha sido desarrollada por:**

Image Processing GUI

Theodoros Giannakopoulos
Dept. of Informatics and Telecommunications
University of Athens

http://www.di.uoa.gr/~tyiannak



(a) Original image (b) 3x3 Median (c) Motion 0 deg

(d) Motion 90 deg (e) Sharp (f) Color Filtering

(g) Color Emphasis (h) Invert Colors (i) Contrast - Brightness

COMBINACIÓN DE IMÁGENES

Imagen Original

Efecto

Imagen Compuesta

```
K=imadd(I,J);
K=imsubstract(I,J);
K=imaabsdiff(I,J);
K=imdivide(I,J);
K=immultiply(I,J);
K=imcomplement(I);
```

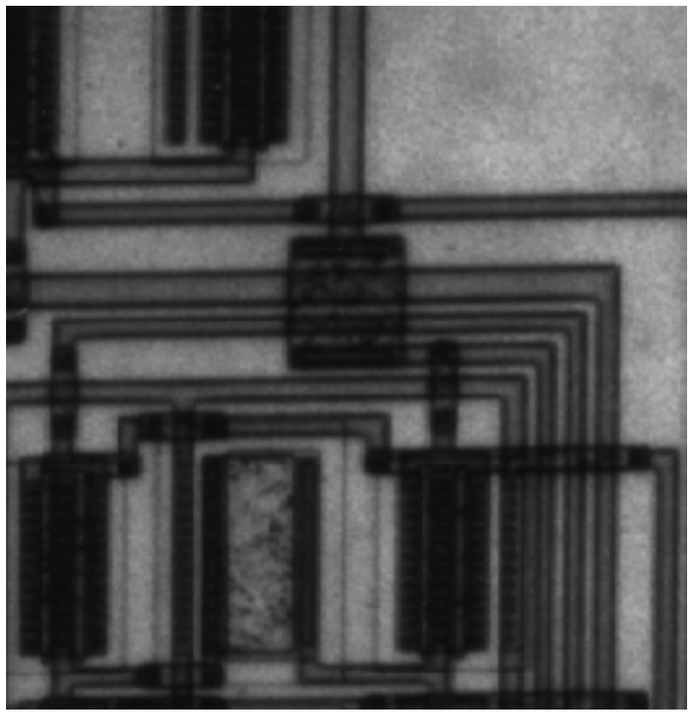
DETECCIÓN DE CONTORNOS

Imagen

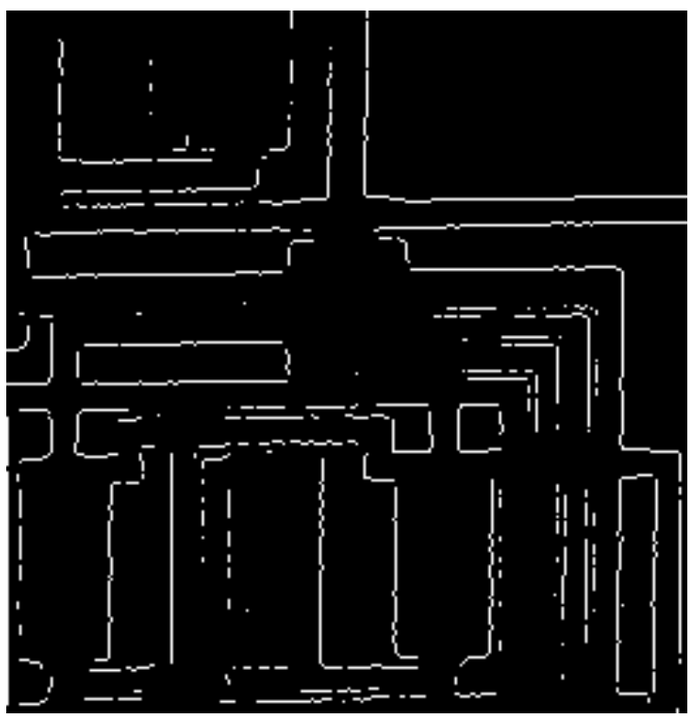
Factor de umbralización

```
K=edge( I , method , threshold );
```

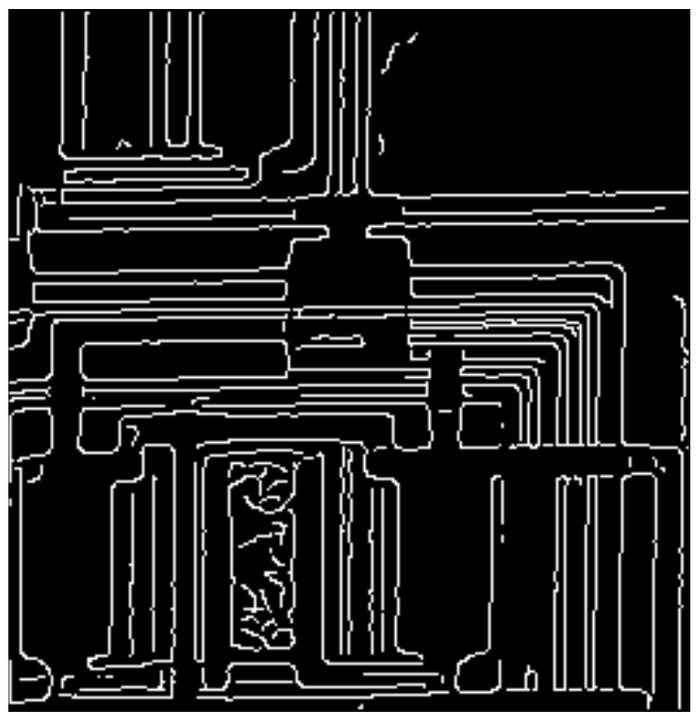
‘sobel’ ‘prewitt’ ’roberts’‘canny’ …



Original



prewitt



canny

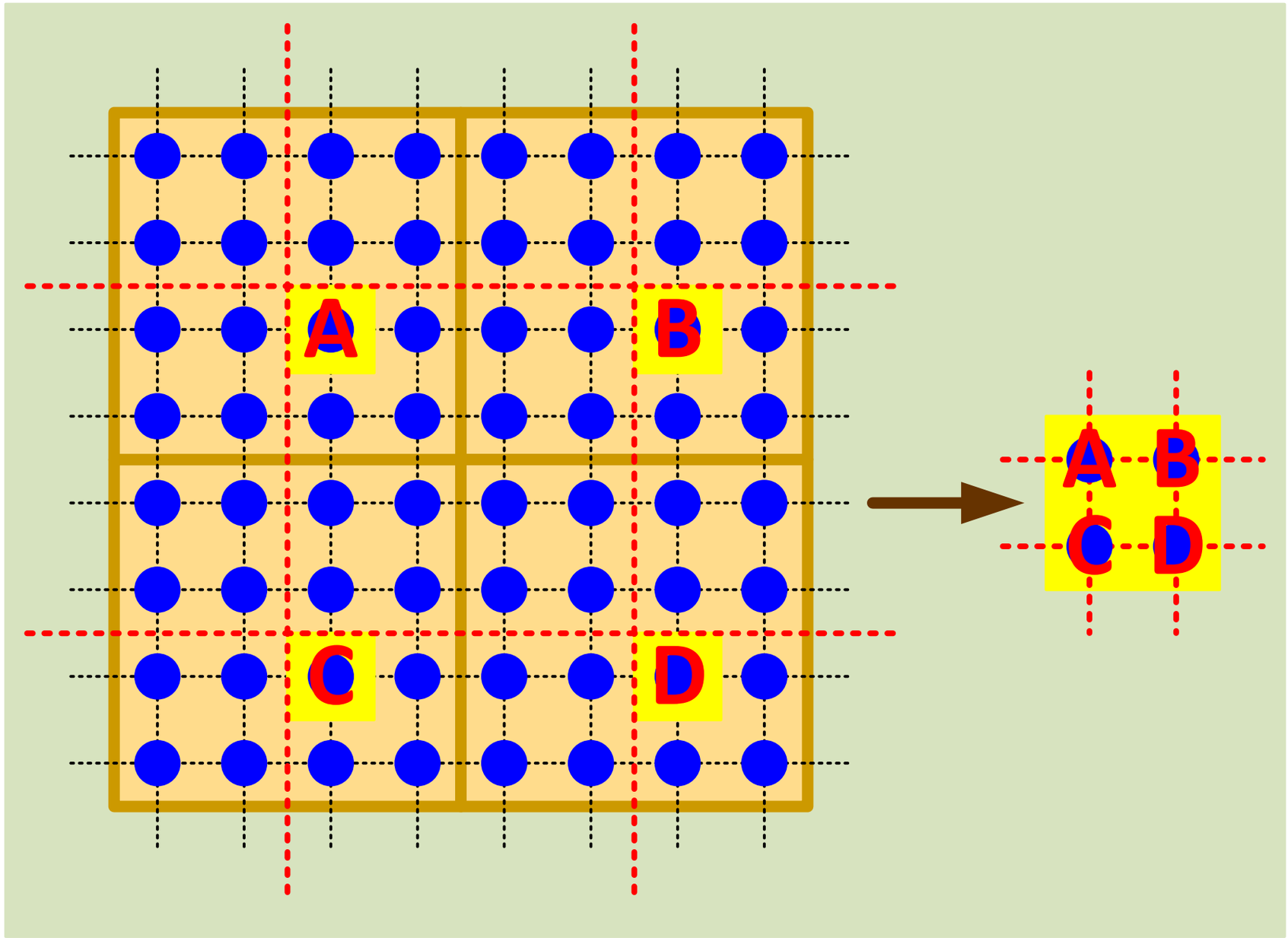
REDIMENSIONAMIENTO DE IMÁGENES

```
                                                    'nearest'
                                                    'Bilinear'
                                                    'bicubic'
                          Imagen
K=imresize( I , scale , method );
```

Factor de escalado → `Factor (p.e. 1.5)`

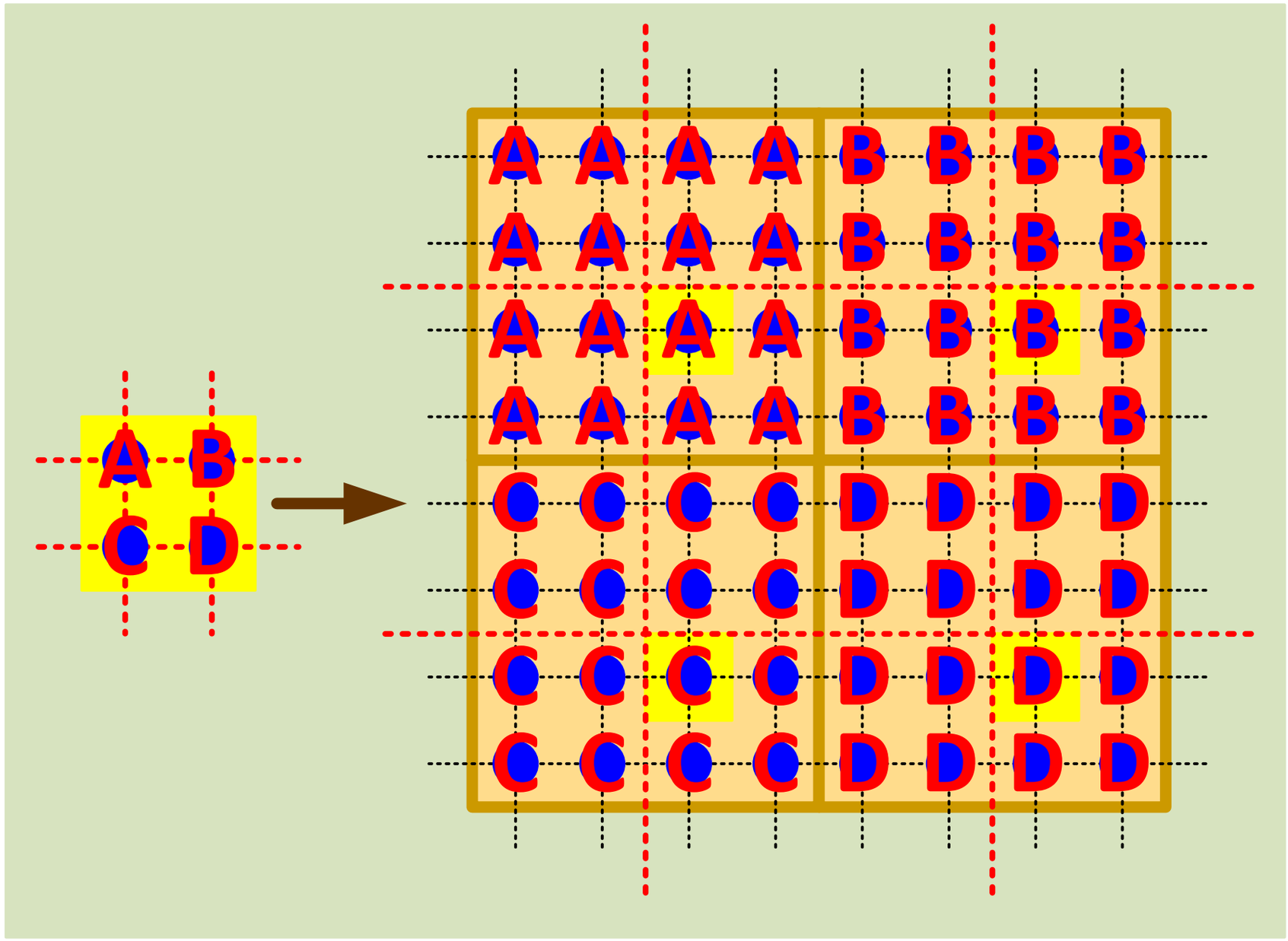
Tamaño final → `[W H] (p.e. [128 128]`

Reducción de tamaño



Reducción del Tamaño a 1/4
Mediante Interpolación de “Orden 0”

Aumento de tamaño



Aumento del Tamaño ×4
Mediante Interpolación de “Orden 0”

Redimensionado mediante interpolación de orden ‘0’

Imagen 1080×1920

Imagen Reducida (1/4)

Imagen Recuperada (×4)

Detalle Central

Redimensionado mediante interpolación cúbica

Imagen 1080×1920

Imagen Reducida (1/4)

Imagen Recuperada (×4)

Detalle Central

MEDIDAS DE CALIDAD DE UNA IMAGEN

La medida de calidad de una imagen es una tarea difícil e imprecisa ya que depende del observador humano que es siempre subjetivo.

Medida de calidad subjetiva

Es un criterio humano. La ITU-R Recommendation BT.500-10 estableció el protocolo DSCQS (*Double Stimulus Continuos Quality Scale*). En este método dos observadores humanos miran dos imágenes y las clasifican como:

Excellent, good, fair, por y bad.

Medida de calidad objetiva

- **Proporciona una alternativa a las medidas de calidad subjetivas.**
- **Es fácil de implementar**

Sea I la imagen de referencia e I' la imagen reconstruida, ambas de dimensión MxN, se define:

- ✓ **El Error Total (E) :**

$$E = \sum_{x=1}^{N} \sum_{y=1}^{M} \left| I(x,y) - I'(x,y) \right|$$

- ✓ **El Error RMS o MSE (*Mean Square Error*):**

$$MSE = \sqrt{\frac{1}{M \cdot N} \sum_{x=1}^{N} \sum_{y=1}^{M} \left| I(x,y) - I'(x,y) \right|^2}$$

- ✓ **El PSNR (*Peak Signal To Noise Ratio*):**

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{(\mathrm{L}-1)^2}{\mathrm{MSE}^2}$$

donde L representa la los niveles de escala de grises de los pixels. Por ejemplo, para datos del tipo uint8 ⇒ L=256. Para este caso, el PSNR se expresa como

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{255 \cdot 255}{\mathrm{MSE}^2}$$

PROCESAMIENTO DE IMAGEN CON MATLAB

➢ Lectura de Ficheros de Imagen

`imfinfo`: devuelve una estructura cuyos campos contienen información de un fichero de imagen.
`imread`: lee una imagen de un fichero y almacena los datos en un *array* numérico.

```
ImagFile1=imfinfo('peppers.png')
f=imread('coins.tif');
```

➢ Escritura de Ficheros de Imagen

`imwrite`: crea un fichero de imagen a partir de un array numérico. Admite diferentes formatos.

```
imwrite(f, 'peppers.png');
imwrite(f, 'peppers.png','Quality',75);
```

- **Conversión a Imagen Gris: `rgb2gray`**

```
f=imread(f, 'peppers.png');
f_gray=rgb2gray(f);
```

- **Conversión a Imagen Binaria: `im2bw`**

```
f_binary=im2bw(f);
imwrite(f_binary,''peppers_binary.tif');
```

- **Visualización de Imágenes: `imshow`**

```
imshow(f)
figure, imshow(f_gray)
figure, imshow(X,map)
figure, imshow(f_binary)
```

- **Redimensionamiento de Imágenes: `imresize`**

```
scale=4;
f_down=imresize(f,1/scale,'nearest');
f_up=imresize(f_down,scale,'bicubic');
```

- **Transformación de Intensidad Gamma: `imadjust`**

```
gamma=2;
f_adjusted=imadjust(f,[],[],gamma);
```

- **Filtrado de Imágenes: `imfilter`**

```
h_average=fspecial('average',31);
f_bl=imfilter(f,h_average,'replicate');
h_unsharp=fspecial('unsharp');
f_sh=imfilter(f,h_unsharp,'replicate');
```

- **Extracción de Bordes: `edge`**

```
f_edge=edge(rgb2gray(f),'canny');
```