



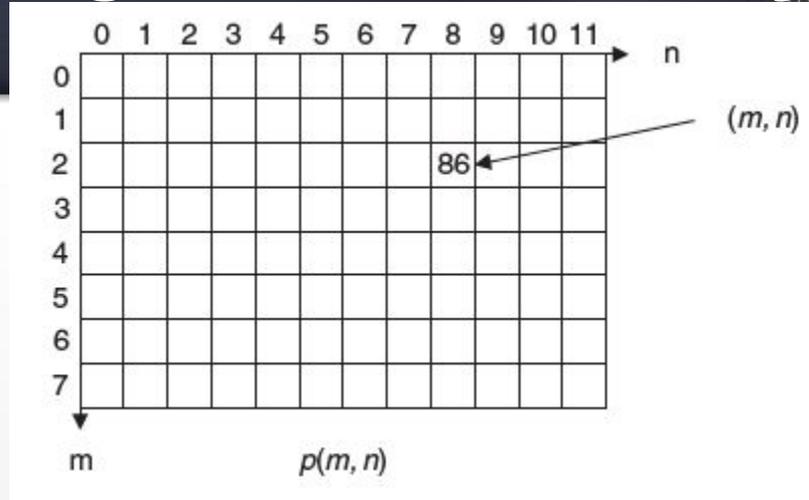
# Procesamiento Digital de Señales

## Unidad 7: Fundamentos del Procesamiento de Imágenes

# Procesamiento de Imágenes: Notación y formatos de datos

- Digitalización de imágenes
- La imagen digital está compuesta por píxeles
- La posición de cada pixel está especificada por un índice para el número de columna y otro para el número de fila (a modo de matriz).
- El valor del píxel indica un grado de intensidad en una imagen monocromática, por ejemplo:  $p(2,8) = 86$
- El número de píxeles en la imagen representan la resolución espacial, relacionado con la calidad de la imagen

# Procesamiento de Imágenes: Notación y formatos de datos



La resolución espacial puede ser tan alta como 1600x1200  
(=1.920.000 pixeles= 1,92 megapixeles)

O baja como 320x200 (64.000 pixeles = 64 kilo pixeles)

El número de la izquierda representa el ancho y el de la derecha el alto.

# Imágenes en escala de grises de 8 bits

- El valor del píxel es codificado en una escala de 0 a 255, donde 0 es negro y 255 es blanco

Ejemplo imagen en escala de grises  
8 bits, de 320 x 240 pixels  
(76,8 Kbytes)



# Imágenes a color en 24 bits

- A cada píxel se le asignan tres números, que corresponden a niveles de los componentes de los colores rojo, verde y azul (RGB) (fórmula aditiva = emisión de luz)
- A cada componente se le asignan 8 bits
- Se pueden lograr  $2^{24} = 16.777.216$  colores



1 bit 2 Color image



8 bits Grayscale image



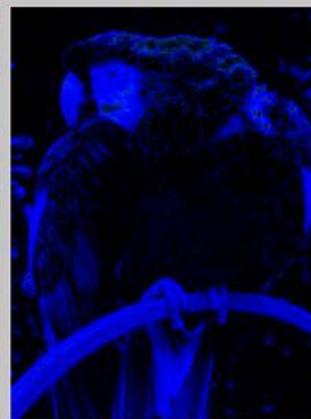
24 bits Truecolor image



8 bits Red Channel



8 bits Green Channel



8 bits Blue Channel



= 24 bits RGB Color

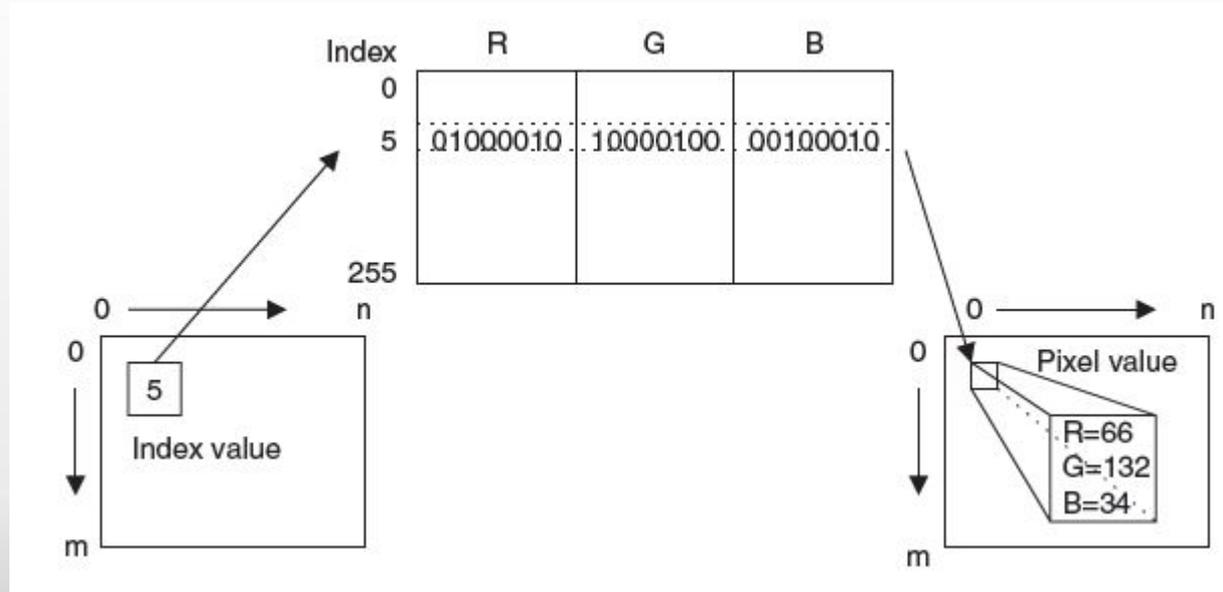
011000110

011001100

# Imágenes a color en 8 bits



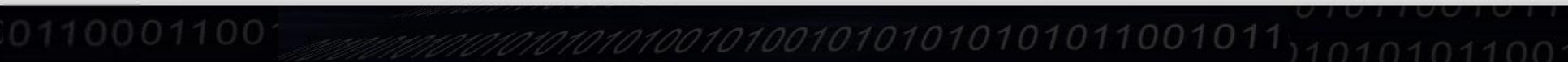
- En este formato se usa un esquema de indexación con una tabla de búsqueda para lograr representar colores



# Imágenes a color en x bits



- En este formato se usa un esquema de indexación con una tabla de búsqueda para lograr representar colores



# Imágenes de intensidad

- Se refiere a una normalización en imágenes en escala de grises, de forma tal que el rango de variación sea de 0 (negro) a 1 (blanco).
- Esto facilita la implementación de procesamiento digital, como filtrado (en Matlab)
- Se vuelve al formato estándar de 8 bits para la visualización

# Conversión RGB y Escala de Grises

- En algunas aplicaciones es necesario convertir de RGB a escala de grises.
- En realidad se transforma a formatos tridimensionales diferentes y de ellos se extrae solo la componente relacionada con la iluminancia.
- Por ejemplo: formato YIQ: Y, iluminancia; canales de crominancia: I, en fase, y Q, en cuadratura (esto último representa detalles del color)
- YIQ fue utilizado en la norma NTSC de televisión

# Conversión RGB y Escala de Grises



- Matrices de conversión:

$$\begin{bmatrix} Y(m, n) \\ I(m, n) \\ Q(m, n) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R(m, n) \\ G(m, n) \\ B(m, n) \end{bmatrix}$$

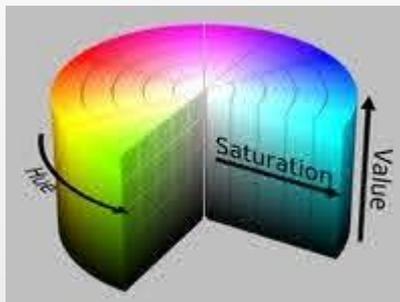
$$\begin{bmatrix} R(m, n) \\ G(m, n) \\ B(m, n) \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y(m, n) \\ I(m, n) \\ Q(m, n) \end{bmatrix}$$

La componente Y acarrea la información de escala de grises



# Otras formas de codificar colores

- HSV (hue, saturation, value), también conocido como HSB (hue, saturation, brightness) es utilizado por artistas debido a la forma más intuitiva de pensar los colores en términos de tonalidad y saturación



- Matlab: `HSV = rgb2hsv(RGB)`

HSV to RGB conversion formula

When  $0 \leq H < 360$ ,  $0 \leq S \leq 1$  and  $0 \leq V \leq 1$ :

$$C = V \times S$$

$$X = C \times (1 - |(H / 60^\circ) \bmod 2 - 1|)$$

$$m = V - C$$

$$(R', G', B') = \begin{cases} (C, X, 0) & , 0^\circ \leq H < 60^\circ \\ (X, C, 0) & , 60^\circ \leq H < 120^\circ \\ (0, C, X) & , 120^\circ \leq H < 180^\circ \\ (0, X, C) & , 180^\circ \leq H < 240^\circ \\ (X, 0, C) & , 240^\circ \leq H < 300^\circ \\ (C, 0, X) & , 300^\circ \leq H < 360^\circ \end{cases}$$

$$(R, G, B) = ((R'+m) \times 255, (G'+m) \times 255, (B'+m) \times 255)$$

# Otras formas de codificar colores

- HSL (**h**ue, **s**aturation, **l**ightness/luminance), también conocido como HLS or HSI (hue, saturation, intensity) es similar a HSV, con "lightness" reemplazando "brightness". La diferencia es que *brightness* de un color puro es igual a brightness del blanco, mientras que *lightness* de un color puro es igual a la lightness de una gris medio.

# Histograma y Ecualización



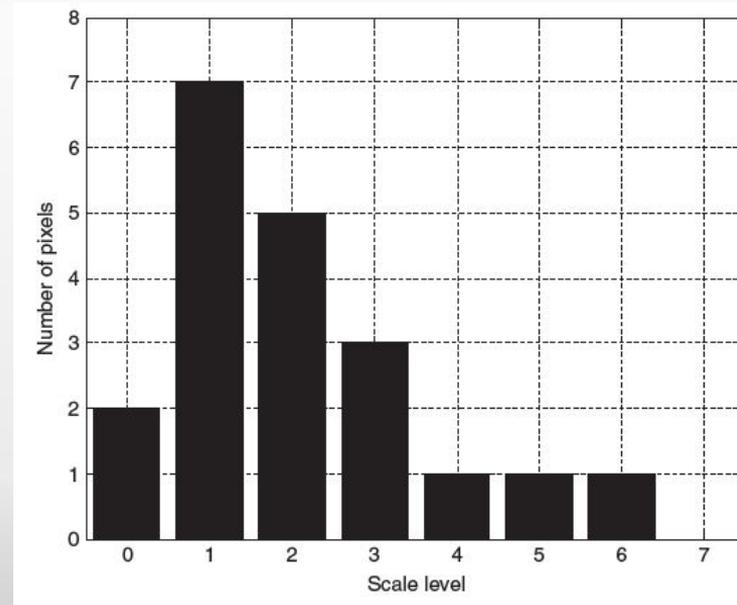
- El histograma es un gráfico que muestra la cantidad de pixeles en cada nivel de la escala de grises o cada índice de color.
- Ejemplo en escala de grises a 3 bits:

0	1	2	2	6
2	1	1	2	1
1	3	4	3	3
0	2	5	1	1



**Pixel counts distribution.**

Pixel $p(m, n)$ Level	Number of Pixels
0	2
1	7
2	5
3	3
4	1
5	1
6	1
7	0



# Ecualización

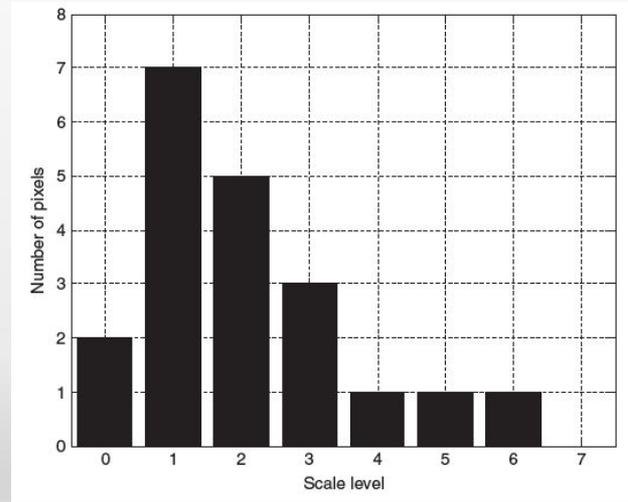
- La ecualización extiende el rango de escala de los pixeles al rango completo, mejorando el contraste

$$p_{eq}(m, n) = \frac{\text{Number of pixels with scale level} \leq p(m, n)}{\text{Total number of pixels}} \times (\text{maximum scale level})$$

- Ejemplo:

0	1	2	2	6
2	1	1	2	1
1	3	4	3	3
0	2	5	1	1

imagen



histograma

# Ecualización

Ejemplo:

$$p_{eq}(m, n) = \frac{\text{Number of pixels with scale level} \leq p(m, n)}{\text{Total number of pixels}} \times (\text{maximum scale level})$$

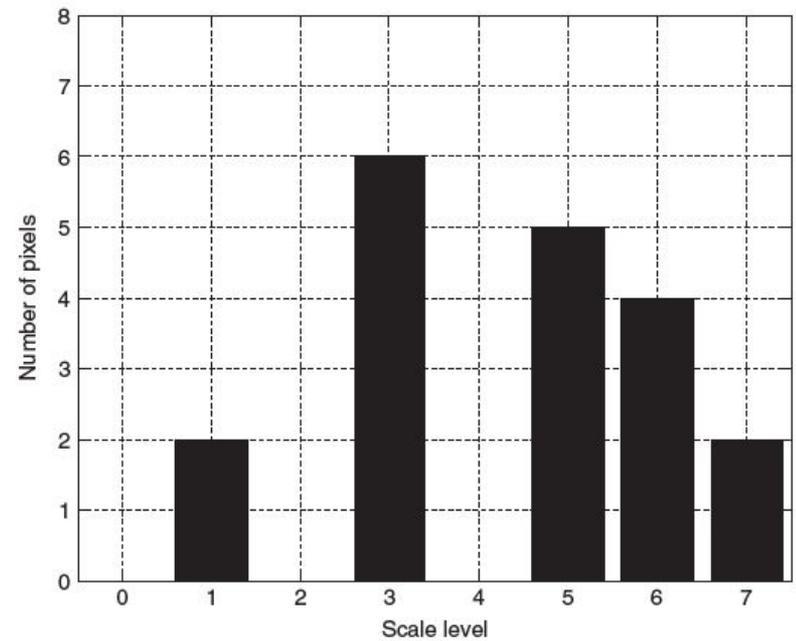
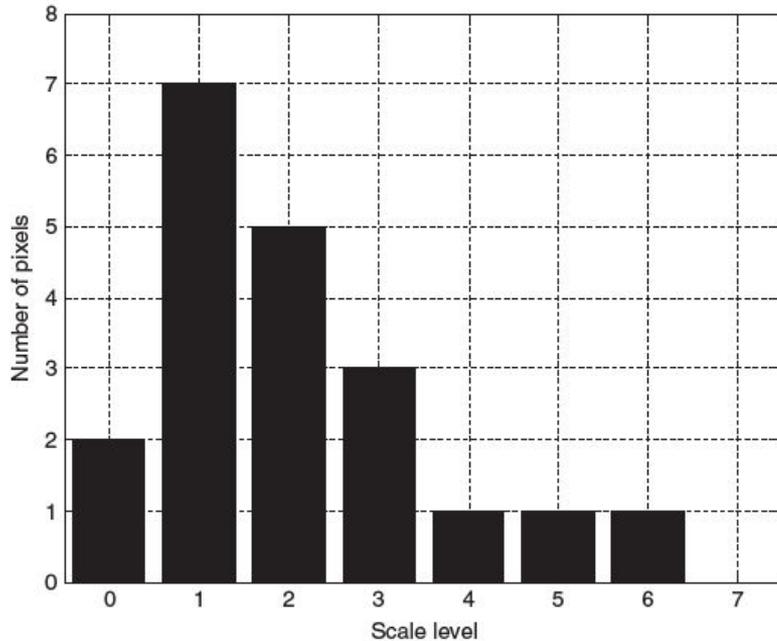
0	1	2	2	6
2	1	1	2	1
1	3	4	3	3
0	2	5	1	1

Image equalization			
Pixel $p(m, n)$		Number of Pixels	
Level	Number of Pixels	$\leq p(m, n)$	Equalized Pixel Level
0	2	2	1
1	7	9	3
2	5	14	5
3	3	17	6
4	1	18	6
5	1	19	7
6	1	20	7
7	0	20	7

# Ecuación

Ejemplo:

$$p_{eq}(m, n) = \frac{\text{Number of pixels with scale level} \leq p(m, n)}{\text{Total number of pixels}} \times (\text{maximum scale level})$$



# Ecualización

Ejemplo:

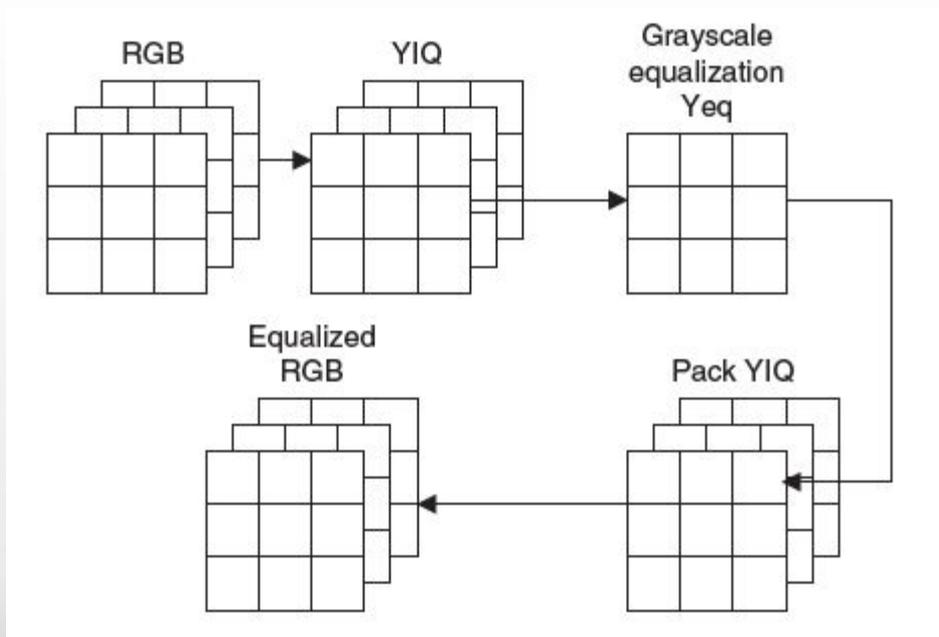
$$p_{eq}(m, n) = \frac{\text{Number of pixels with scale level} \leq p(m, n)}{\text{Total number of pixels}} \times (\text{maximum scale level})$$



# Ecualización de imágenes a color en 24 bits

- Para ecualizar imágenes RGB, primero se convierten a YIQ
- Se toma el canal Y (generalmente tiene alrededor del 93% de la energía)
- Luego, el canal Y es ecualizado de la misma forma que en escala de grises
- Se vuelven a componer YIQ (I y Q no se modifican)
- Se convierte YIQ a RGB

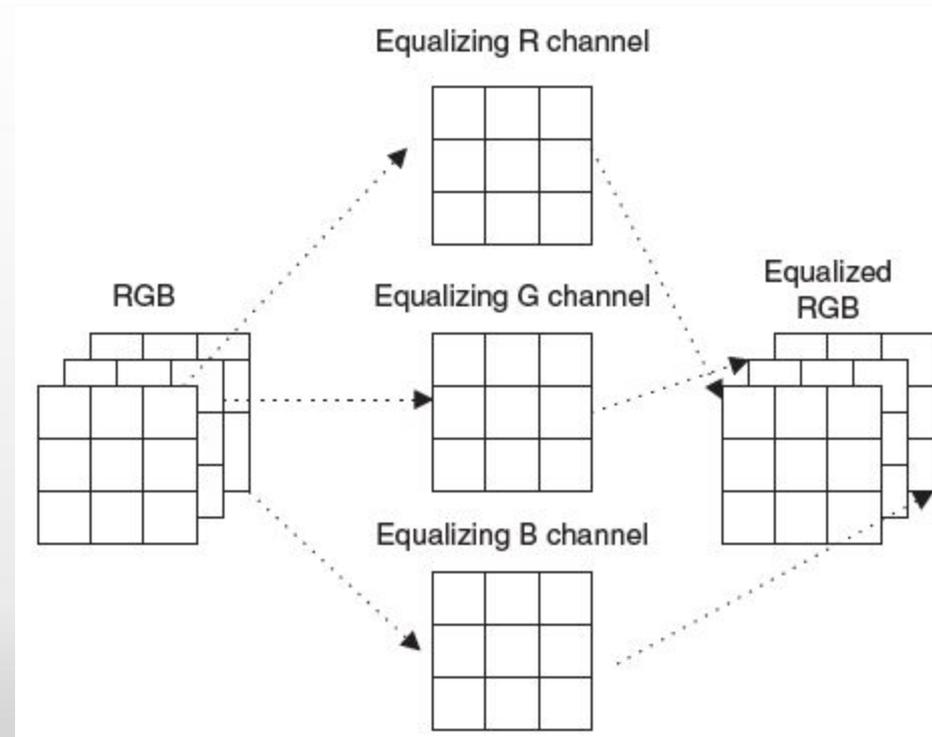
# Ecuación de imágenes a color en 24 bits



# Ecualización de imágenes a color en 24 bits

Es posible también ecualizar cada componente de color de forma separada.

Se logran algunos efectos, por ejemplo, si se ecualiza solo el R, la imagen lucirá más roja, porque este color extiende su rango de variación.

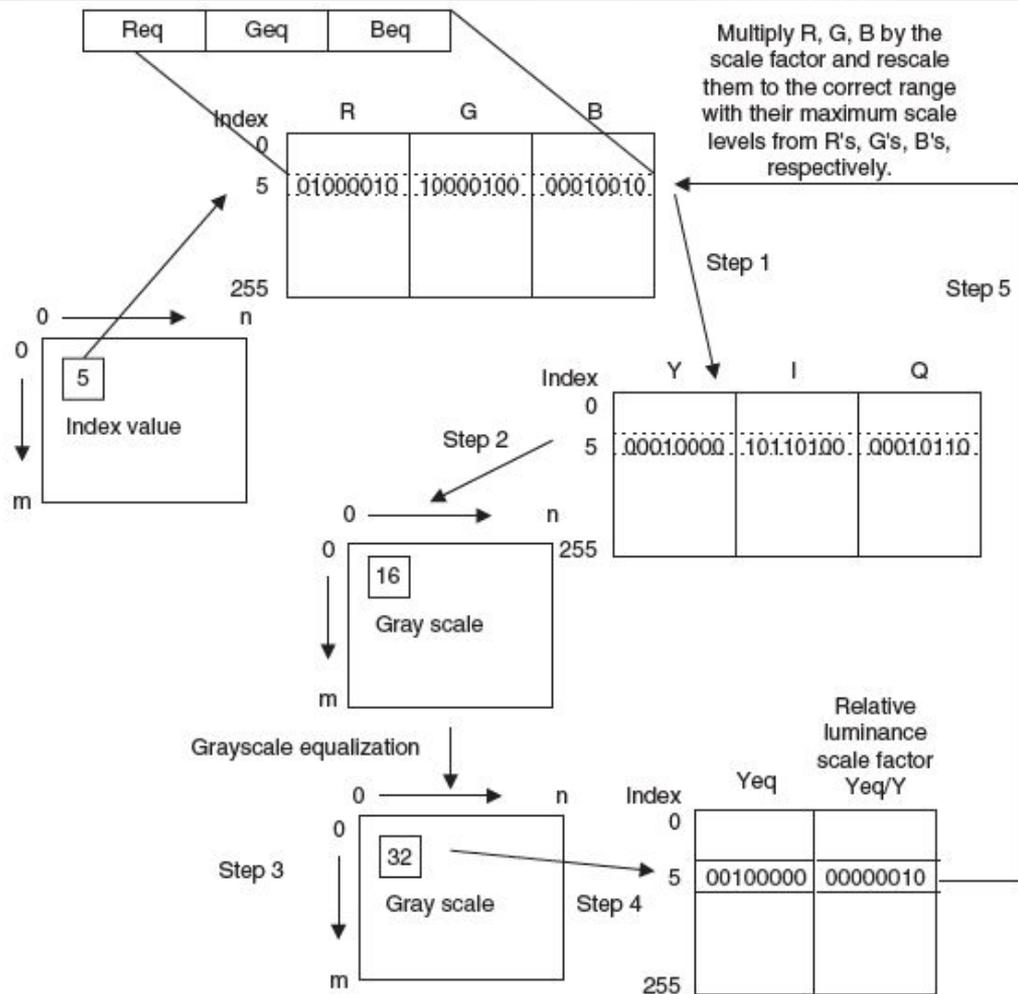


# Ecualización de imágenes a color codificadas en 8 bits

La ecualización en imágenes color indexadas en 8 bits es más complicada, debido a que el valor del pixel es el índice para ingresar al mapa de color, donde hay tres componentes (RGB) por cada uno

Por ello, se espera que después de la ecualización el índice no cambie, pero sí el color al cual está indicando

# Ecualización para imágenes a color codificadas en 8 bits:



# Ejemplo:

Utilizar el programa suministrado para analizar los efectos de la ecualización en la siguiente imagen:

**24-bit color**



# Ajustes de niveles y contrastes

- El ajuste de niveles puede ser utilizado para extender el nivel de pixel en una imagen para incrementar el contraste y desplazar el nivel del pixel para cambiar efectos de visión
- También es requerido para modificar resultados desde el filtrado de imágenes u otra operación para lograr un adecuado rango para la visualización

# Ajuste Lineal de Nivel de Pixel

- Corresponde a una fórmula lineal, donde el rango de variación inicial es mapeado a al rango completo de variación

$$p_{adjust}(m, n) = Bottom + \frac{p(m, n) - L}{H - L} \times (Top - Bottom),$$

where  $p(m, n)$  = original image pixel

$p_{adjust}(m, n)$  = desired image pixel

H = maximum pixel level in the original image

L = minimum pixel level in the original image

Top = maximum pixel level in the desired image

Bottom = minimum pixel level in the desired image

- El resultado es una mejora en el contraste (también es posible reducir el rango)

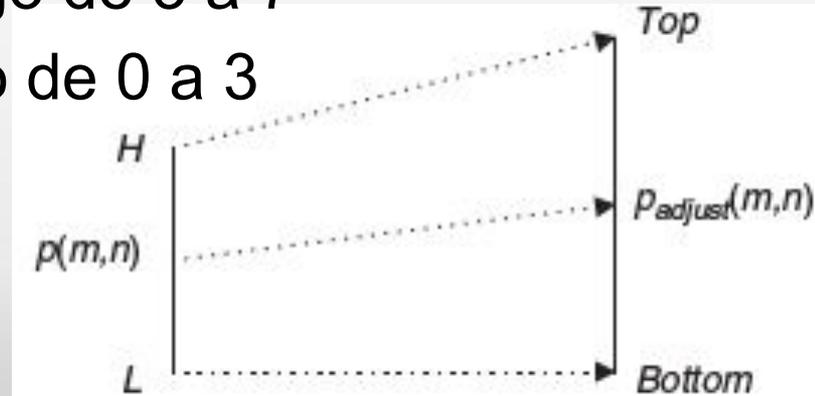
# Ajuste Lineal de Nivel de Pixel



- Ejemplo: imagen en 3 bits:

$$\begin{bmatrix} 3 & 4 & 4 & 5 \\ 5 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 \\ 3 & 5 & 3 & 4 \end{bmatrix},$$

- a) Realizar un ajuste de nivel a rango completo
- b) Desplazar el nivel superior del rango de 3 a 7
- c) Desplazar el nivel inferior del rango de 0 a 3



# Ajuste Lineal de Nivel de Pixel

- Ejemplo: imagen en 3 bits:

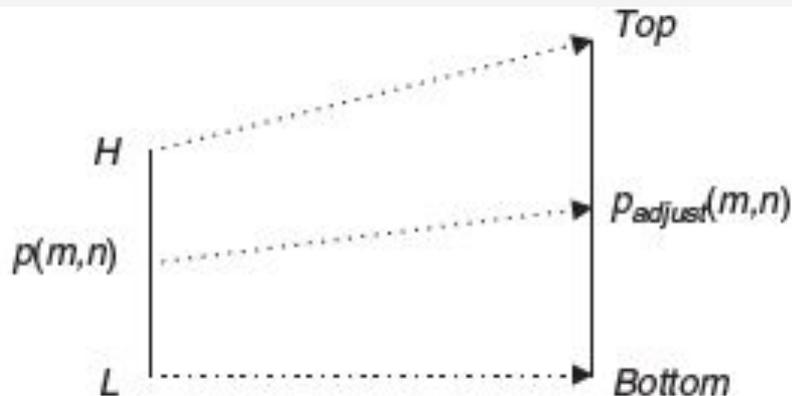
$$H = 5, L = 3, \text{Top} = 2^3 - 1 = 7, \text{Bottom} = 0.$$

$$\begin{bmatrix} 3 & 4 & 4 & 5 \\ 5 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 \\ 3 & 5 & 3 & 4 \end{bmatrix},$$

$$p_{\text{adjust}}(m, n) = \text{Bottom} + \frac{p(m, n) - L}{H - L} \times (\text{Top} - \text{Bottom}),$$

Pixel $p(m, n)$	Level	Full Range
3		0
4		4
5		7

$$\begin{bmatrix} 0 & 4 & 4 & 7 \\ 7 & 0 & 0 & 0 \\ 4 & 4 & 4 & 7 \\ 0 & 7 & 0 & 4 \end{bmatrix}$$



# Ajuste Lineal de Nivel de Pixel

- Ejemplo: imagen en 3 bits:

$$\begin{bmatrix} 3 & 4 & 4 & 5 \\ 5 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 \\ 3 & 5 & 3 & 4 \end{bmatrix},$$

$$p_{adjust}(m, n) = Bottom + \frac{p(m, n) - L}{H - L} \times (Top - Bottom),$$

$$\begin{bmatrix} 0 & 4 & 4 & 7 \\ 7 & 0 & 0 & 0 \\ 4 & 4 & 4 & 7 \\ 0 & 7 & 0 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 5 & 5 & 7 \\ 7 & 3 & 3 & 3 \\ 5 & 5 & 5 & 7 \\ 3 & 7 & 3 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 & 2 & 3 \\ 3 & 0 & 0 & 0 \\ 2 & 2 & 2 & 3 \\ 0 & 3 & 0 & 2 \end{bmatrix}$$

Pixel $p(m, n)$ Level	Full Range	Range [3-7]	Range [0-3]
3	0	3	0
4	4	5	2
5	7	7	3

# Ajuste Lineal de Nivel de Pixel

- Ejemplo:



Original



Stretching [0 0.5] to [0 1]



Shift-up



Shift-down

# Ajuste de Nivel para Visualización

- Ejemplo: dos imágenes de 8 bits se suman

Los valores de los pixeles resultantes pueden ser tan bajos como 0 o tan altos como 510.

Se puede utilizar un ajuste lineal de escala para volver al rango de 0 a 255

$$\begin{bmatrix} 30 & 25 & 5 & 170 \\ 70 & 210 & 250 & 30 \\ 225 & 125 & 50 & 70 \\ 28 & 100 & 30 & 50 \end{bmatrix} + \begin{bmatrix} 30 & 255 & 50 & 70 \\ 70 & 3 & 30 & 30 \\ 50 & 200 & 50 & 70 \\ 30 & 70 & 30 & 50 \end{bmatrix} = \begin{bmatrix} 60 & 280 & 55 & 240 \\ 140 & 213 & 280 & 60 \\ 275 & 325 & 100 & 140 \\ 58 & 179 & 60 & 100 \end{bmatrix}$$

# Ajuste de Nivel para Visualización

- Ejemplo: dos imágenes de 8 bits se suman:

$$p_{scaled}(m, n) = \frac{p(m, n) - \text{Minimum}}{\text{Maximum} - \text{Minimum}} \times (\text{Maximum scale level}).$$

Maximum = 325

Minimum = 55

Maximum scale level = 255,

60	280	55	240
140	213	280	60
275	325	100	140
58	179	60	100

5	213	0	175
80	149	213	5
208	255	43	80
3	109	5	43

# Mejora por filtrado de imagen



- Al igual que en el caso de imágenes unidimensionales, se pueden diseñar filtros pasa bajos, pasa altos, pasa banda y elimina banda para imágenes
- Se discuten los casos más comunes:
  - Filtro pasa bajos para eliminar ruido
  - Filtro de mediana para remover ruido impulsivo
  - Filtros de detección de borde para mejorar los límites de objetos



# Filtrado de ruido pasa bajos

- Se aplica de forma convolutiva un nucleo (kernel) constituido por bloques cuadrados de pixeles de diferentes tamaños

3 × 3 average kernel:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4 × 4 average kernel:

$$\frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

# Operación de filtrado por nucleo

Ejemplo: Realizar el filtrado digital de una imagen utilizando un nucleo convolucional de 2x2, y comparar el resultado con el original:

$$4 \times 4 \text{ original image: } \begin{bmatrix} 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \end{bmatrix}$$

$$4 \times 4 \text{ corrupted image: } \begin{bmatrix} 99 & 107 & 113 & 96 \\ 92 & 116 & 84 & 107 \\ 103 & 93 & 86 & 108 \\ 87 & 109 & 106 & 107 \end{bmatrix}$$

$$2 \times 2 \text{ average kernel: } \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

# Operación de filtrado por nucleo

99	107	113	96	0
92	116	84	107	0
103	93	86	108	0
87	109	106	107	0
0	0	0	0	0

To process the first element, we know that the first kernel covers the image elements as  $\begin{bmatrix} 99 & 107 \\ 92 & 116 \end{bmatrix}$ . Summing each product of the kernel element and the corresponding image pixel value, multiplying a scale factor of  $\frac{1}{4}$ , and rounding the result, it follows that

$$\frac{1}{4}(99 \times 1 + 107 \times 1 + 92 \times 1 + 116 \times 1) = 103.5$$
$$\text{round}(103.5) = 104.$$

In the processing of the second element, the kernel covers  $\begin{bmatrix} 107 & 113 \\ 116 & 84 \end{bmatrix}$ . Similarly, we have

$$\frac{1}{4}(107 \times 1 + 113 \times 1 + 116 \times 1 + 84 \times 1) = 105$$
$$\text{round}(105) = 105.$$

# Operación de filtrado por nucleo

99	107	113	96	0
92	116	84	107	0
103	93	86	108	0
87	109	106	107	0
0	0	0	0	0

The process continues for the rest of the image pixels. To process the last element of the first row, 96, since the kernel covers only  $\begin{bmatrix} 96 & 0 \\ 107 & 0 \end{bmatrix}$ , we assume that the last two elements are zeros. Then:

$$\frac{1}{4}(96 \times 1 + 107 \times 1 + 0 \times 1 + 0 \times 1) = 50.75$$

$$\text{round}(50.75) = 51.$$

Finally, we yield the following filtered image:

$$\begin{bmatrix} 104 & 105 & 100 & 51 \\ 101 & 95 & 96 & 54 \\ 98 & 98 & 102 & 54 \\ 49 & 54 & 53 & 27 \end{bmatrix}.$$

# Operación de filtrado por núcleo



Noisy image



3x3 kernel



4x4 kernel



8x8 kernel

# Filtrado por mediana

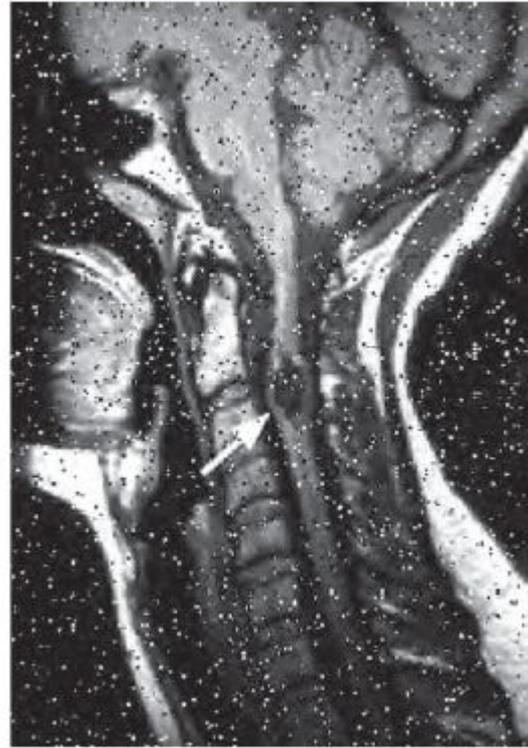
- Es un tipo de filtro no lineal muy efectivo para remover ruido impulsivo (ruido sal y pimienta) en la imagen
- Reemplaza el nivel de gris de cada pixel por la mediana del gris de los pixeles vecino

0	0	0	0	0	0
0	100	255	100	100	0
0	100	255	100	100	0
0	255	100	100	0	0
0	100	100	100	100	0
0	0	0	0	0	0

The values covered by the kernel are:

100, 100, 100, 100, 100, 100, 255, 255, 255.

# Filtrado por mediana



# Detección de Borde

En muchas aplicaciones, como en reconocimiento de patrones, identificaciones biométricas (como iris y huellas digitales), la información de borde es requerida

Se puede utilizar una convolución diferencial por núcleo. Un núcleo utilizado es el de Sobel:

Horizontal Sobel edge detector:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Vertical Sobel edge detector:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

# Detección de Borde

El nucleo Laplaciano es utilizado para mejorar tanto en la horizontal como vertical:

Laplacian edge detector:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

# Detección de Borde

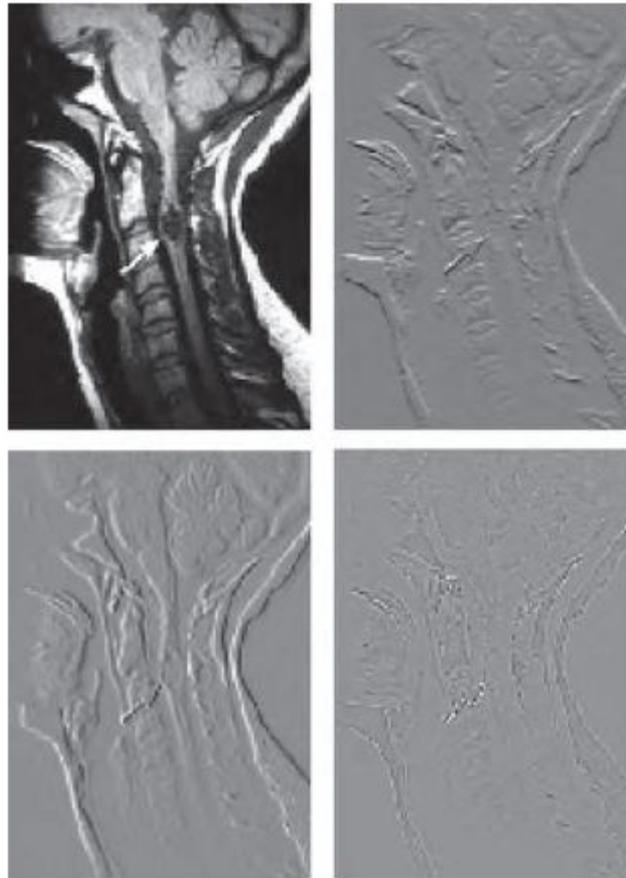
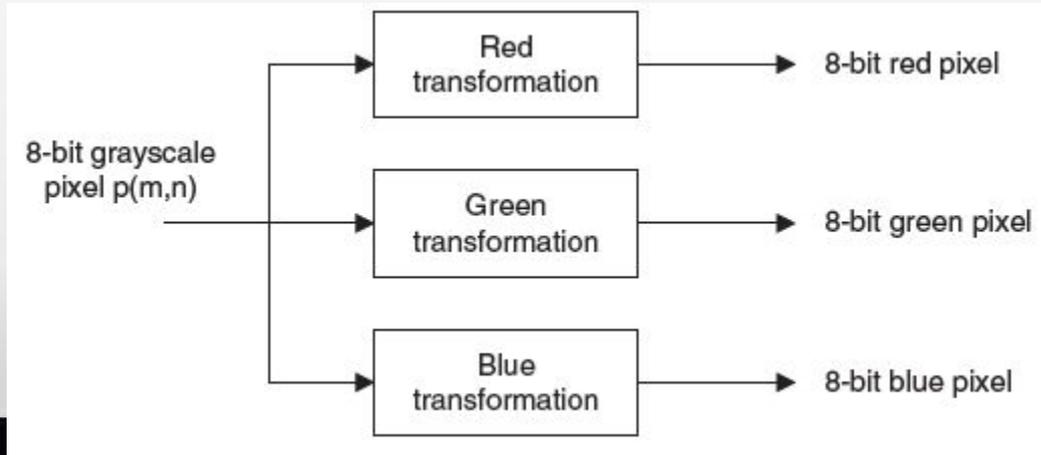


Image edge detection. (*Upper left*) Original image; (*upper right*) result from Sobel horizontal edge detector; (*lower left*) result from Sobel vertical edge detector; (*lower right*) result from Laplacian edge detector.

# Generación y Detección de Pseudo Color

- Este tema trata de la aplicación de transformaciones a una imagen en escala de grises para transformarla en una imagen a color
- En detección de objetos, la generación de pseudo color resalta lo que se quiere detectar

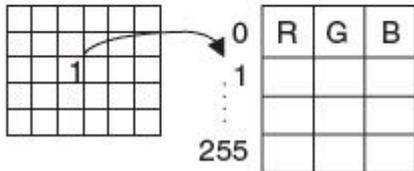


# Generación y Detección de Pseudo Color

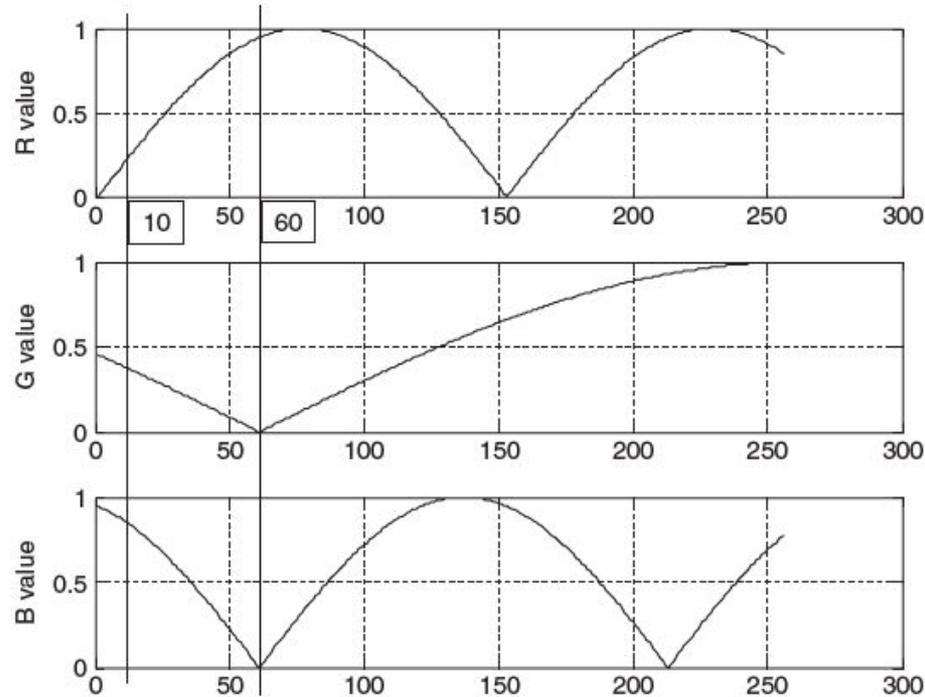
Ejemplo de regla de transformación: senoides desfasadas

Indexed color image

- 1) pixel value is an index; there are 256 indices
- 2) each index points to a color map entry with three values: Red, Green and Blue

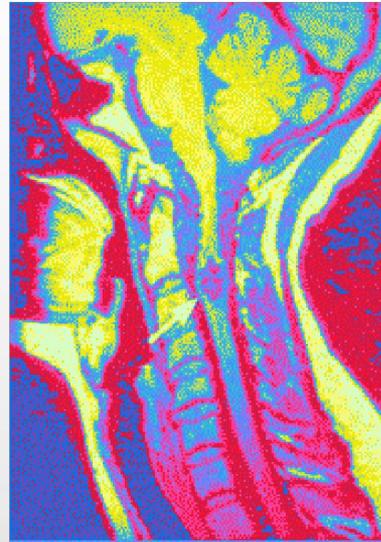
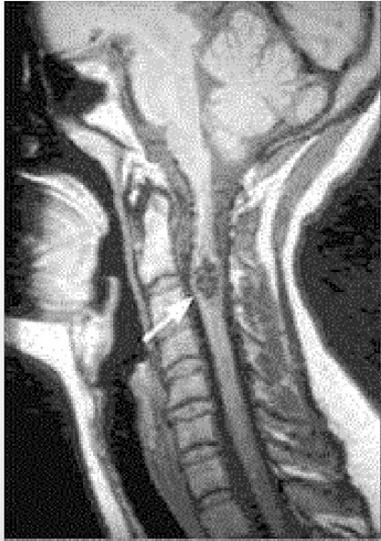


```
[X, map]=imread('file name','format');
```



# Generación y Detección de Pseudo Color

## Ejemplo



# Espectro de Imágenes

- DFT de 2 dimensiones

$$X(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} p(m,n) W_M^{um} W_N^{vn},$$

where  $W_M = e^{-j\frac{2\pi}{M}}$  and  $W_N = e^{-j\frac{2\pi}{N}}$ ,  
 $m$  and  $n$  = pixel locations  
 $u$  and  $v$  = frequency indices.

- a. Determine the 2D-DFT coefficients and magnitude spectrum for the following  $2 \times 2$  image:

$$\begin{bmatrix} 100 & 50 \\ 100 & -10 \end{bmatrix}.$$

**Solution:**

- a. Since  $M = N = 2$ , applying Equation (13.16) leads to

$$X(u, v) = p(0, 0)e^{-j\frac{2\pi u \times 0}{2}} \times e^{-j\frac{2\pi v \times 0}{2}} + p(0, 1)e^{-j\frac{2\pi u \times 0}{2}} \times e^{-j\frac{2\pi v \times 1}{2}} \\ + p(1, 0)e^{-j\frac{2\pi u \times 1}{2}} \times e^{-j\frac{2\pi v \times 0}{2}} + p(1, 1)e^{-j\frac{2\pi u \times 1}{2}} \times e^{-j\frac{2\pi v \times 1}{2}}.$$

Thus, we have DFT coefficients as

$$X(u, v) = \begin{bmatrix} 240 & 160 \\ 60 & -60 \end{bmatrix}.$$

For  $u = 0$  and  $v = 0$ , we have

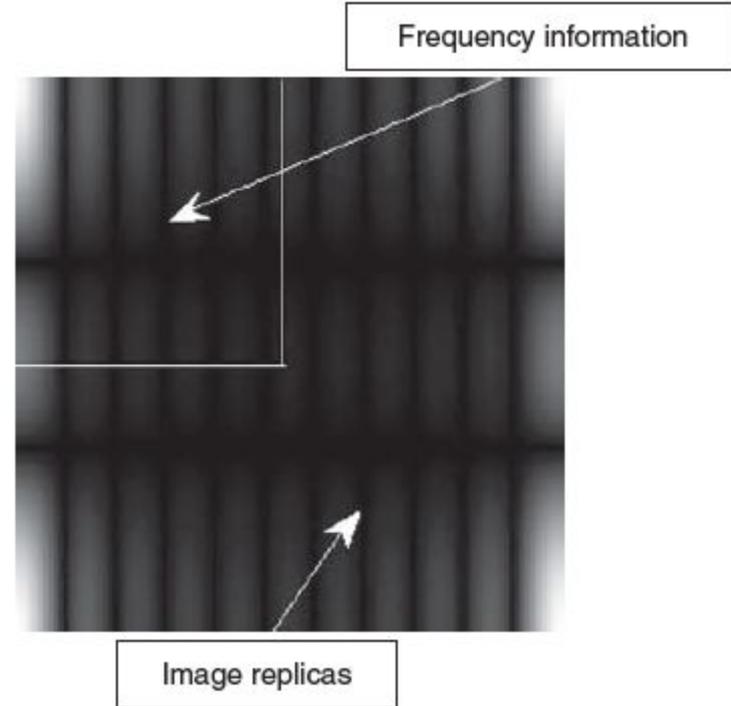
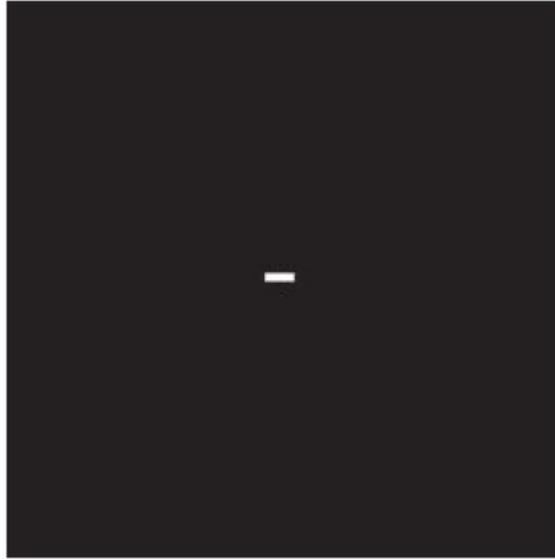
$$X(0, 0) = 100e^{-j0} \times e^{-j0} + 50e^{-j0} \times e^{-j0} + 100e^{-j0} \times e^{-j0} - 10e^{-j0} \times e^{-j0} \\ = 100 + 50 + 100 - 10 = 240$$

For  $u = 0$  and  $v = 1$ , we have

$$X(0, 1) = 100e^{-j0} \times e^{-j0} + 50e^{-j0} \times e^{-j\pi} + 100e^{-j0} \times e^{-j0} - 10e^{-j0} \times e^{-j\pi} \\ = 100 + 50 \times (-1) + 100 - 10 \times (-1) = 160$$

# Espectro de Imágenes

- Ejemplo:



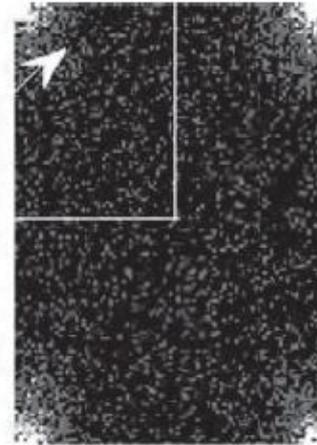
El espectro de magnitud es escalado de 0 a 255 para ser representado en escala de grises.

# Ejemplo:



A

Frequency  
content



B



C

Frequency  
content



D

Comments: It is applied a  
Gaussian low-pass filter