

Visió per Computador processat d'imatges

Felipe Lumbreras

Dept. Ciències de la Computació / Centre de Visió per Computador

Universitat Autònoma de Barcelona

<http://www.cvc.uab.es/shared/teach/a102784/>



Universitat Autònoma
de Barcelona



Conceptos fundamentales

Imágenes digitales

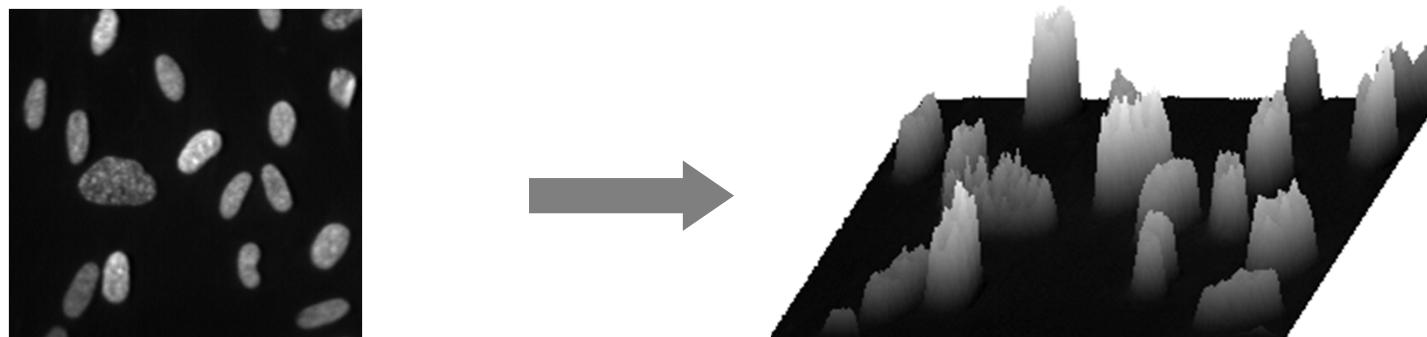
Muestreo y cuantización

Tipos de imágenes

Mejora de imágenes (operaciones básicas)

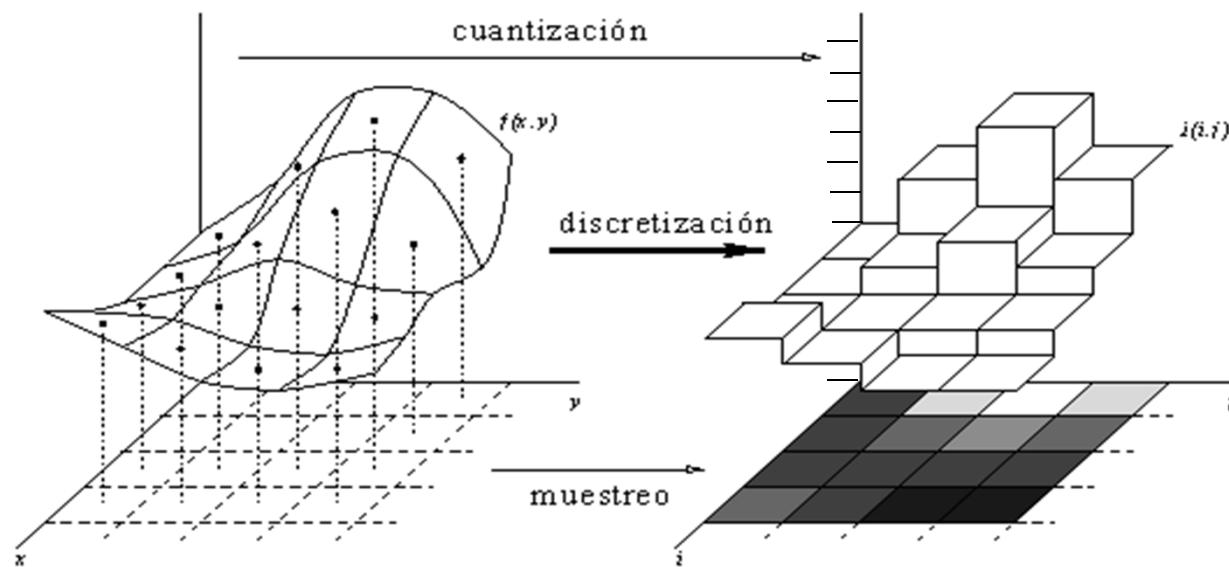
Representación de imágenes digitales

- Una imagen es una función bidimensional $I(x,y)$ que a cada punto (x,y) le hace corresponder el valor de la intensidad lumínica en ese punto.
- En imágenes monocromáticas el valor de luminancia $I(x,y)$ recibe el nombre de nivel de gris.
- Una imagen es una estructura de datos (**píxeles**) que contiene información discreta. Esta discretización la realiza el sensor en dos sentidos: muestreo y cuantización.

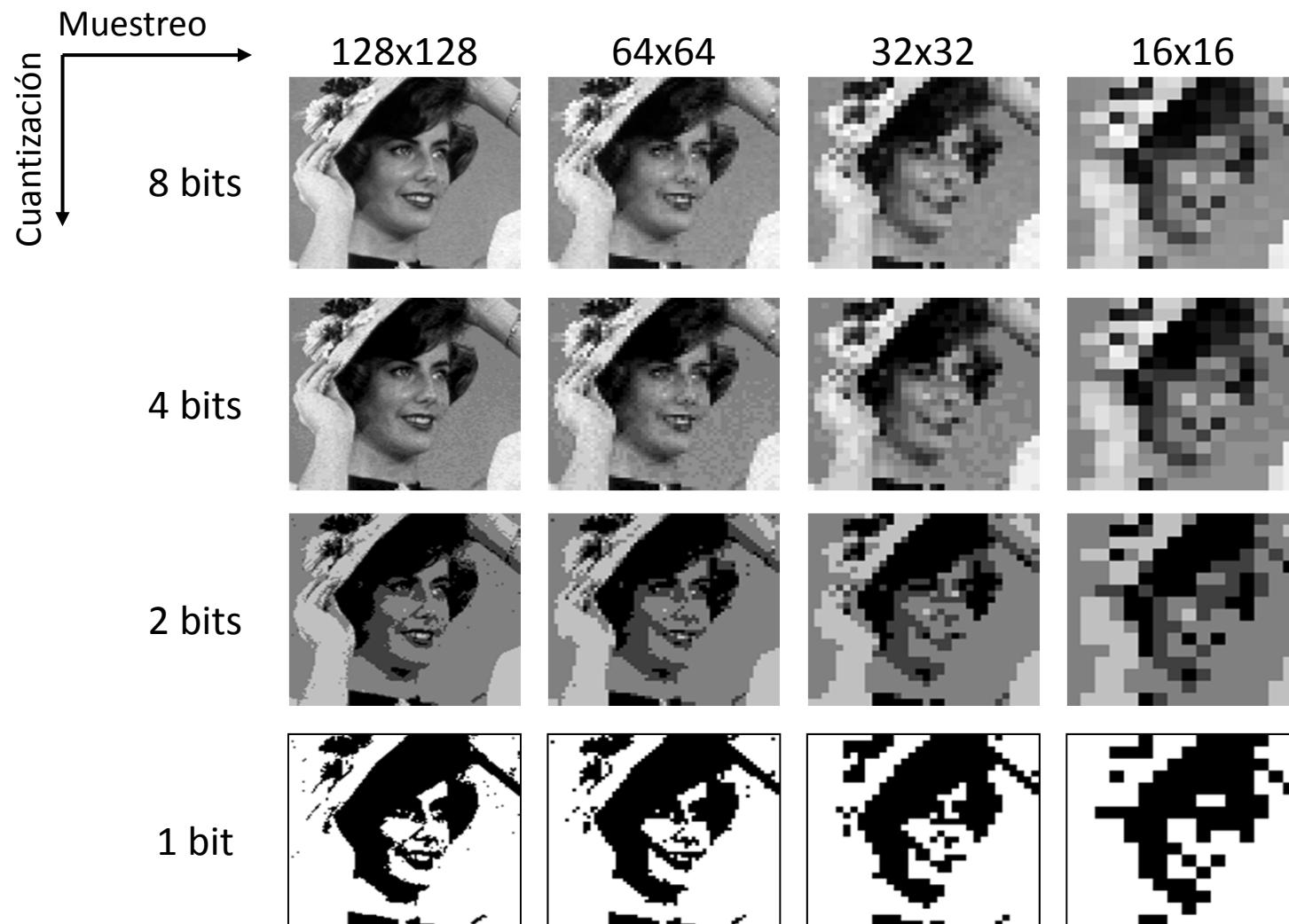


Muestreo y cuantización

- **Muestreo:** Discretización en el espacio. Generado al captar una señal continua del mundo real con un sensor discreto. Obtiene una representación computacional en forma de imagen.
- **Cuantización:** Discretización en amplitud. Generada al almacenar la señal en un formato digital (8 bits, 16 bits, ...)



Muestreo y cuantización. Ejemplo

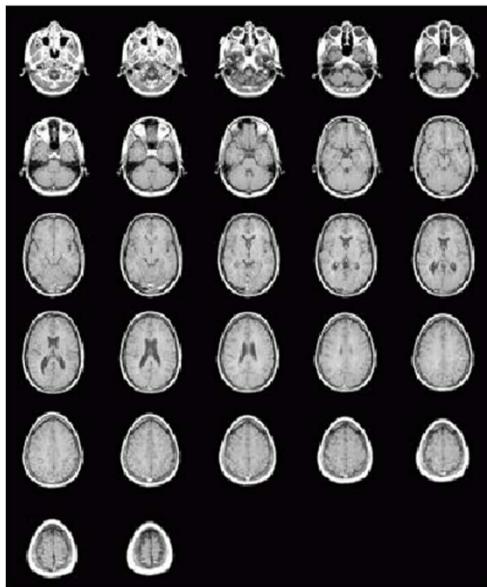


Tipos de imágenes

- **Imagen binaria:** dos valores, blanco y negro.
- **Imagen monocromática:** un único componente espectral, matizado por distintos componentes de intensidad
- **Imagen color:** tres componentes cromáticas: rojo, verde, azul
- **Secuencia de imágenes:** cada imagen representa la escena en un instante de tiempo $I(x,y,t)$
- **Imagen 3D:** imágenes apiladas en z (secciones de un objeto)
Imagen range: El valor de un píxel indica profundidad (envolvente 3D).
- **Imagen estéreo:** dos imágenes de la misma escena en el mismo instante de tiempo desde dos puntos de vista diferentes.

Tipos de imágenes

of the same di
There is some
perimental eviden
on the curves
metrically as po
what more arbit



Las imágenes en MATLAB

- Lo más habitual es que las imágenes de intensidad se representen como una matriz 2D de valores **uint8** (enteros de 0 a 255).
- Pero para operar con ellas muchas veces tendremos que pasarlas a valores **double** (valores reales de 0 a 1).
- Para hacer esto tenemos las funciones:

```
>> imA = im2uint8(im);  
>> imB = im2double(im);
```

- También podemos hacer cast a uint8 o double, si controlamos los rangos.

```
>> imA = uint8(im);  
>> imB = double(im);
```

Cargar y mostrar una imagen

- Para cargar una imagen podemos usar la función **imread**:

```
>> im = imread('pout.tif');
```

- Y para mostrarla usaremos la función **imshow**:

```
>> imshow(im);
```

- Si el resultado tiene poco contraste, podemos indicarle que nos muestre los valores adaptados al mínimo y máximo de la imagen:

```
>> imshow(im, []);
```

Ejemplos de imágenes en MATLAB

- Imagen binaria: 'text.png'
- Imagen en grises (de intensidad): 'cameraman.tif'
- Imagen a color: 'peppers.png'
- Imagen 3D: load mri

Mejora de imágenes

Introducción

Histograma

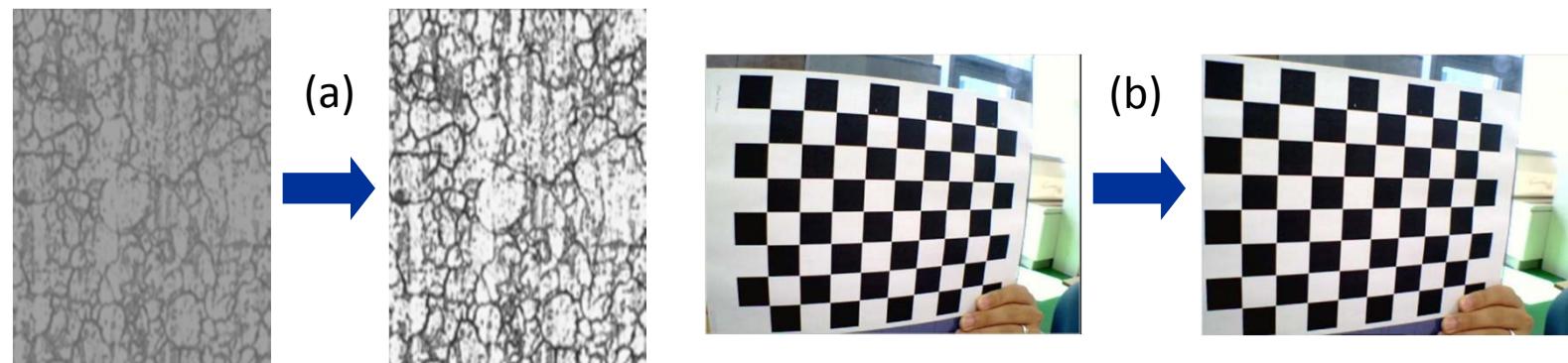
Transformaciones puntuales

Transformaciones aritméticas

Introducción

- **Mejora:** corrección de efectos producidos por el proceso de adquisición o de transmisión de la imagen.
- Conjunto de técnicas cuyo objetivo es procesar una imagen para obtener otra mejor según algún propósito (mejor visualización, eliminación de ruido, etc.). Dos tipos:
 - Correcciones fotométricas (a)
 - Correcciones geométricas (b)

Ejemplos:



Histograma

- Antes de abordar las operaciones con imágenes, veremos una herramienta muy útil para el estudio de éstas.

Histograma: vector de longitud igual al número de niveles de intensidad. La posición i -ésima, $H[i]$, de este vector es igual al número de píxeles de la imagen con nivel de intensidad i .

Histograma acumulado: la posición i -ésima refleja el número de píxeles con nivel de gris menor o igual a i .

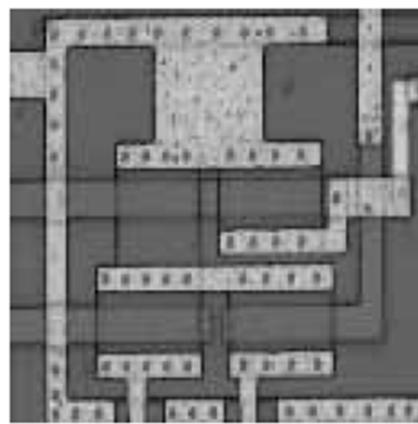
- Podemos ver el histograma como una densidad de probabilidad y el histograma acumulado como su función de distribución asociada.

Cálculo del
histograma

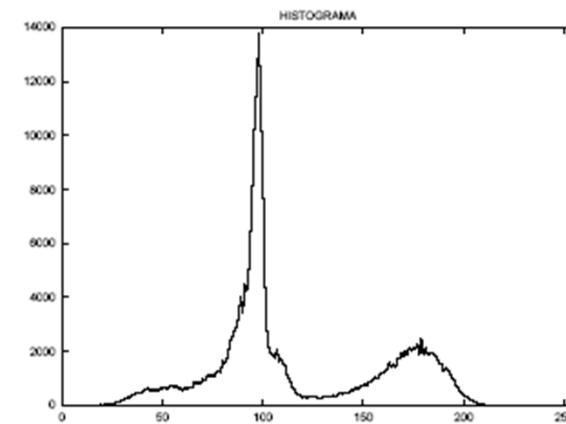
```
Inicializar todos los valores  $H[i]=0$ 
para todos los píxeles de la imagen  $(i,j)$ 
   $int=\text{pixel}(i,j)$ 
   $H[int]=H[int]+1$ 
fin para
```

Histograma. Ejemplo

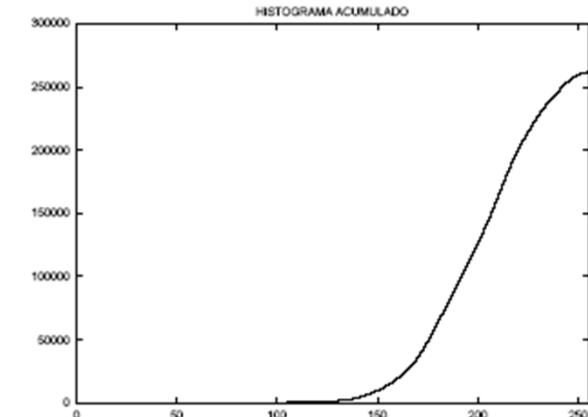
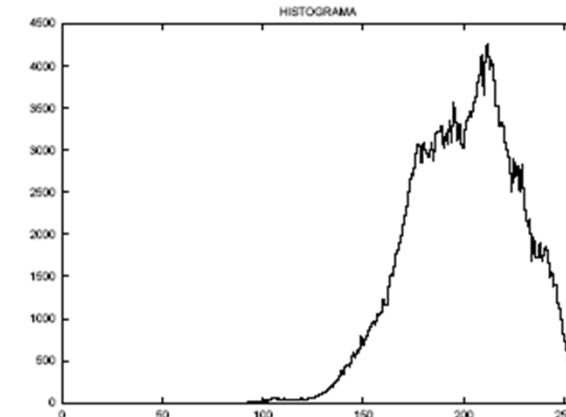
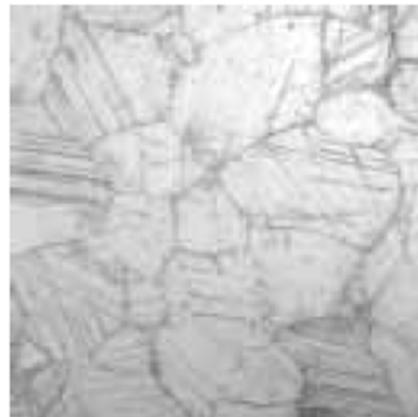
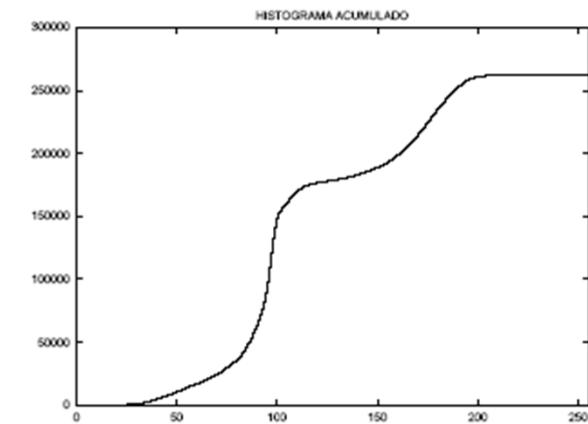
Imagen original



Histograma



Histograma
acumulado



Cálculo de Histograma

- Si queremos ver el histograma de una imagen, podemos usar **imhist**:

```
>> imhist(im)
```

- También podemos guardarla en una variable :

```
>> h = imhist(im);
```

```
>> size(h)
```

```
ans = 256 1
```

- Esto nos permite manipularlo. Por ejemplo, podemos calcular el histograma acumulado, o visualizarlo en otro formato:

```
>> hA = cumsum(h);
```

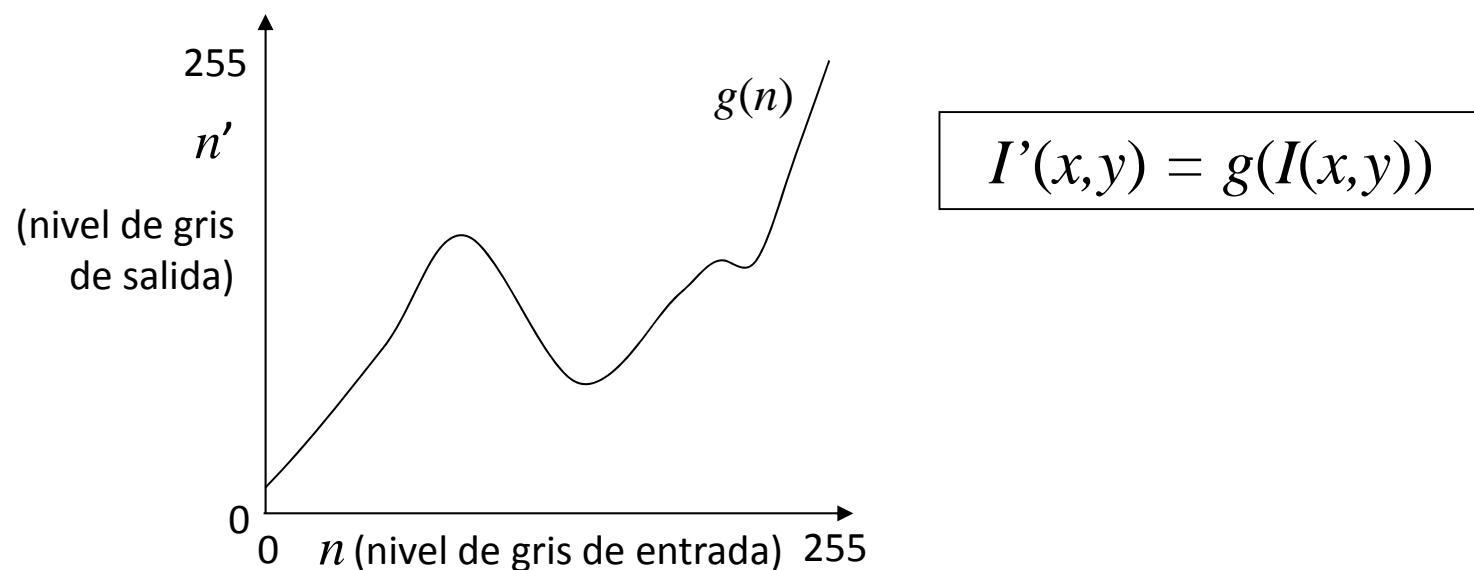
```
>> plot(h); xlim([0 255]);
```

Transformaciones puntuales

- Transformaciones puntuales simples
 - Umbral simple y doble
 - LUT
 - Transformaciones lineales
 - Imagen negativo
 - Maximización del contraste
 - Transformaciones no lineales
 - Logaritmo
 - Exponencial
 - Operaciones lógicas
- Ecualización

Transformaciones puntuales simples

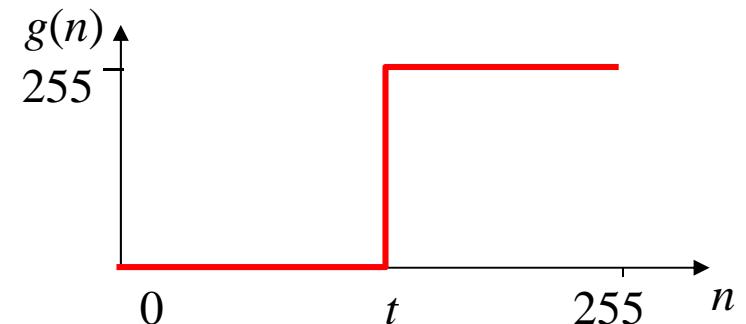
- Trabajan con los píxeles como elementos individuales. Se aplica una función que depende sólo del valor del píxel.
- Denotando por $I(x,y)$ el valor de la imagen I en el píxel (x,y) , una transformación puntual $g(n)$ transformar la imagen en función de los niveles de gris n de los píxeles de ésta.



Umbralización (threshold, binarización)

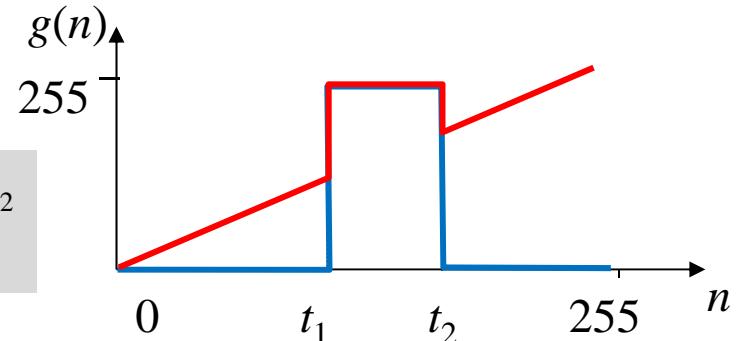
- **Umbral simple:** Los puntos con un nivel de intensidad mayor o igual que un cierto umbral t reciben el nivel máximo y el resto el nivel mínimo.

$$g(n) = \begin{cases} 255 & \text{si } I(x, y) \geq t \\ 0 & \text{sino} \end{cases}$$



- **Umbral doble:** Se elige un intervalo $[t_1, t_2]$ que se lleva al máximo. Una posible variación para mostrar el resultado sobre la imagen original es:

$$g(n) = \begin{cases} 255 & \text{si } t_1 \leq I(x, y) \leq t_2 \\ I(x, y) & \text{sino} \end{cases}$$



Umbral simple y doble. Ejemplos

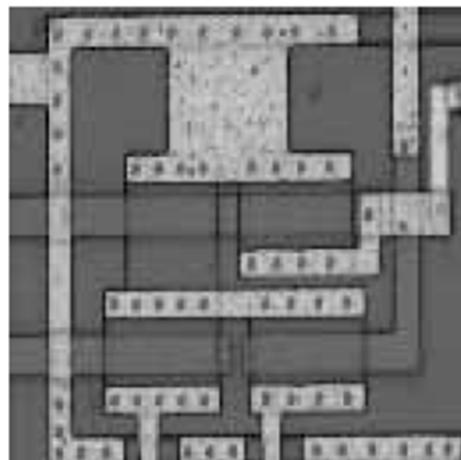


Imagen
original



Umbral
simple
 $(t = 180)$



Umbral doble
 $(t_1 = 100,$
 $t_2 = 150)$

Umbral simple y doble

- Mediante umbrales, vamos a seleccionar diferentes regiones de la imagen:

```
>> im = imread('coins.png');  
>> subplot(2,3,1); imshow(im);  
>> subplot(2,1,2); imhist(im);
```

- **Ejercicio 1:** Seleccionar el valor de t_0 para detectar el fondo:

```
>> imFondo = (im <= t0);  
>> subplot(2,3,2); imshow(imFondo);
```

- **Ejercicio 2:** Seleccionar el intervalo $[t_1, t_2]$ para detectar la moneda más oscura (aunque queden restos de las otras):

```
>> imMonedaOscura = (im>=t1) & (im<=t2);  
>> subplot(2,3,3); imshow(imMonedaOscura);
```

Transformaciones lineales

- Transformación de una imagen I con un rango de valores determinado (G_{\min}, G_{\max}) en una nueva imagen $I' = g(I)$, con rango (R_{\min}, R_{\max}), siendo g una función lineal.

$$\begin{aligned} g(I(x, y)) &= \frac{R_{\max} - R_{\min}}{G_{\max} - G_{\min}}(I(x, y) - G_{\min}) + R_{\min} = \\ &= \frac{R_{\max} - R_{\min}}{G_{\max} - G_{\min}} I(x, y) - \left[\frac{R_{\max} - R_{\min}}{G_{\max} - G_{\min}} G_{\min} - R_{\min} \right] \end{aligned}$$

Pendiente (m), si:

$m < 1$: disminución de contraste

$m > 1$: aumento del contraste

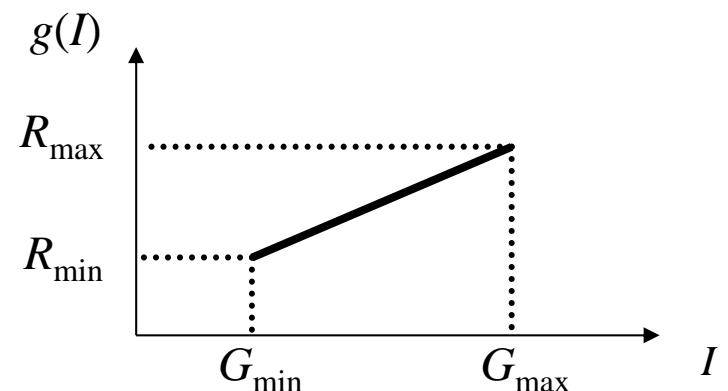


Imagen negativo

- Contraste inverso

$$g(n) = 255 - I(x, y)$$

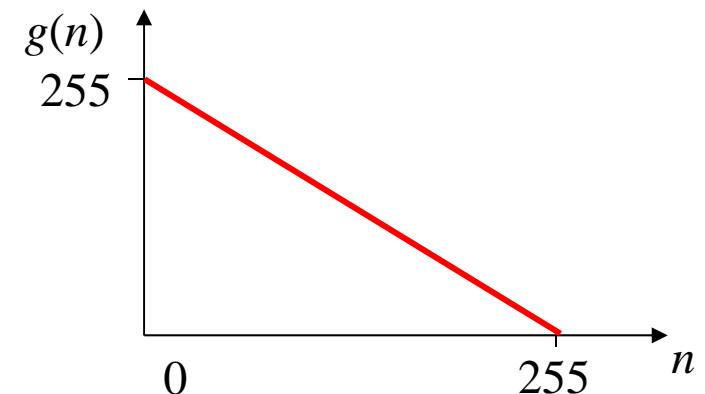


Imagen original

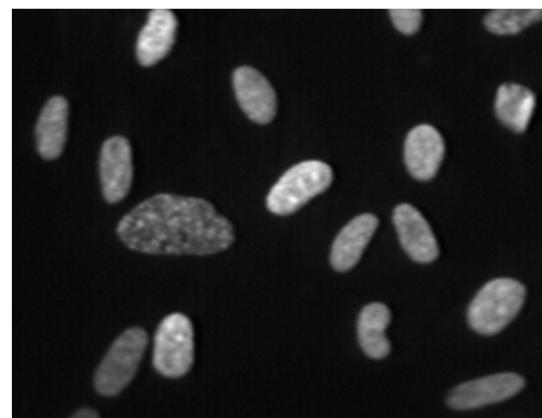
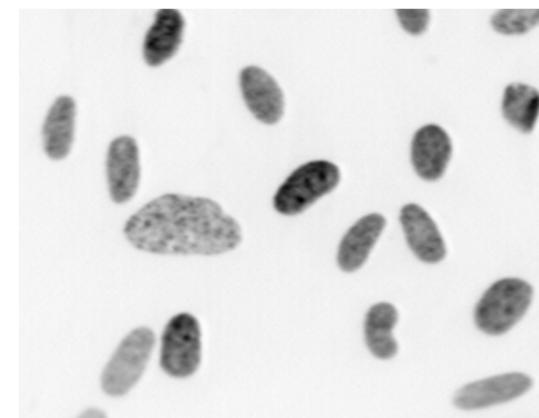


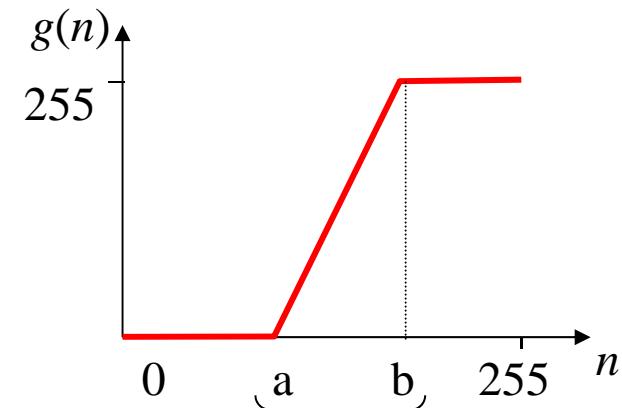
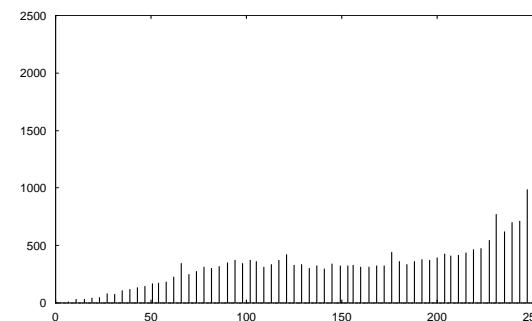
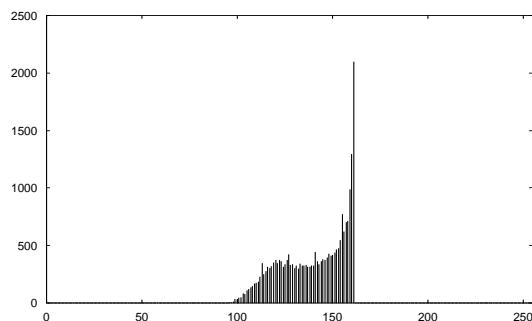
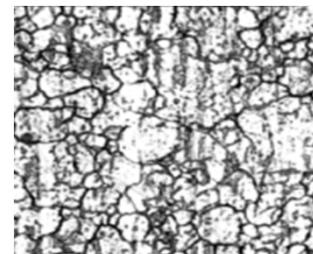
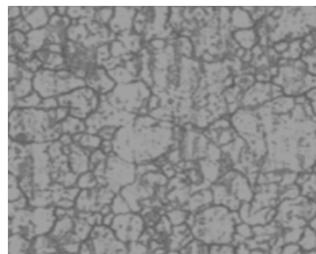
Imagen negativo



Maximización del contraste

- Consiste en aplicar una transformación lineal que haga que la imagen ocupe todo el rango posible. Se produce un “estiramiento” del histograma.

$$g(n) = 255 \left[\frac{I(x, y) - \min(I(x, y))}{\max(I(x, y)) - \min(I(x, y))} \right]$$



Rango de grises de la
imagen original

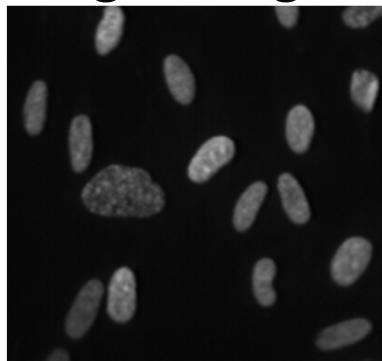
Transformaciones no lineales

- Las transformaciones lineales modifican el rango de I . En cambio, las transformaciones no lineales modifican el contraste de una imagen dentro del rango de I .
- Dos tipos de transformaciones:
 - **Logarítmicas:** realce de píxeles oscuros sin saturar píxeles brillantes.
$$g(I(x, y)) = \log(I(x, y) + 1)$$
 - **Exponenciales:** realce de píxeles brillantes sin saturar píxeles oscuros.
$$g(I(x, y)) = e^{I(x, y)}$$

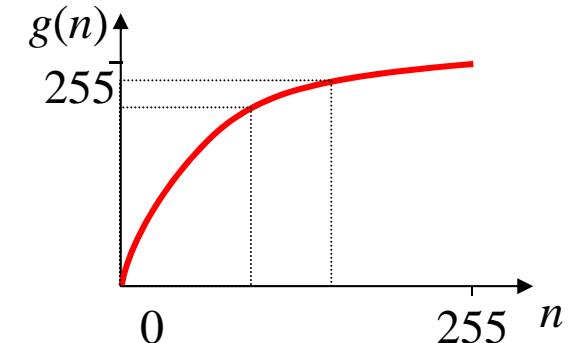
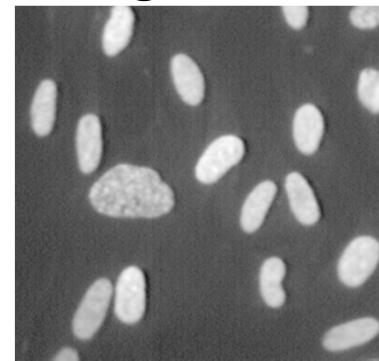
Logaritmo

$$g(n) = \log(I(x, y) + 1)$$

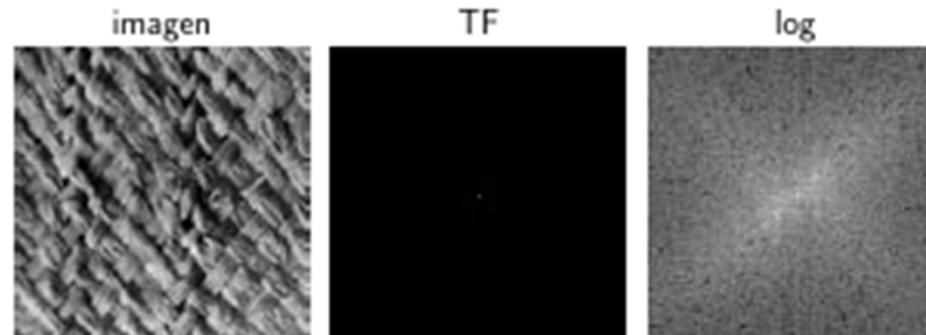
Imagen original



Logaritmo

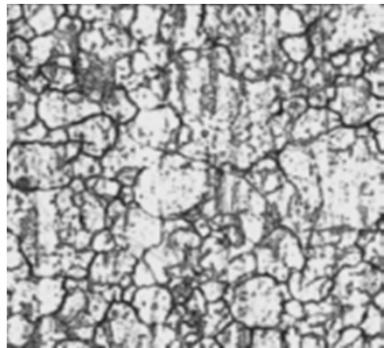


Útil para visualizar imágenes con rango dinámico amplio
(p. ej. transformada de Fourier).



Exponencial

Imagen original



Exponencial

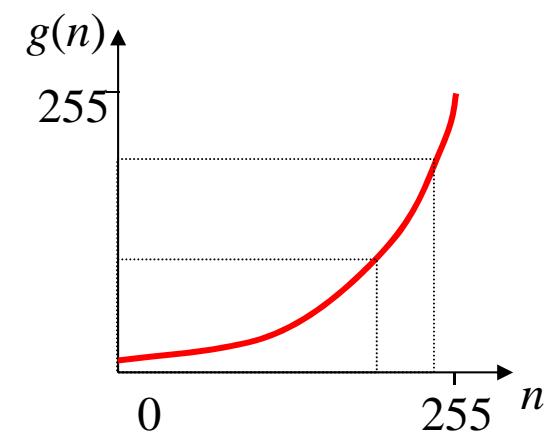
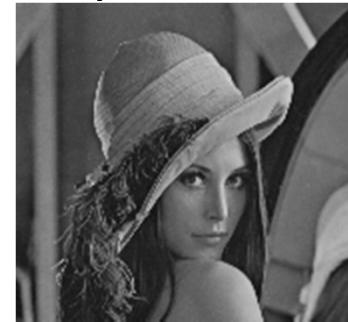


$$g(n) = e^{I(x,y)}$$

Imagen original

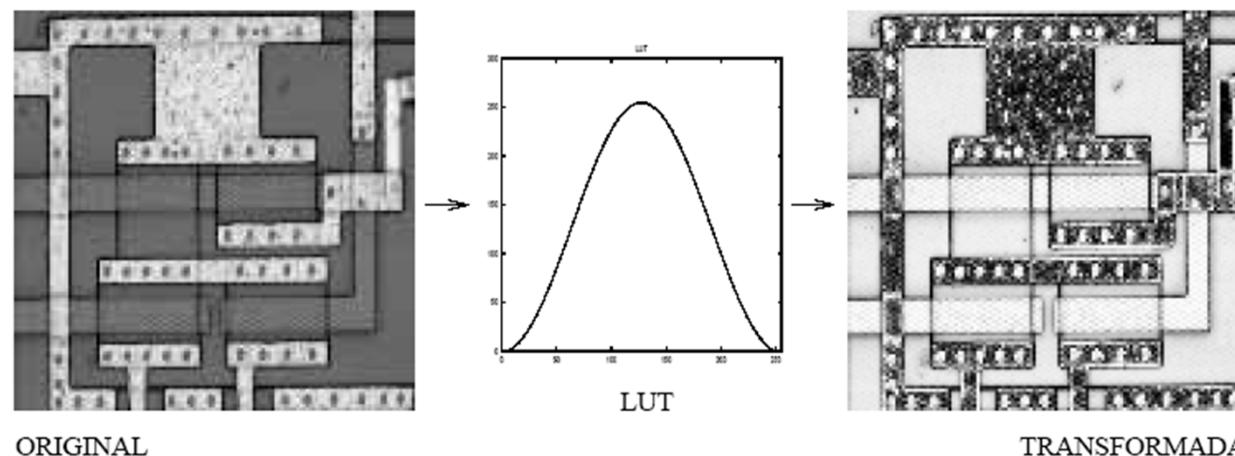
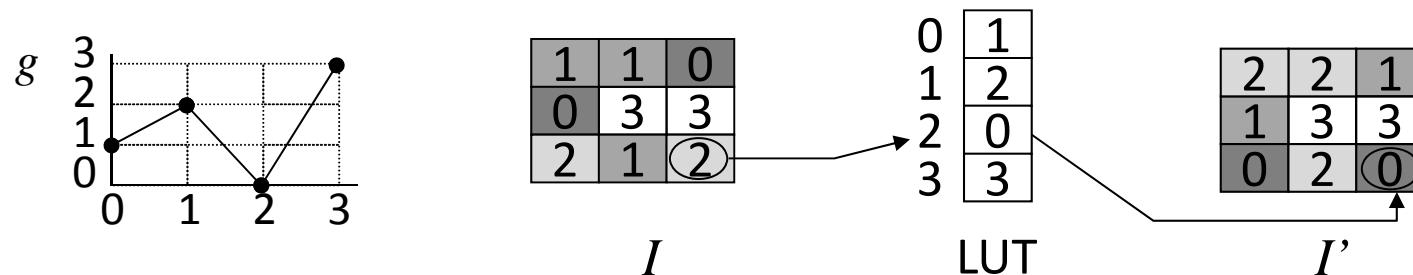


Exponencial



L.U.T. (*look-up table*)

- Al trabajar con señales discretas y finitas, la transformación g se puede representar en forma de vector de indexación (LUT).



L.U.T. (*look-up table*)

- Aplicar una LUT es definir un vector cuyos valores indican las posibles salidas. Los valores de los píxeles de la imagen de entrada indexan los nuevos valores.
- **Ejemplo:** para $g(x) = 255 - x$ haremos:

```
>> x = 0:255;  
>> lut = 255 - x;  
>> imLUT = uint8(lut(im + 1));  
>> imshow([im, imLUT]);
```

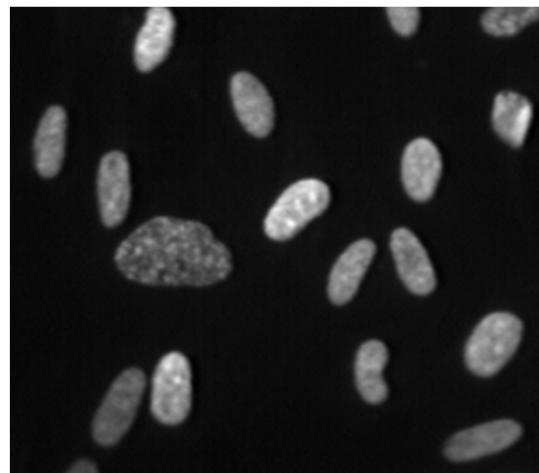
- **Ejercicio:** Calcular el logaritmo con una lut.

Operaciones lógicas

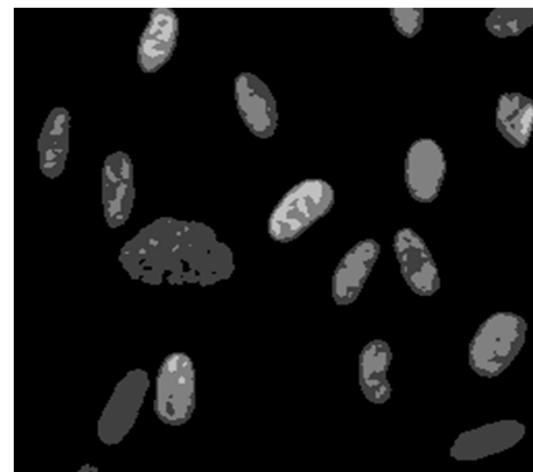
- **AND, OR, NOT, XOR** a nivel de bit entre el valor de cada píxel y una cierta máscara binaria. Por ejemplo, el **AND** lógico entre una imagen y la constante 11110000 hace una recuantización uniforme de una imagen Byte (8 bits, 256 niveles) a 16 niveles de intensidad (4 bits) únicamente.
- Otro tipo de operaciones lógicas a nivel de bit son los **desplazamientos a la izquierda y derecha** de bits, equivalentes en imágenes Byte al producto y división por potencias de 2, pero a una mayor velocidad.

Operaciones lógicas. Ejemplo

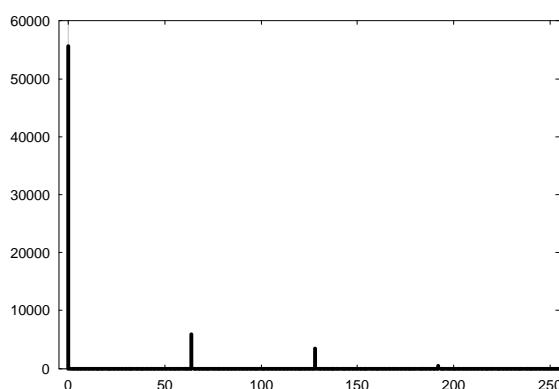
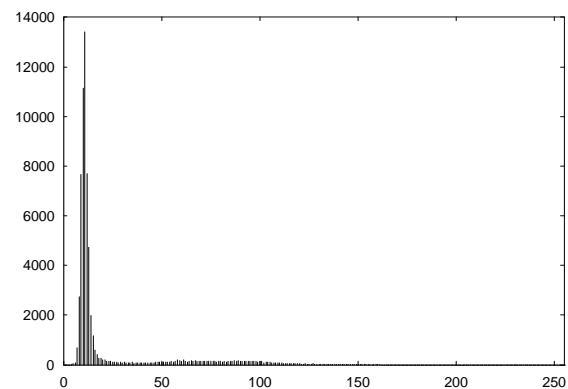
Imagen original



AND con 11000000

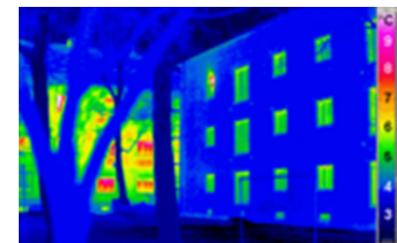


Recuantización
a 4 niveles de
intensidad



Pseudocolor y falso color

- **Pseudocolor:** visualizar en color una imagen de tonos de gris. Asignando a cada nivel de gris un color.
- Normalmente se diseña la paleta de colores con variaciones graduales. Por ejemplo, de colores fríos a cálidos.
- **Falso color:** Consiste en modificar los canales de color para que el color resultante resalte algún aspecto de la imagen. Se cambia el color típico de los objetos, captando la atención del observador.



Pseudocolor y falso color

- Podemos aplicar pseudocolor para incrementar el contraste:

```
>> im = imread('cell.tif');  
>> imP=label2rgb(im,@jet,'black','noshuffle');  
>> subplot(1,2,1); imshow(im);  
>> subplot(1,2,2); imshow(imP);
```

- Esta asignación también se puede hacer desordenada:

```
>> im = imread('cell.tif');  
>> imP=label2rgb(im,@jet,'black','shuffle');  
>> subplot(1,2,1); imshow(im);  
>> subplot(1,2,2); imshow(imP);
```

Ecualización

- La ecualización persigue que todos los niveles de gris tengan la misma frecuencia de aparición (**histograma plano**). En imágenes discretas sólo será “aproximadamente” plano.
- puede aproximarse por una LUT que consiste en el histograma acumulado llevado al rango de la imagen de entrada:

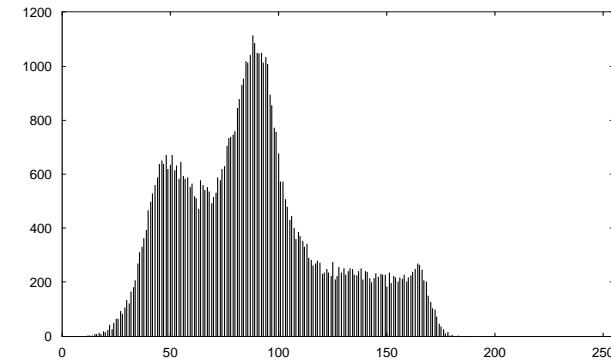
$$g(n) = \frac{1}{N \cdot M} \sum_{i=1}^{I(x,y)} H[i]$$

- Variaciones:
 - **Ecualización local**: para cada píxel se calcula el histograma en una vecindad.
 - **Ecualización logarítmica**: se eleva el histograma acumulado a una constante cambiando la forma del histograma final.

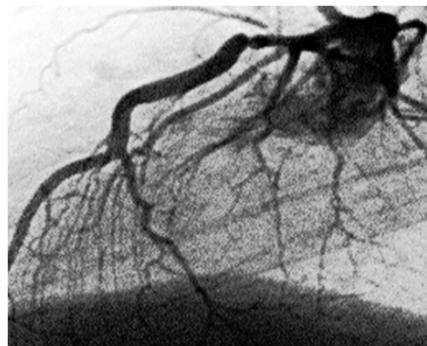
Ecualización. Proceso



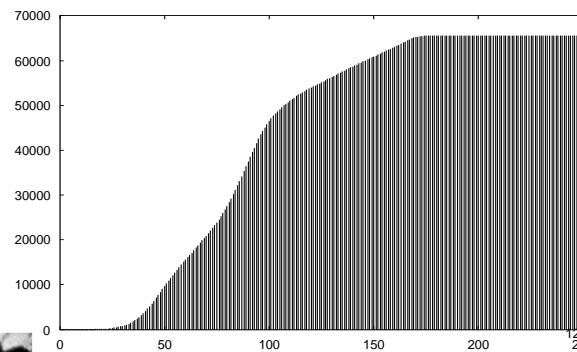
Histograma



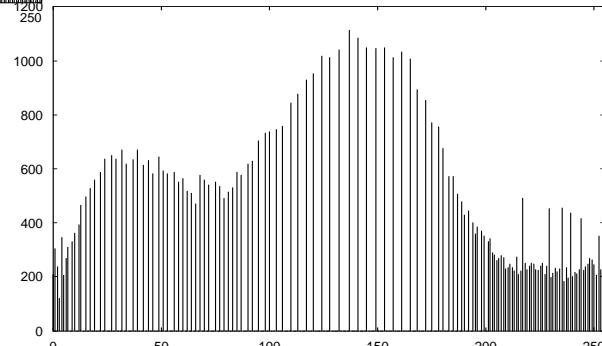
LUT



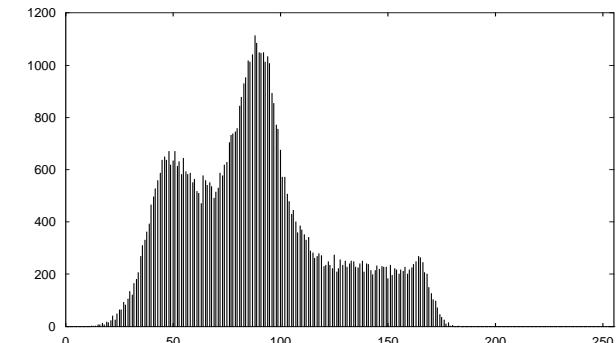
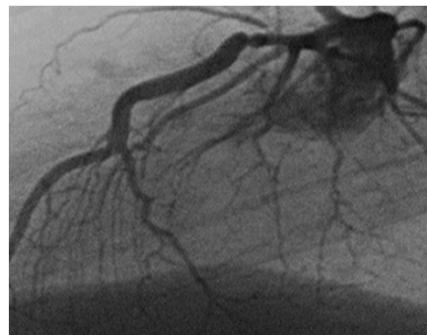
Histograma
acumulado



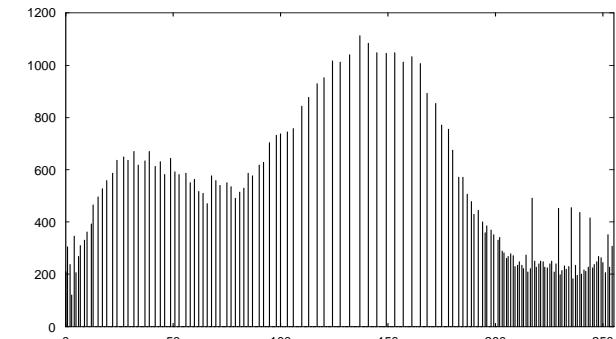
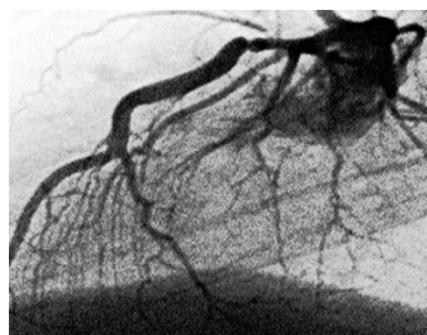
Histograma
final



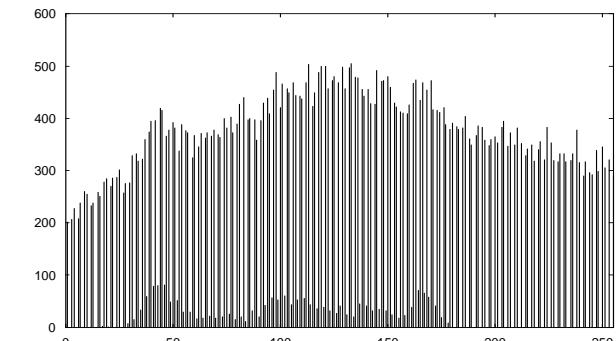
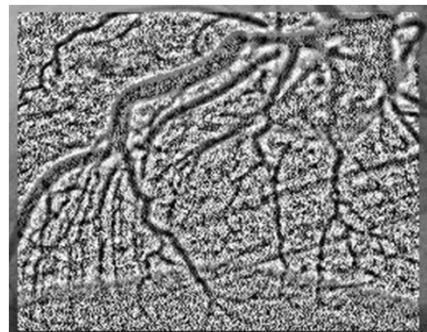
Ecualización. Ejemplo



Ecualización
global



Ecualización
local con
ventana de
13x13



Transformaciones aritméticas

- Las operaciones aritméticas son el resultado de aplicar, punto a punto, un operador a dos imágenes:

$$I'(x, y) = I_1(x, y) \circ I_2(x, y)$$

- Donde el operador \circ puede ser cualquier operación aritmética o lógica:
 - Suma
 - Resta
 - Producto
 - División
 - AND, OR, XOR

Transformaciones aritméticas

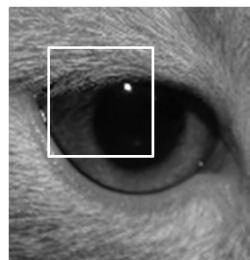
- En MATLAB se pueden realizar operaciones aritméticas con operadores (+,-,.*./) o con funciones:
 - **Suma:** imadd(im1,im2)
 - **Resta:** imsubtract(im1,im2)
 - **Multiplicación:** immultiply(im1,im2)
 - **División:** imdivide(im1,im2)
- **Ejemplo:** Eliminación de variación del fondo de una imagen.

```
>> im = imread('rice.png');  
>> background = imopen(im, strel('disk',15));  
>> im2 = imsubtract(im,background);  
>> subplot(1,3,1);imshow(im);  
>> subplot(1,3,2);imshow(background);  
>> subplot(1,3,3);imshow(im2); colormap jet;
```

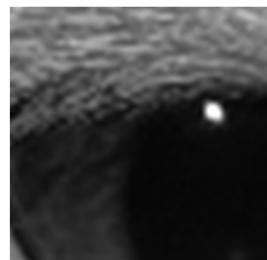
Adición

- Uno de los usos más importantes es la reducción de ruido debido a la adquisición. Si la escena es estacionaria, podemos promediar N imágenes del mismo objeto.
- Si el ruido es:
 - Aditivo: resultado=imagen+ruido.
 - No correlacionado: no forma ningún patrón.
 - De promedio nulo: valores distribuidos simétricamente entorno al 0.
- Promediando N imágenes aumentamos la *relación señal-ruido* (SNR) en un factor $N^{1/2}$. A mayor SNR , mejor imagen.

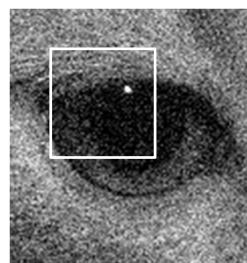
Adición. Ejemplo



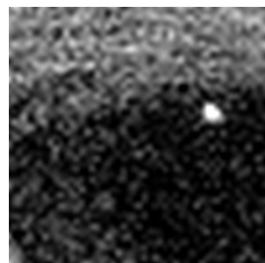
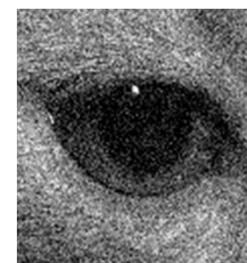
original



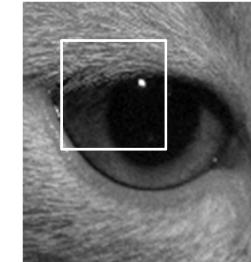
detalle



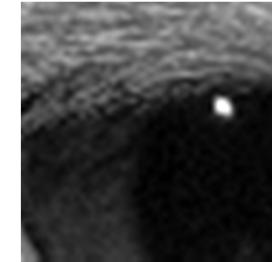
... (64 ims) ...



detalle



promedio

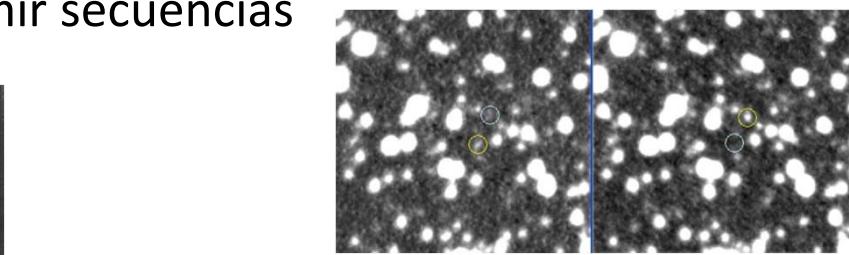
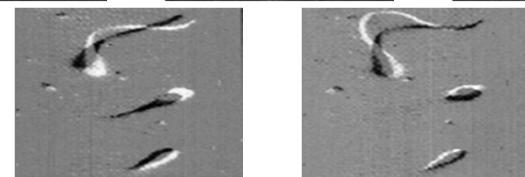


detalle

Sustracción

**Eliminación de fondo (*background subtraction*),
disparidad, detección de movimiento**

- detectar presencia, ausencia o movimiento.
- eliminar de la imagen los elementos que no cambian.
- reducir información para comprimir secuencias



Producto

- Elección de **regiones de interés** (ROI): multiplicación de la imagen por una máscara con 1 en la zona de interés y 0 en el resto.
 - **Filtrado** en el dominio frecuencial (Fourier): en el espacio de frecuencias el producto equivale a la convolución.

- Reducción de **efectos de borde**: multiplicación por una imagen cuyos valores decaen hacia la periferia.

Producto. Ejemplos

- Regiones de interés.

imagen



máscara

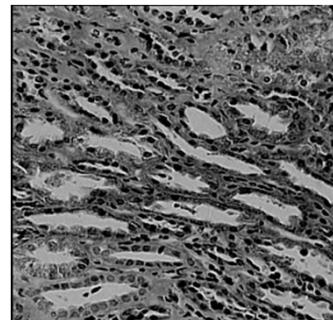


ROI's

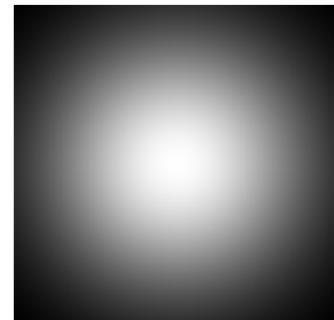


- Reducción de efectos de borde.

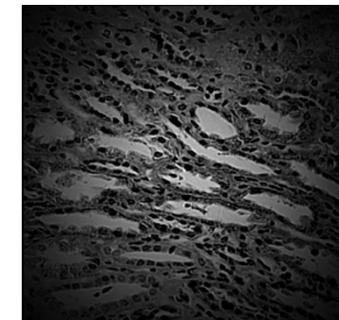
original



máscara

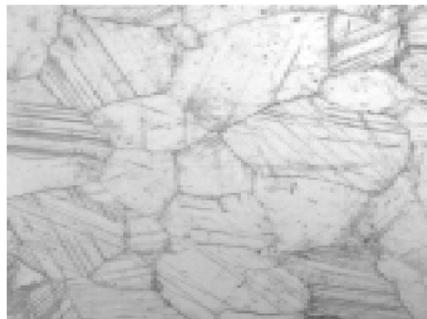


resultado



División

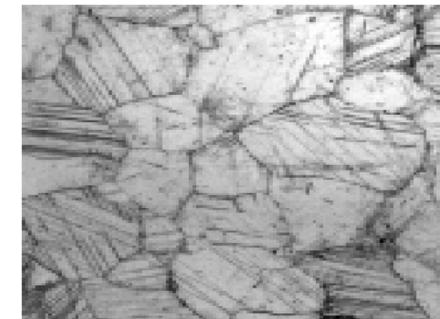
- Si podemos modelar la formación de la imagen con una multiplicación podemos eliminar un factor con la división.
Por ejemplo, imagen de microscopía (luz transmitida).
 - t_m transmitancia de muestra. t_f la iluminación no homogénea que queremos eliminar. I_i es la intensidad de iluminación constante.
 - I_m, I_f son imagen muestra y fondo. Tenemos: $I_m = I_i t_f t_m, I_f = I_i t_f$.
- Nos interesa la muestra: $t_m = I_m / I_f$



muestra



fondo



corrección