



# **Operaciones puntuales y procesamiento de histogramas**

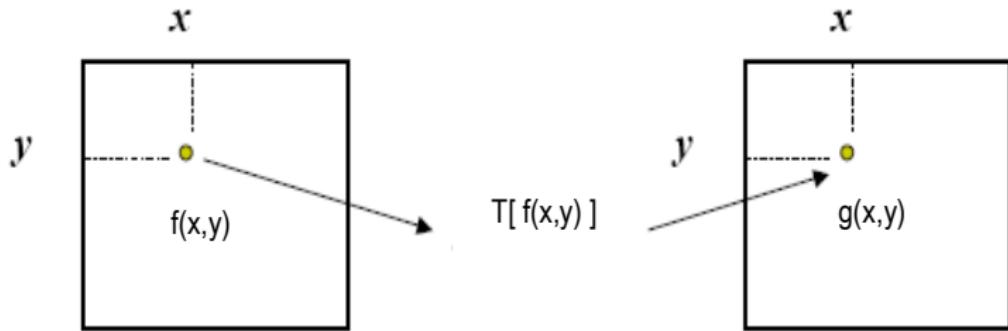
Unidad 1

BME423 · Procesamiento de imágenes médicas

Alejandro Veloz

Ingeniería Biomédica

# Operadores puntuales

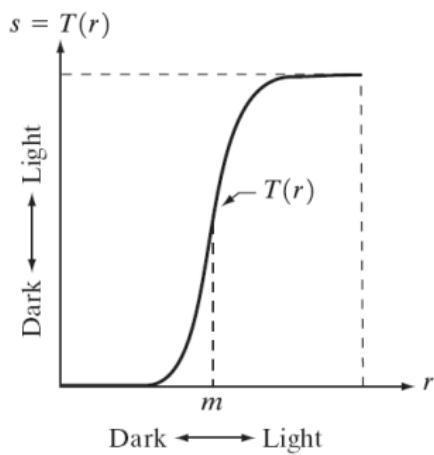


$$g(x, y) = T[ f(x, y) ]$$

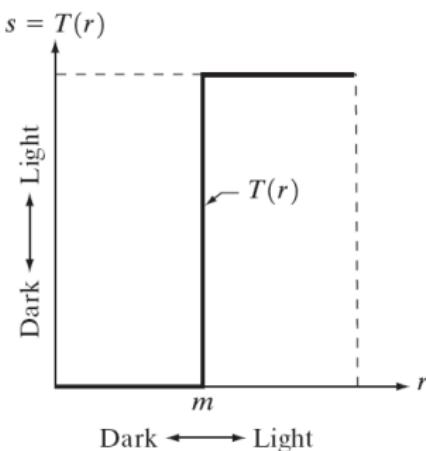
El valor de la **imagen de salida**  $g$  en el punto  $(x, y)$  sólo depende del valor de la **imagen de entrada**  $f$  en la posición  $(x, y)$ .

# Operadores puntuales

*contrast stretching*



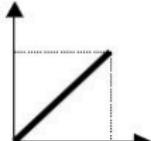
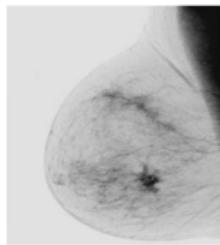
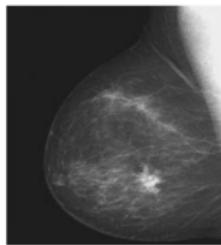
Caso límite: *thresholding function*



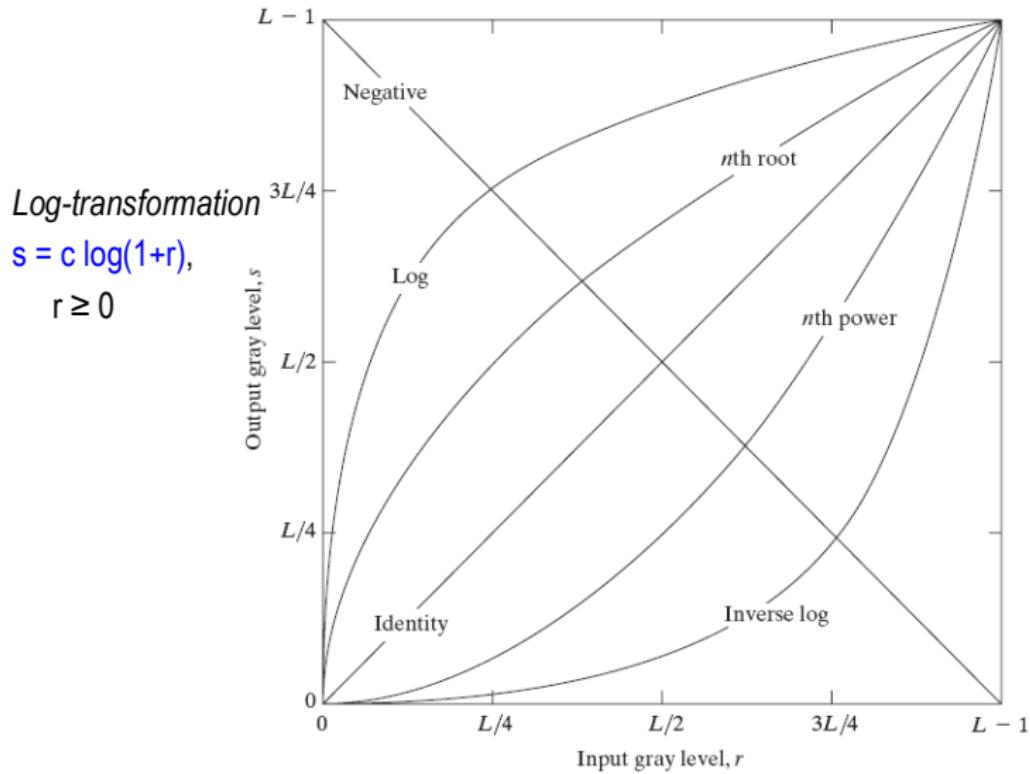
# Operadores puntuales

## Operadores lineales:

- (i) Operador identidad:  $s = r$
- (ii) Operador inverso o negativo:  $s = L - 1 - r$



# Operadores puntuales



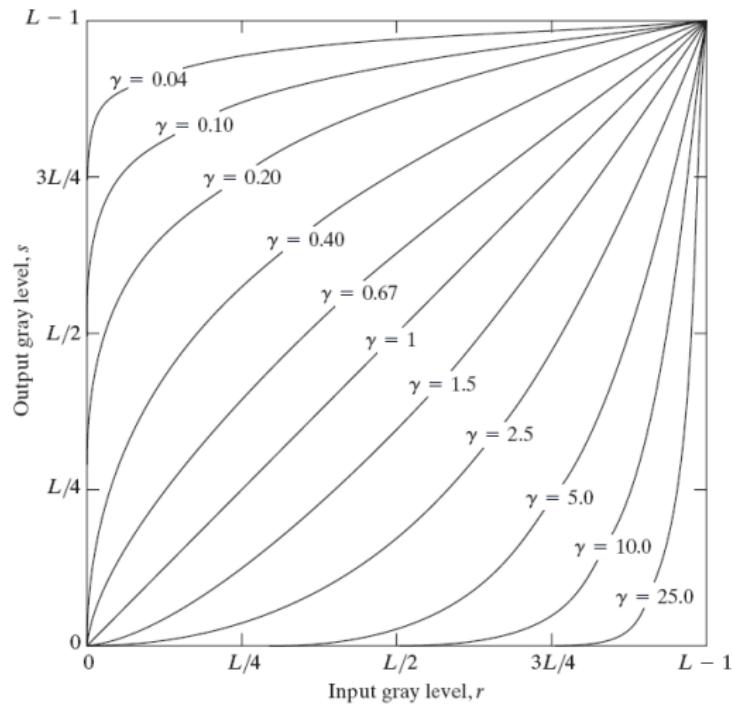
# Operadores puntuales

*Power-law transformation*

$$s = c r^\gamma,$$

$$c, \gamma > 0$$

$$s = c (r + \varepsilon)^\gamma$$



# Operadores puntuales

*Power-law transformation function*

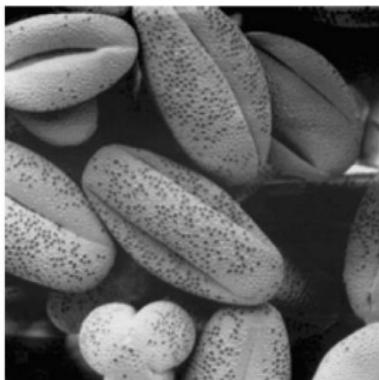
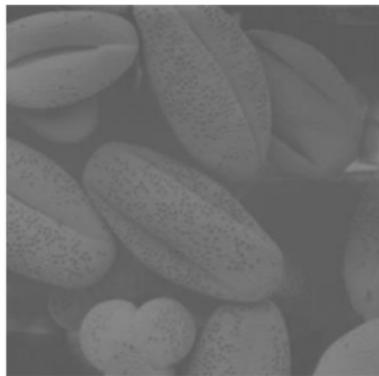
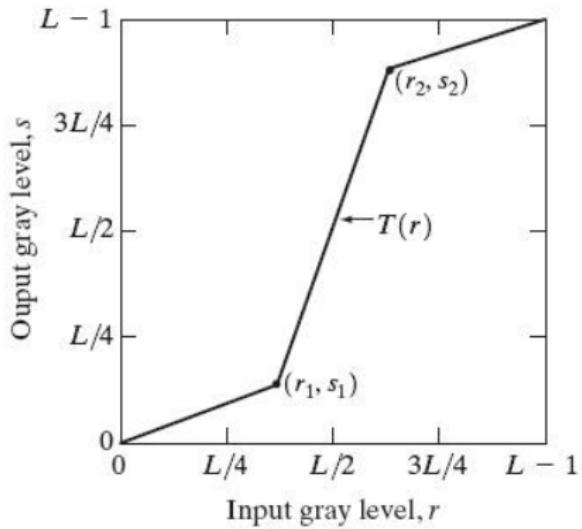
a b  
c d

(a) Aerial image.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 3.0, 4.0$ , and  $5.0$ , respectively. (Original image for this example courtesy of NASA.)



# Operadores puntuales

*Piecewise-Linear Transformation  
Functions*



# Operadores puntuales

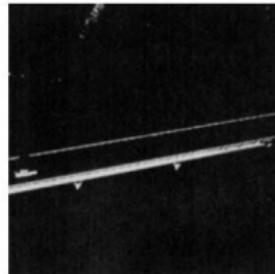
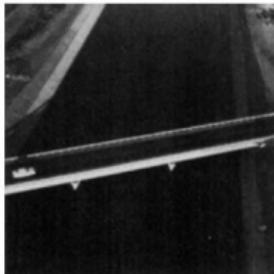
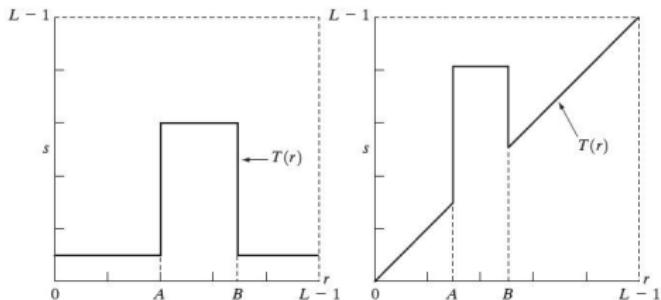
*Piecewise-Linear Transformation  
Functions*

Gray-level slicing.

$$g(x,y) = k$$

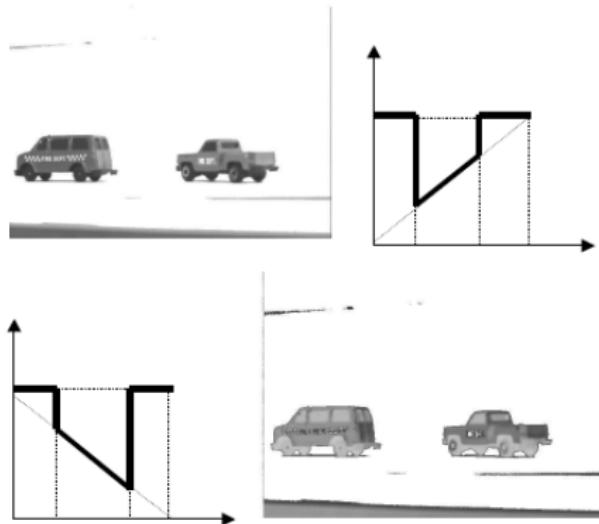
para  $A \leq f(x,y) \leq B$ ,  
con  $k$ ,  $A$  y  $B$  en  
 $[0,L-1]$ .

$g(x,y) = f(x,y)$   
en otro caso.



# Operadores puntuales

*Piecewise-Linear Transformation  
Functions*



# Operadores puntuales

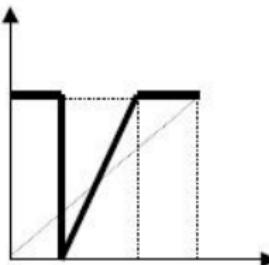
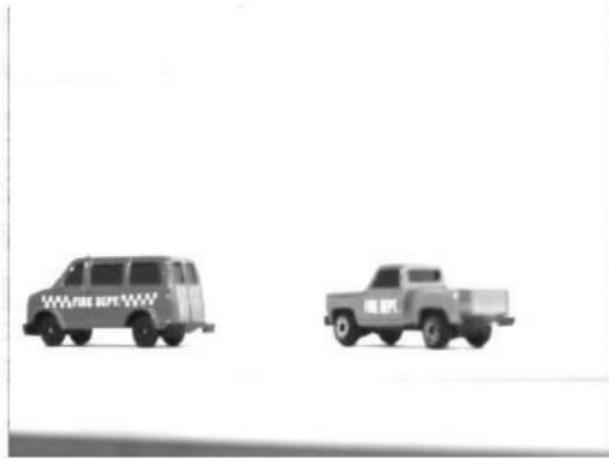
## *Piecewise-Linear Transformation Functions*

$$g(x,y) = (L - 1) (f(x,y) - A) / (B - A)$$

para  $A \leq f(x,y) \leq B$ , con  $A$  y  $B$  en  $[0,L-1]$ .

$$g(x,y) = L - 1$$

en otro caso.



# Operadores puntuales

## *Piecewise-Linear Transformation Functions*

Reducción de niveles de gris.

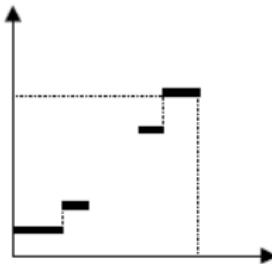
$$g(x,y) = k_1 \quad \text{para} \quad 0 \leq f(x,y) \leq th_1,$$

$$g(x,y) = k_2 \quad \text{para} \quad th_1 < f(x,y) \leq th_2,$$

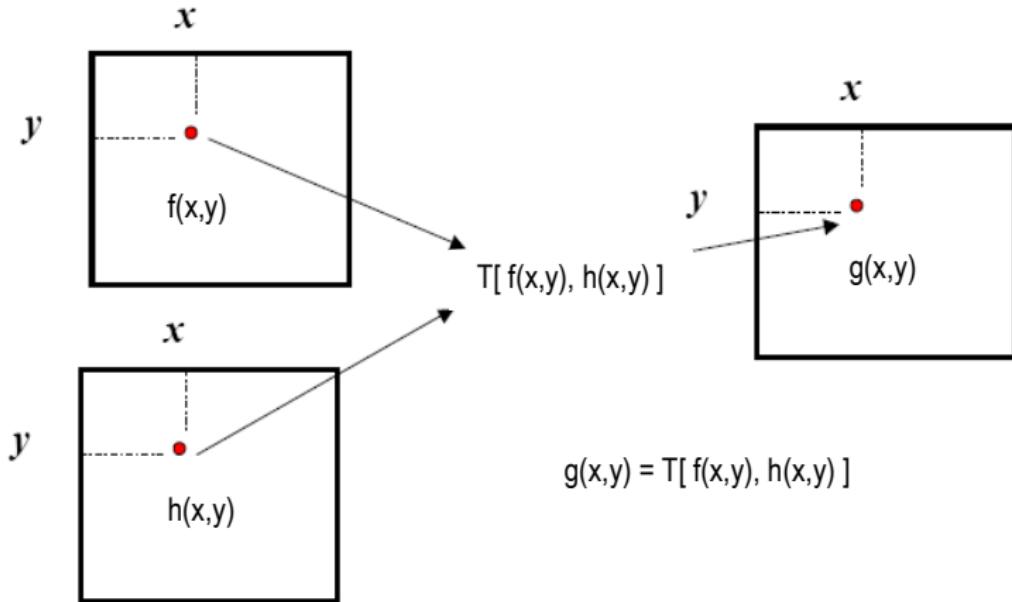
...

$$g(x,y) = k_n \quad \text{para} \quad th_n < f(x,y) \leq L - 1,$$

con  $th_1, th_2, \dots, th_n$  en  $[0, L-1]$



# Operadores puntuales



# Operadores puntuales



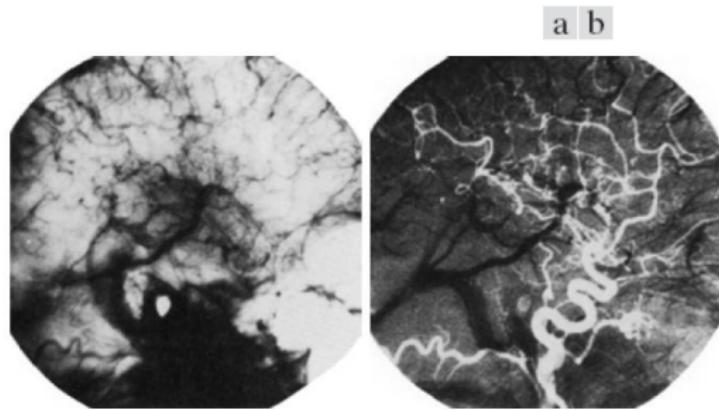
$$g(x,y) = [ f(x,y) + h(x,y) ] / 2$$



# Operadores puntuales

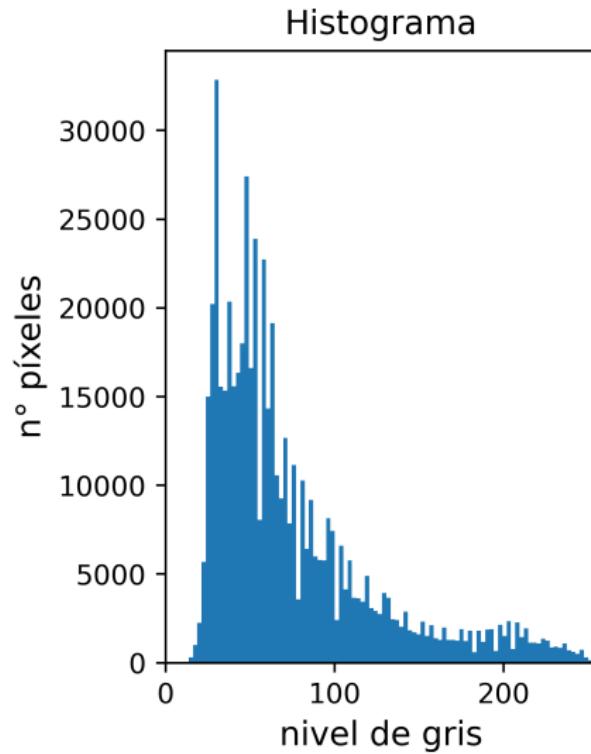
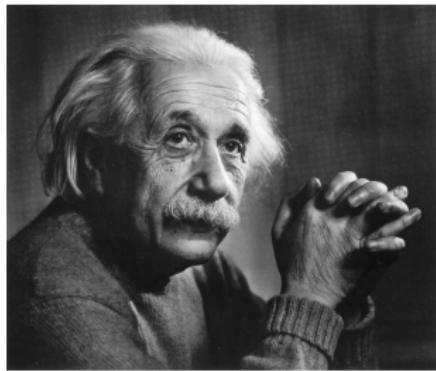
$$g(x,y) = f(x,y) - h(x,y)$$

*Mask mode radiography*



Enhancement by image subtraction.  
(a) Mask image.  
(b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

# Histograma de una imagen



# Histograma de una imagen

```
import os
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

filename = "einstein.tif"

img = cv.imread(filename, 0)

fig, axs = plt.subplots(num=1, figsize=(7,4), nrows=1, ncols=2)

axs[0].imshow(img, cmap="gray", vmin=0, vmax=255)
axs[0].set_axis_off()

axs[1].hist(img.flatten(), bins=100)
axs[1].set_xlim([0,255])
axs[1].set_title("Histograma", fontsize=12)
axs[1].set_xlabel("nivel de gris", fontsize=12)
axs[1].set_ylabel("num. pixeles", fontsize=12)

plt.savefig("einstein_histo.png", dpi=500)
plt.show()
```

# Histograma de una imagen

- Consideremos una imagen de  $N$  píxeles que posee niveles de intensidad  $\{r_0, \dots, r_{L-1}\}$
- El n° de píxeles de la imagen que tienen intensidad  $r_k$ :

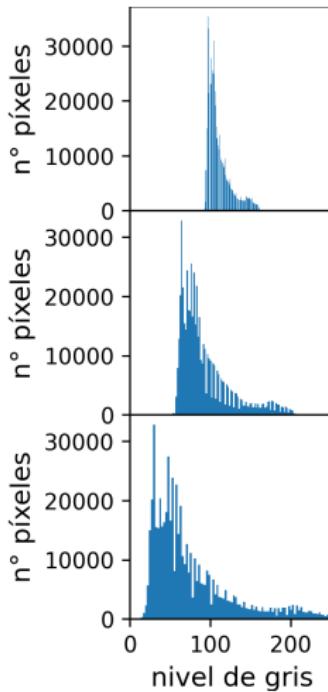
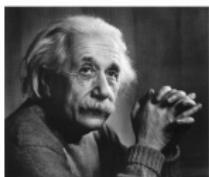
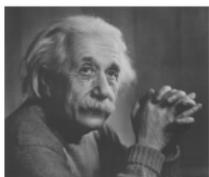
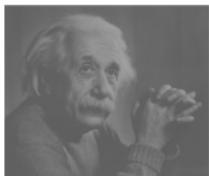
$$h(r_k) = n_k$$

- Al dividir cada  $n_k$  por el número total de píxeles  $N$  de la imagen, resulta la probabilidad de ocurrencia de  $r_k$ :

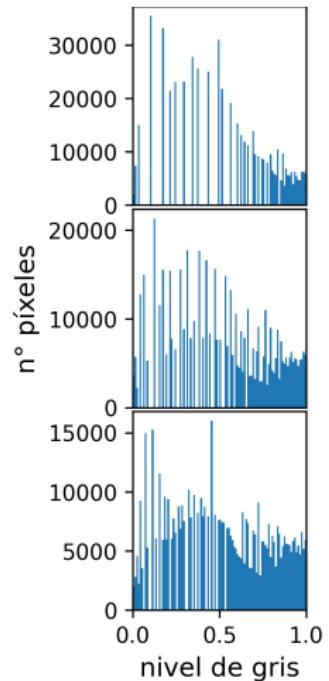
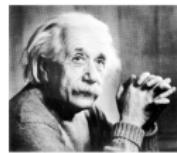
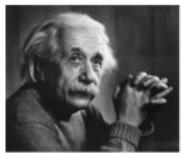
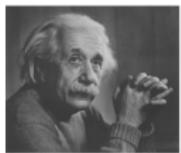
$$p(r_k) = \frac{n_k}{N}, \quad \sum_{k=0}^{L-1} p(r_k) = 1$$

donde  $p(r_k)$  es la prob. de que un píxel tenga intensidad  $r_k$

# Histograma de una imagen



# Histograma de una imagen



# Ecualización de histograma

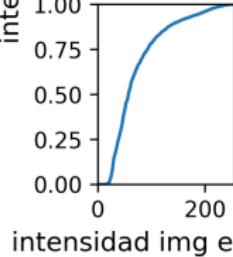
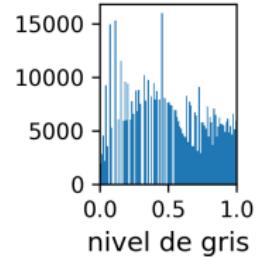
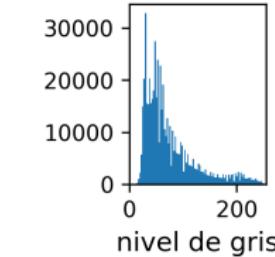
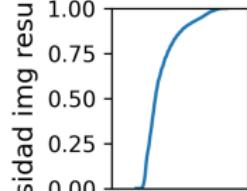
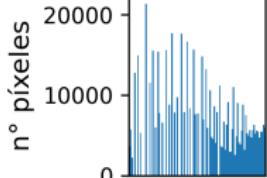
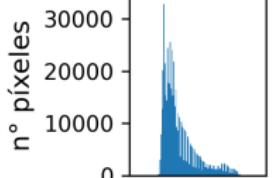
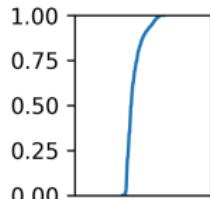
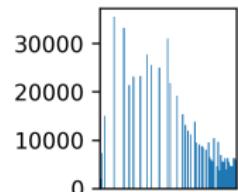
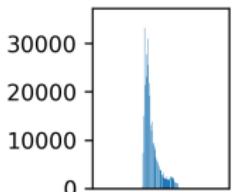
- Consiste en usar como función de transformación la función de distribución empírica
- La intensidad del píxel  $i$  se denota por  $x_i \in \{r_0, \dots, r_{L-1}\}$ , con  $i \in \{1, \dots, N\}$  y se le asigna masa  $p(x_i) = 1/N$
- Se tendrá una nueva imagen con intensidades  $\{s_0, \dots, s_{L-1}\}$ , dadas por:

$$s_k = \hat{F}(r_k) = \sum_{i=1}^N \frac{I(x_i \leq r_k)}{N} = \sum_{j=0}^k p(r_j) = \sum_{j=0}^k \frac{n_j}{N} \quad \in [0, 1]$$

con

$$I(x_i \leq r_k) = \begin{cases} 1, & \text{si } x_i \leq r_k \\ 0, & \text{en otro caso} \end{cases}$$

# Ecualización de histograma



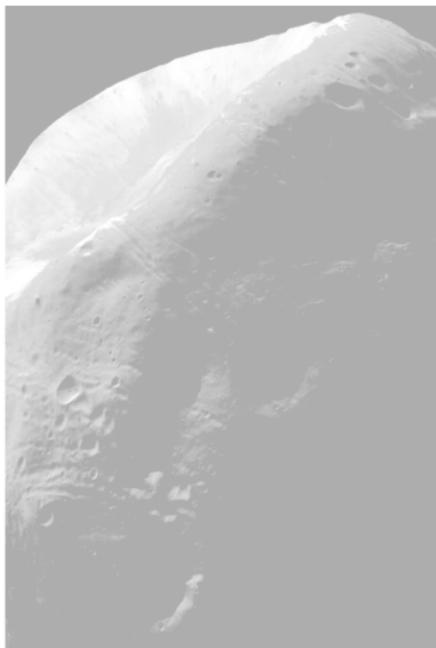
# Ecualización de histograma

```
import time
import numpy as np

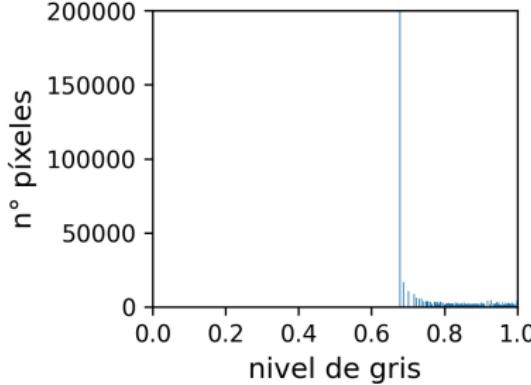
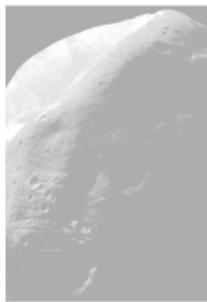
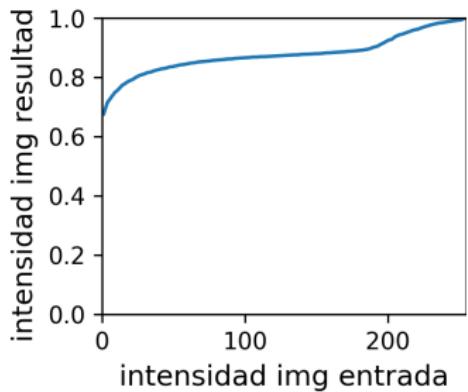
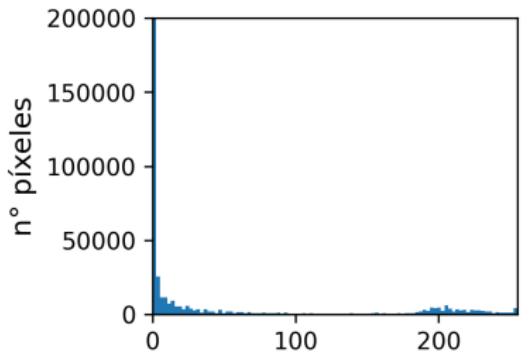
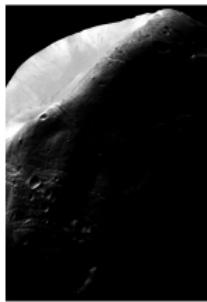
def eq_hist(img, nbins=255):
    t1 = time.time()
    imgv = img.flatten()
    h, rk = np.histogram(imgv, bins=nbins)
    rk = rk[:-1] + (rk[1:] - rk[:-1])/2
    pk = h / h.sum()
    sk = np.cumsum(pk)
    imgv_out = np.interp(imgv, rk, sk)
    print('elapsed time: ', time.time()-t1)
    return imgv_out.reshape( img.shape )
```

# Ecualización de histograma

Hay casos donde la ec. de histograma no funciona, ¿por qué?



# Ecualización de histograma



# Especificación de histograma (specification, matching)

$$s_k = T(r_k) = \sum_{j=0}^k p(r_j) = \sum_{j=0}^k \frac{n_j}{N}$$

$$z_k = G^{-1} \underbrace{[T(r_k)]}_{s_k}$$

$$k = 0, 1, 2, \dots, L - 1$$

