



Recuperación de Información Multimedia

Procesamiento de Imágenes

CC5213 – Recuperación de Información Multimedia
Departamento de Ciencias de la Computación
Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2019



Procesamiento de Imágenes

- Una imagen es una señal bidimensional discretizada
 - Cada valor de la imagen se le llama “pixel” (picture element)
 - Ancho (coordenada **x**) y Alto (coordenada **y**)
 - Canales (usualmente 1, 3 ó 4)
 - Profundidad (usualmente un entero de 8 bits sin signo por canal)
 - Ubicación del origen (usualmente superior izquierda)



Procesamiento de Imágenes

■ Procesamientos comunes:

- ☐ Blur o desenfoque
- ☐ Smoothing o suavizado
- ☐ Sharp o enfoque
- ☐ Eliminación de ruido
- ☐ Resize, crop
- ☐ Ajustar brillo/contraste



Procesamiento de Imágenes

- Tipos de operaciones
 - 1) Operadores punto a punto
 - Por pixel
 - 2) Operadores lineales
 - Convolución
 - 3) Operadores no-lineales



1) Operadores punto a punto

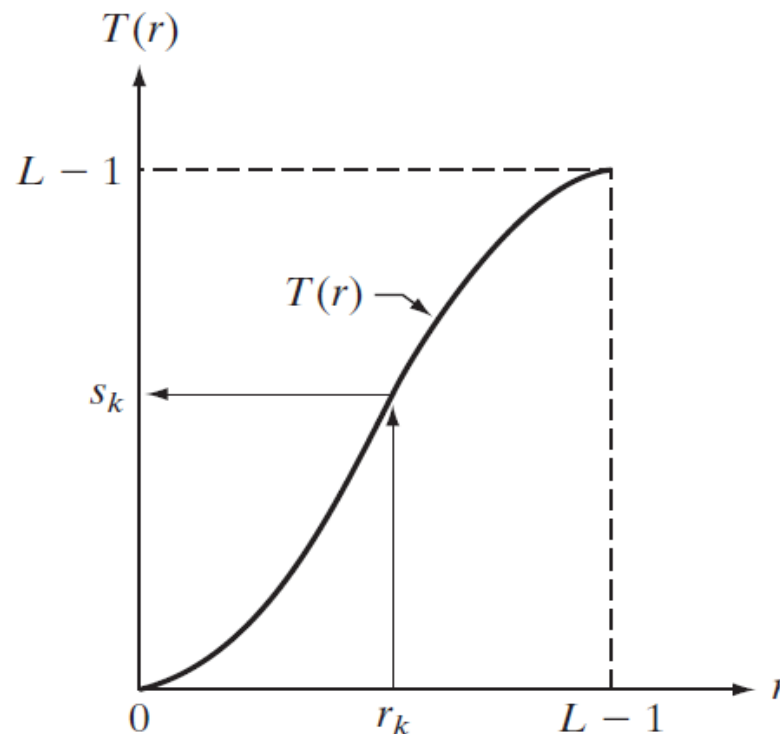
- Modificar cada pixel de la imagen de entrada I en forma independiente según una función $h()$:

$$G(i, j) = h(I(i, j))$$

- h podría depender globalmente de I , o de la posición (i, j)
- Para imágenes de 8 bits, h usualmente será una tabla de 256 entradas

1) Operadores punto a punto

- Reasignar valores de gris según alguna función (usualmente monótona creciente)



1) Operadores punto a punto

■ Ej. Brillo/contraste

□ h es un ajuste lineal

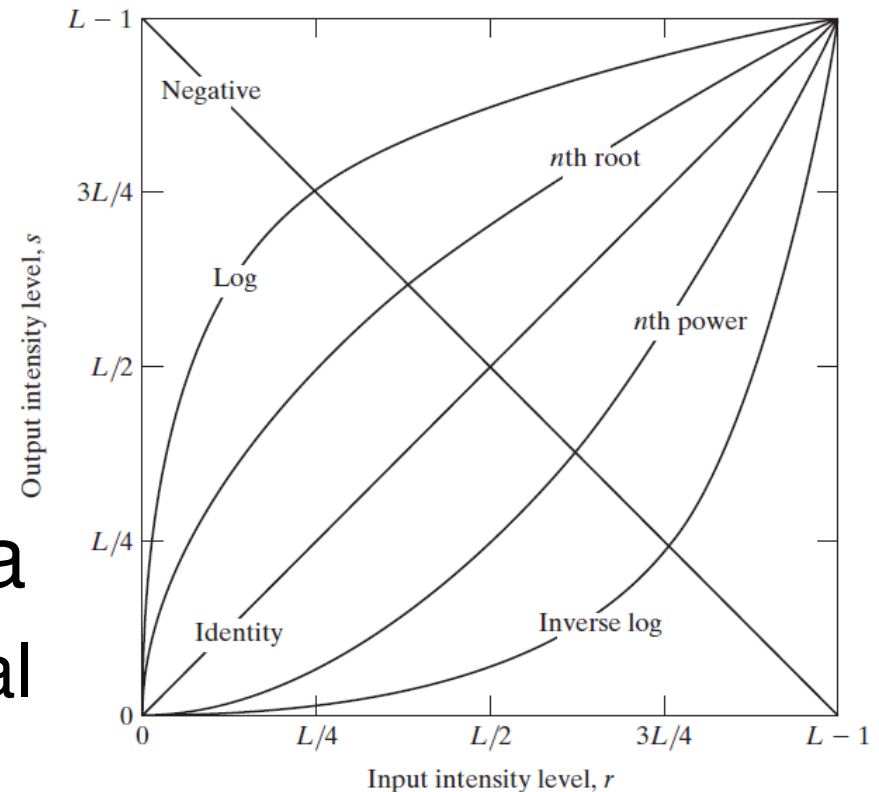
$$G(i, j) = aI(i, j) + b$$

■ Ej. Corrección gamma

□ h es un ajuste no lineal

$$G(i, j) = I(i, j)^{\frac{1}{\gamma}}$$

$$\gamma \approx 2.2$$



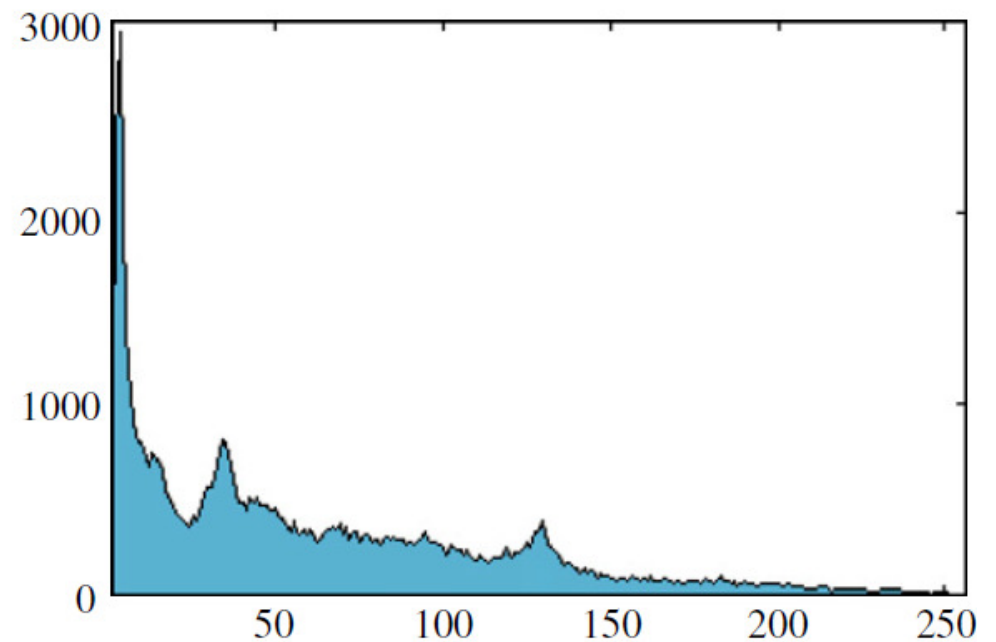


Histograma

- Contar el número de píxeles en la imagen con cada valor de intensidad posible
- Cada contador en el histograma se le llama “bin”
- Normalización: Dividir cada bin por la cantidad de píxeles
 - La suma de los bins es 1
 - Puede ser visto como la probabilidad de que un píxel tenga cierto valor

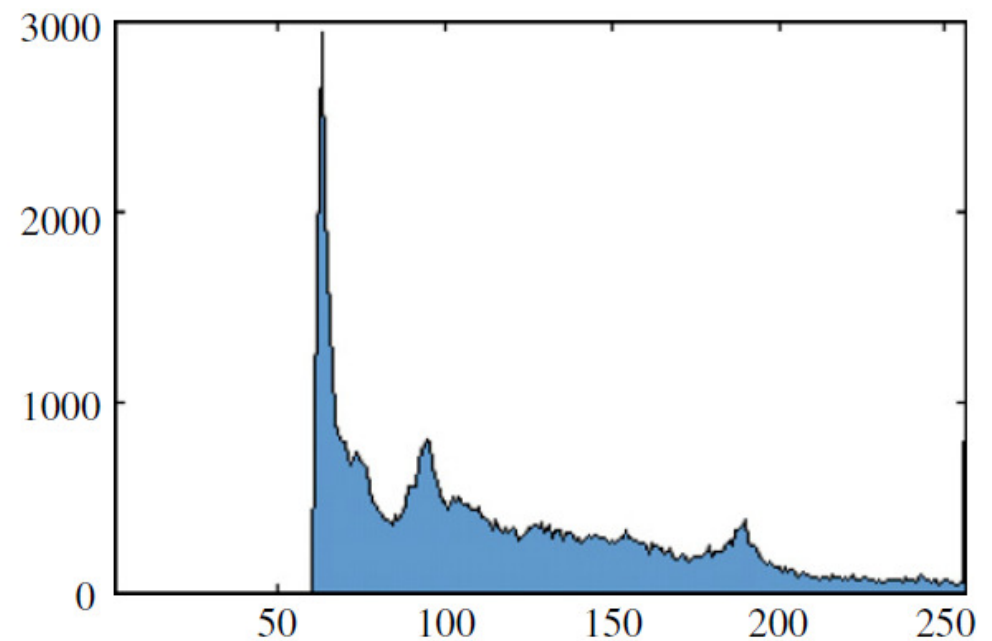
Histograma de intensidades

■ Ejemplo:

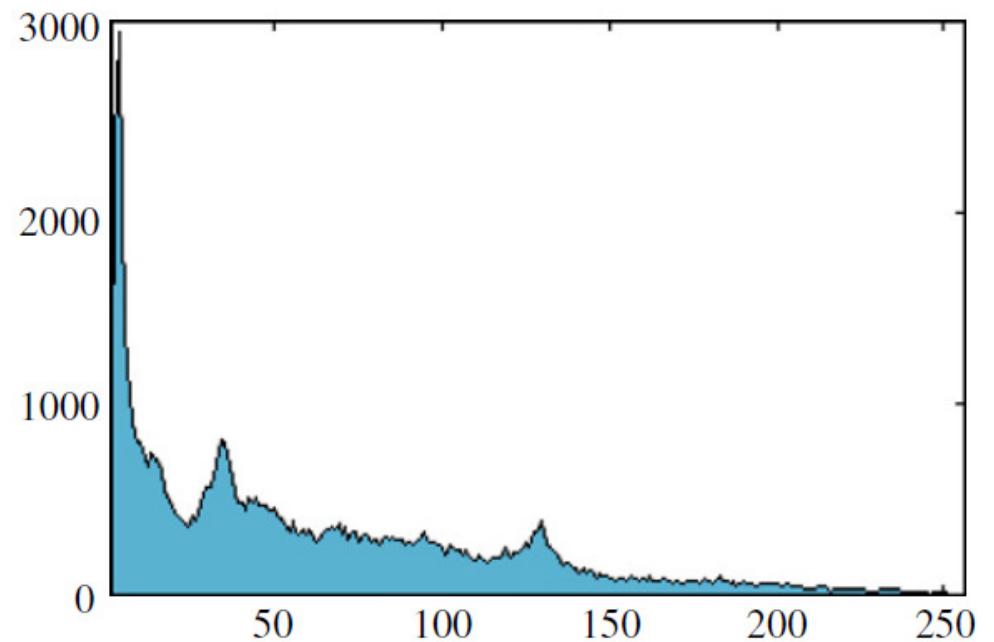


Histograma de intensidades

- Sumar 60 a cada valor

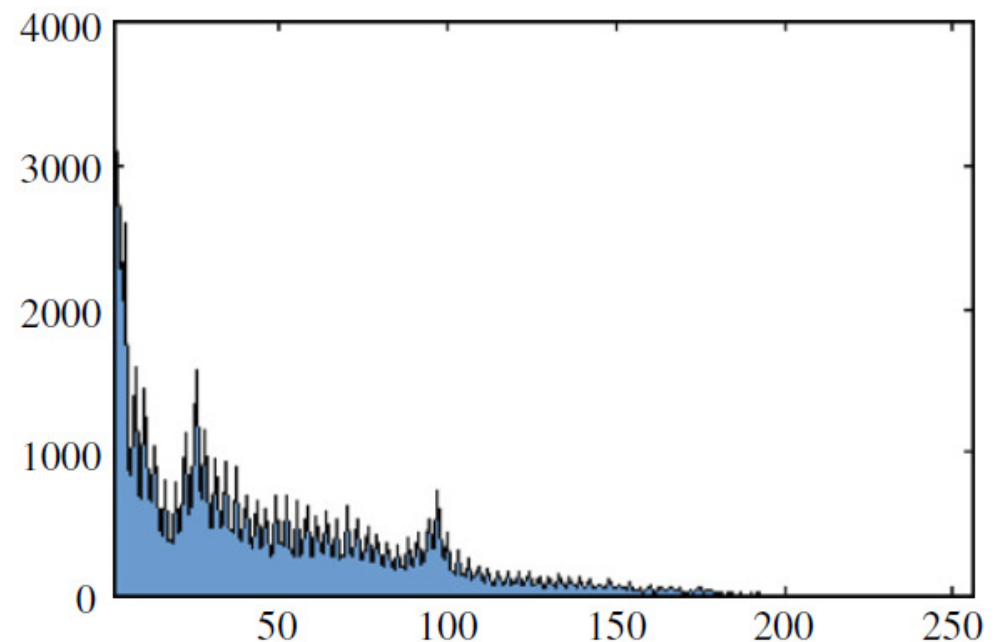


Histograma de intensidades

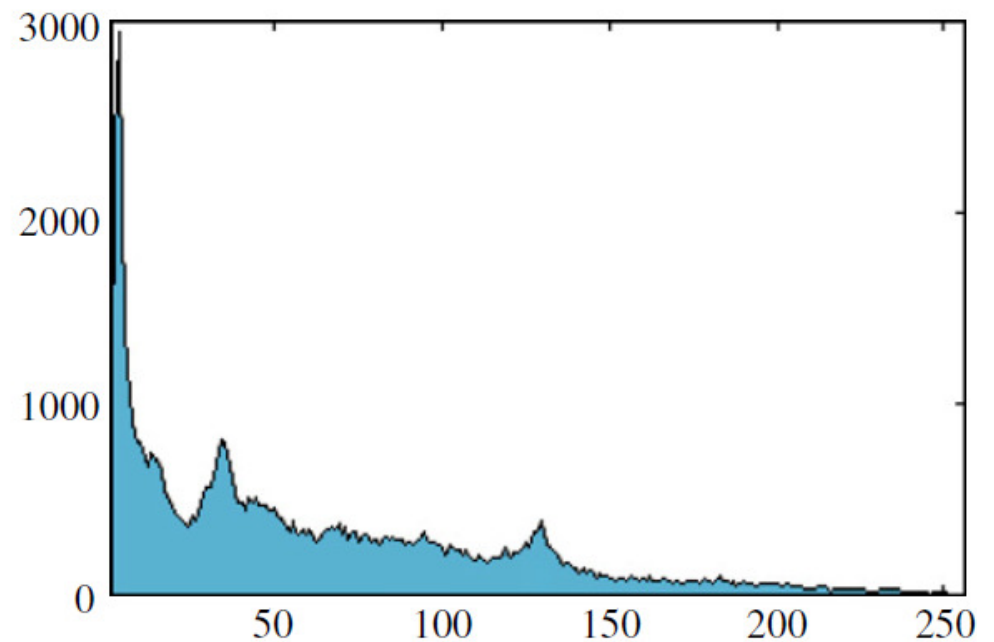


Histograma de intensidades

- Multiplicar por 0.75 cada valor

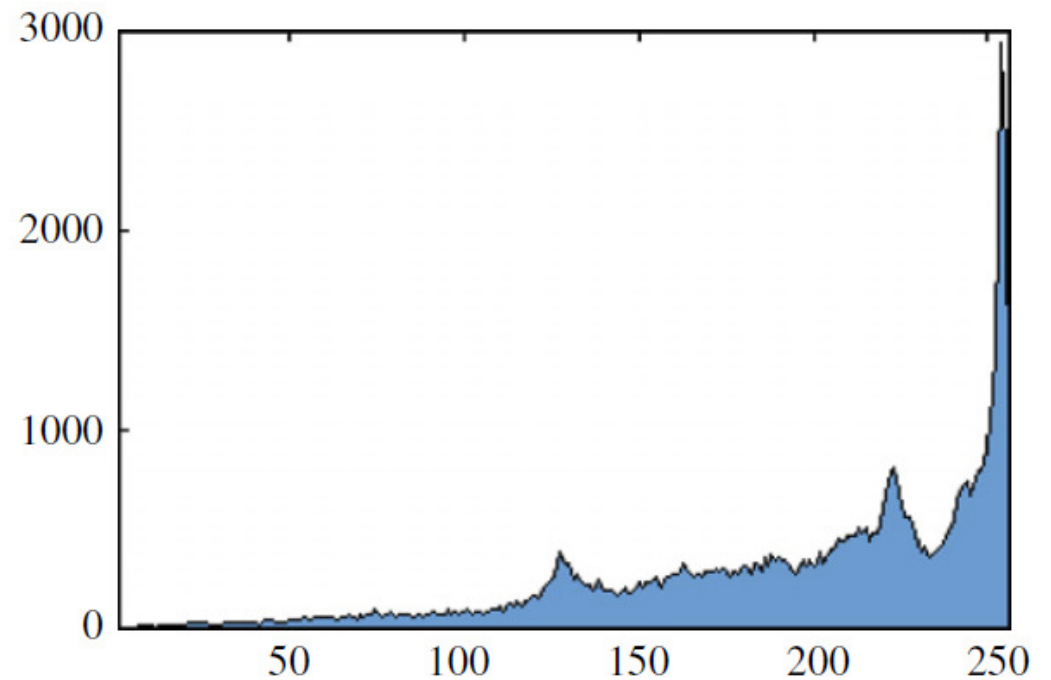


Histograma de intensidades

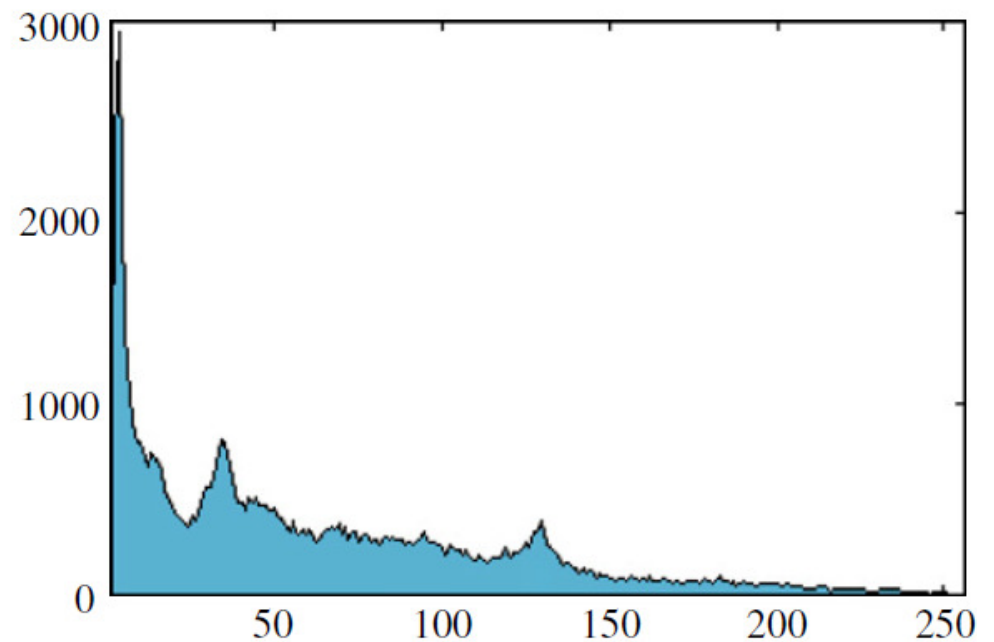


Histograma de intensidades

- Invertir cada valor ($255-b$)

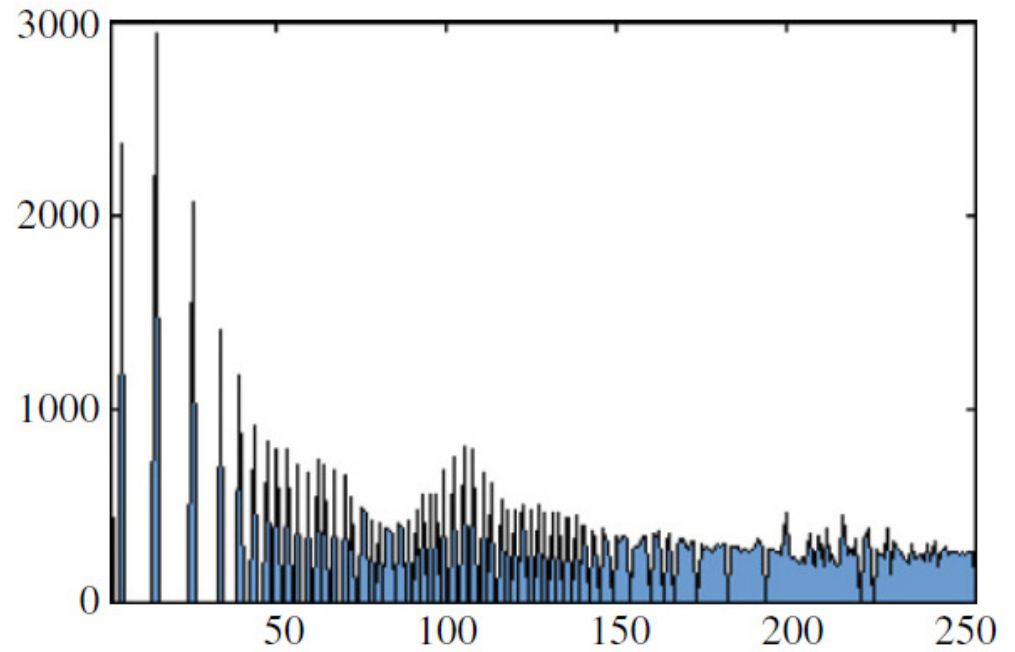


Histograma de intensidades



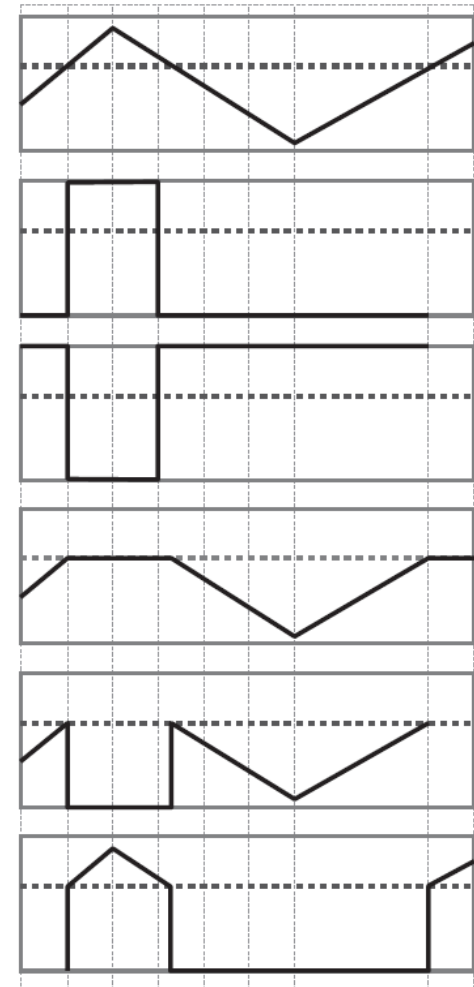
Histograma de intensidades

- Ecualización, reasignar grises para lograr una distribución cercana a uniforme



1) Operadores punto a punto

- Umbralizar o Binarizar (to threshold)
 - Dada una imagen en tonos de grises, h asigna un valor (usualmente blanco o negro) al comparar $I(i,j)$ contra una constante llamada umbral o “threshold”





Umbral de Binarización

- ¿Cómo encontrar automáticamente un buen valor umbral?
 - Algoritmo de Otsu
 - Se asume que la imagen tiene pixeles de fondo y de primer plano
 - En el histograma debieran resaltar dos conjuntos, el umbral debe dividir ambos conjuntos
 - Probar todos los posibles valores de umbral, elegir el umbral que minimiza:

$$P_{\text{blanco}} \sigma_{\text{blanco}}^2 + P_{\text{negro}} \sigma_{\text{negro}}^2$$

2) Operadores lineales

- El valor de un nuevo pixel es la suma ponderada de una ventana de pixeles
- Operador “convolución”
 - Máscara o Kernel

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

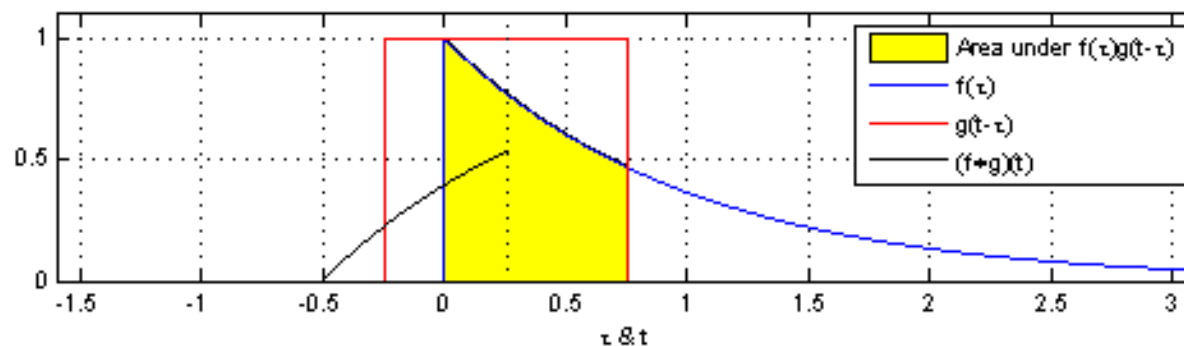
69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

Convolución

■ 1D continuo:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x - t)dt$$

Animación en <http://en.wikipedia.org/wiki/Convolution>





Convolución

- 2D continuo:

$$(f * g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(u, v) g(x - u, y - v) du dv$$

- 2D discreto:

$$(f * g)(x, y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) g(x - i, y - j)$$



Convolución

■ Propiedades:

$$(f * g) = (g * f)$$

$$(f * g) * h = f * (g * h)$$

$$f * (g + h) = (f * g) + (f * h)$$

$$a(f * g) = (af) * g = f * (ag)$$



Kernels para Blur o Smooth

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1



Blur

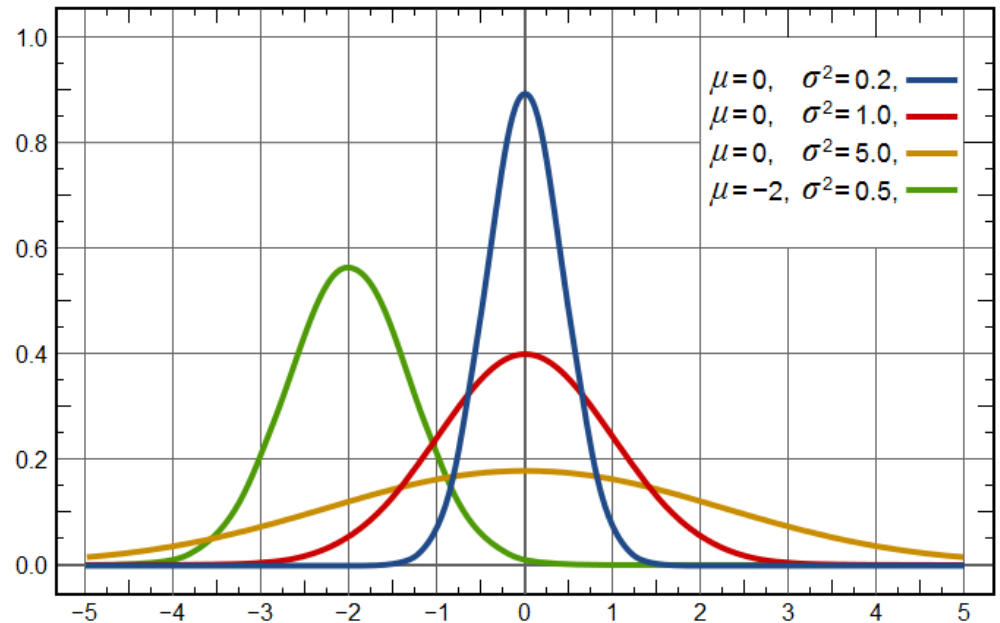
- Propiedades deseables de un filtro pasabajos:
 - ☐ Normalizado (suma 1)
 - ☐ Simétrico
 - ☐ Decreciente hacia los extremos
 - ☐ Considera todas las celdas
 - ☐ Circular ...

Kernel Gaussiano

■ Gaussiana 1D:

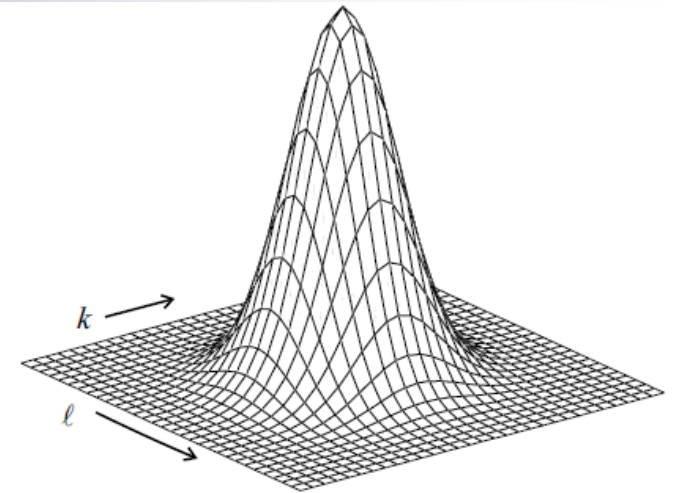
$$G_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

$$G_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Kernel Gaussiano

- Gaussiana 2D:



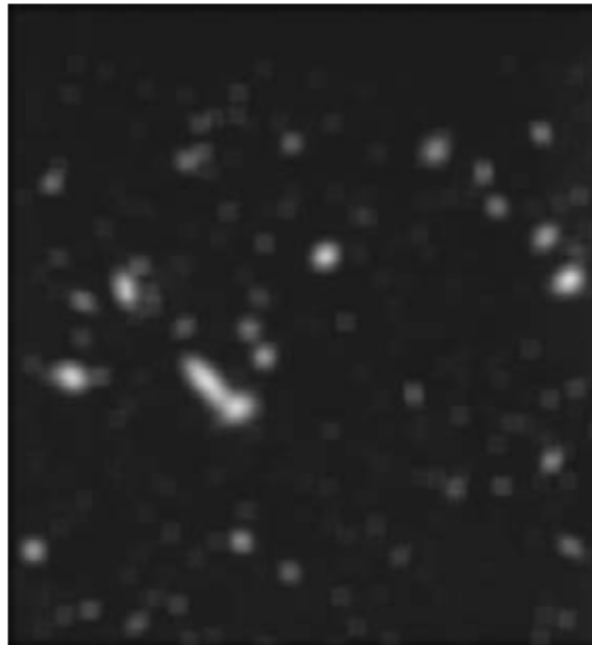
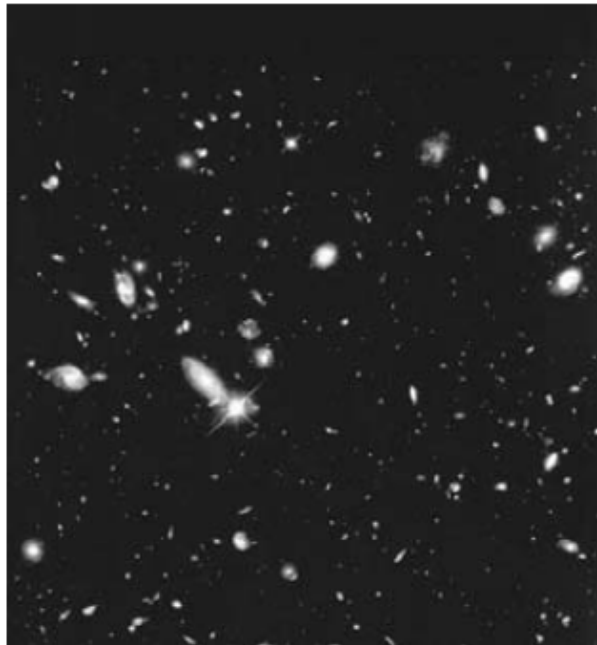
$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$G_{\mu,\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}}$$

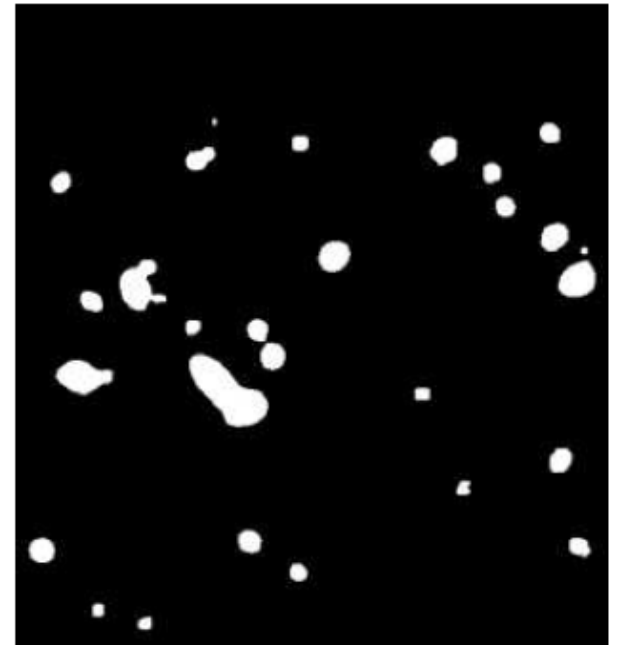
$$G_{\mu,\sigma_x,\sigma_y}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]}$$



Blur



Promedio 15x15

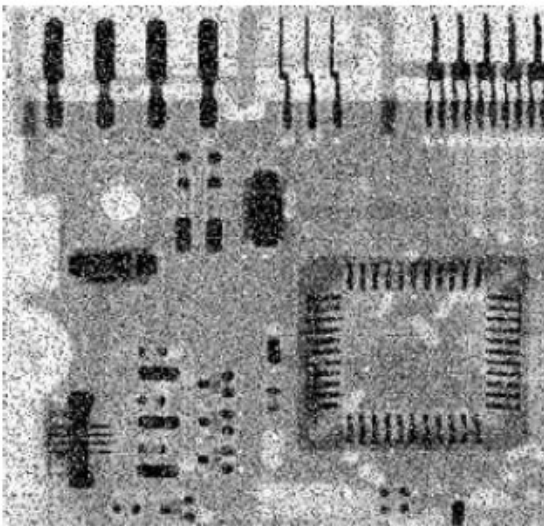


+ Umbral

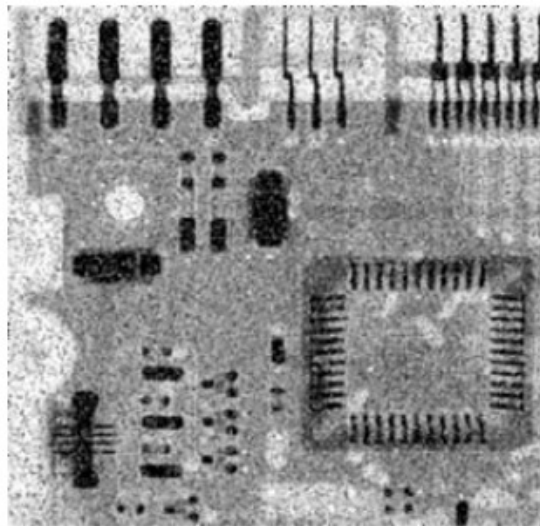
3) Operadores no lineales

- Filtro de mediana

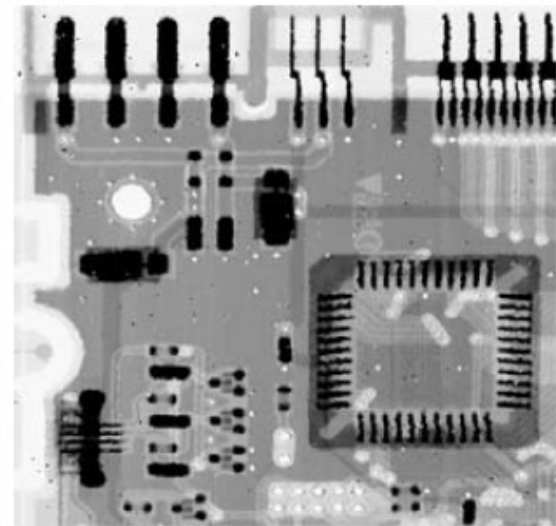
- Ruido “sal y pimienta” o impulsivo



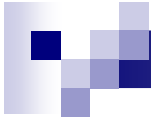
Ruido S&P



Promedio 3x3



Mediana 3x3



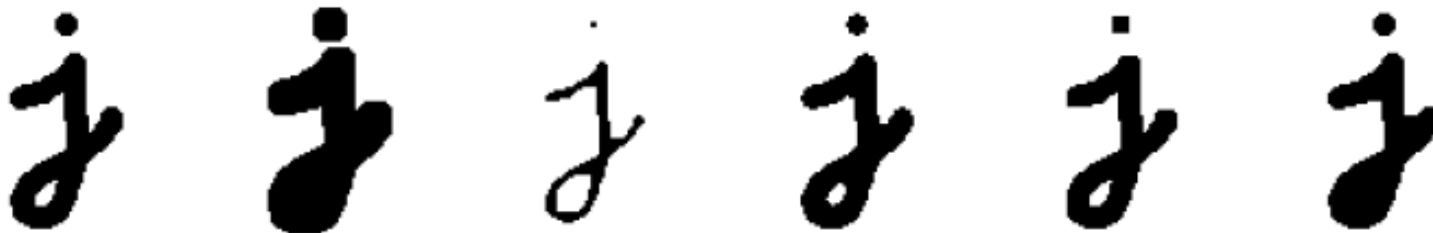
3) Operadores no lineales

- Filtro promedio descartando percentiles menores y mayores
- Filtro bilateral, promediar descartando pixeles que “difieren mucho” dentro de la ventana

3) Operadores no lineales

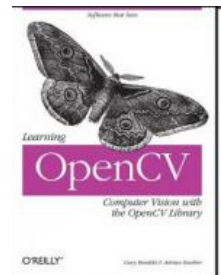
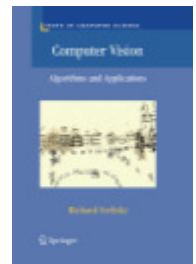
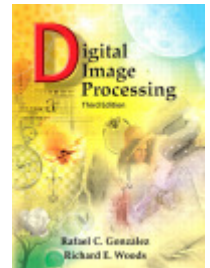
■ Filtros morfológicos:

- Entrada imagen binarizada (0 y 1).
- Structuring element= figura de tamaño S con unos
- Dilation= convolución y binarizar con umbral 1
- Erosion= convolución y binarizar con umbral S
- Majority= convolución y binarizar con umbral $S/2$
- Opening= dilate(erode(imagen))
- Closing= erode(dilate(imagen))



Bibliografía

- **Digital Image Processing.** González et al. 2008.
 - Cap. 2 (intro), 3 (filtros).
- **Computer Vision. Algorithms and Applications.** Szeliski. 2011.
 - Cap. 3.1, 3.2, 3.3 (filtros).
- **Learning OpenCV.** Bradski et al. 2008.
 - Cap. 2.





Pregunta

- ¿Que sucede al hacer convolución con el siguiente kernel?

-1	0	1
-1	0	1
-1	0	1