

Eliminación de Anomalías en una imagen

Juan Humberto Sossa Azuela

!!! GRABAR !!!

Objetivo:

Familiarizarse con las técnicas y herramientas básicas de procesado de imagen.

Después de realizar esta actividad se podrá identificar anomalías (ruido o artefactos) en imágenes.

Se podrán aplicar operaciones para la **reducción** o **eliminación**.

Descripción:

Buscar y seleccionar imágenes con anomalías a corregir.

Describir qué anomalías o **tipología de ruido** afectan a dichas imágenes.

Utilizar operaciones a nivel de pixel, lineales o no lineal o basadas en histograma para resolver dichas anomalías.

La solución se implementará en un notebook Python que describirá y mostrará en pantalla los resultados de los principales pasos.

Descripción:

El notebook deberá constar de las siguientes secciones:

- 1) Descripción** del problema y/o anomalía.
- 2) Al menos 2 imágenes** de ejemplo con la anomalía a eliminar.
- 3) Solución** propuesta.
- 4) Ejecución paso a paso del algoritmo** comentada.

Descripción:

El algoritmo propuesto **no debe ser *ad hoc***, sino que debe de poder **extrapolarse** a otras imágenes **con la misma anomalía**.

Para asegurarse esto, se debe aplicar el mismo algoritmo a al menos 2 imágenes con la anomalía a corregir.

Descripción:

La solución aportada **no debe ser básica**: repetición de una solución bien conocida o existente en una librería.

Se pueden utilizar funcionalidades proporcionadas por las librerías, **pero** la implementación de la operación principal debe ser **propia**.

No se permite copiar código de internet.

En caso de que se reutilicen ideas deberá **referenciarse la fuente**.

Forma de entrega:

Se deben entregar **dos** documentos:

- 1) un notebook Python con la solución propuesta y los **ficheros adicionales que se necesiten**.
- 2) un **informe** donde se explique el tipo de anomalías, métodos empleados para corregirlas y resultados. Se puede **incluir** bibliografía.

NOTA: Extensión máxima del informe: 6 páginas.

Rúbrica:

Criterio	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	La solución es aplicable a otras imágenes con la misma anomalía (no es ad hoc)	2	20%
Criterio 2	La solución aplicada no es básica (bien conocida), la operación principal es propia y elimina las anomalías suficientemente	3	30%
Criterio 3	El notebook ejecuta correctamente y muestra los resultados de los principales pasos.	3	30%
Criterio 4	Código claro y comprensible	1	10%
Criterio 5	Informe claro y sin defectos o carencias	1	10%
Criterio 6	Existe plagio no debidamente referenciado o el trabajo es similar y con los mismos punto de fallo que el de otro grupo	-10	-100%
		10	100 %

Digital Image Processing

4

FOURTH EDITION



Rafael C. Gonzalez
Richard E. Woods

P Pearson

INTELIGENCIA ARTIFICIAL APLICADA A ROBÓTICA Y AUTOMATIZACIÓN

HUMBERTO SOSSA AZUELA
FERNANDO REYES CORTÉS



Paradigmas para el tratamiento digital de imágenes

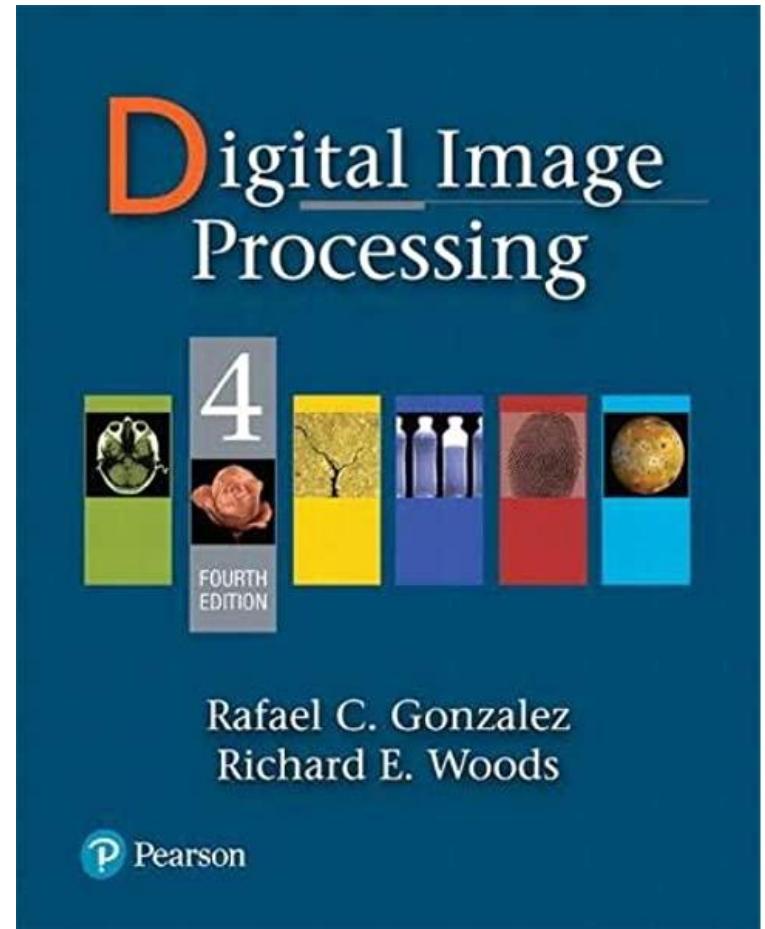
UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS

Adquisición de
imagen

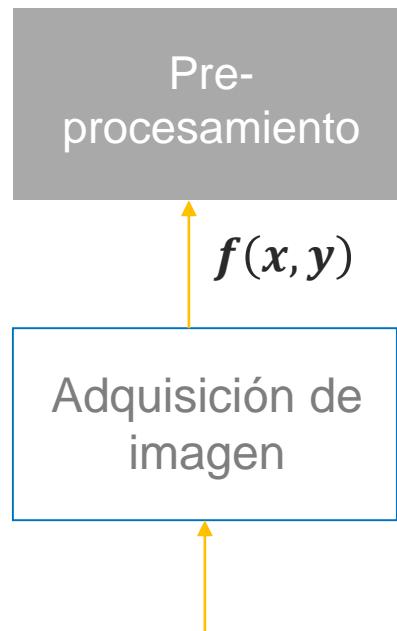


Entrada

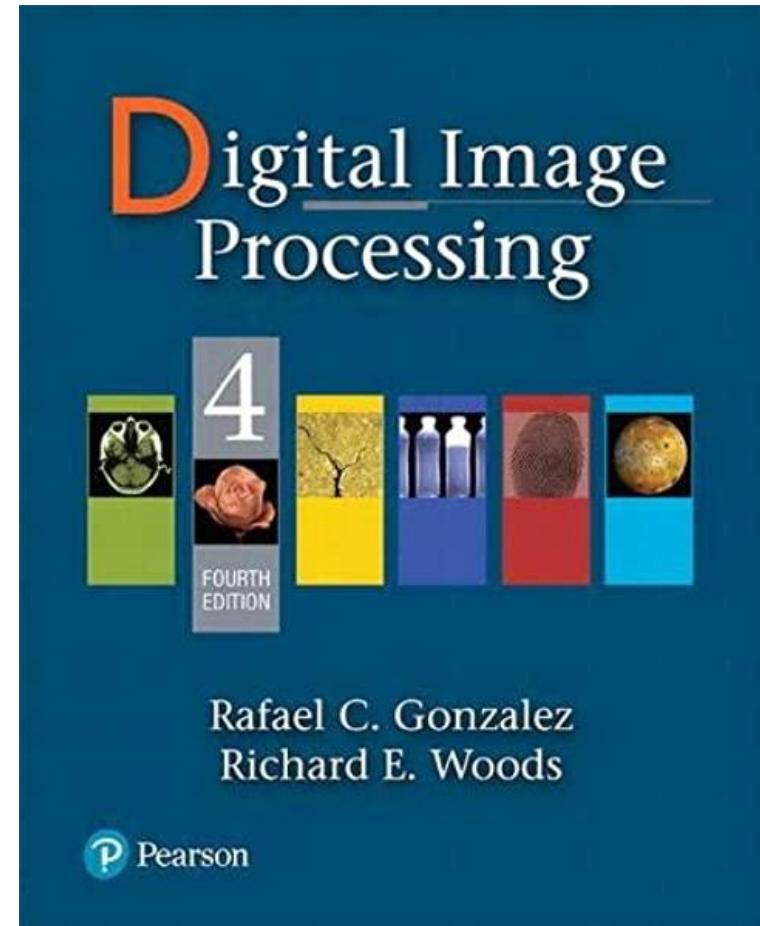
Gonzalez and Woods



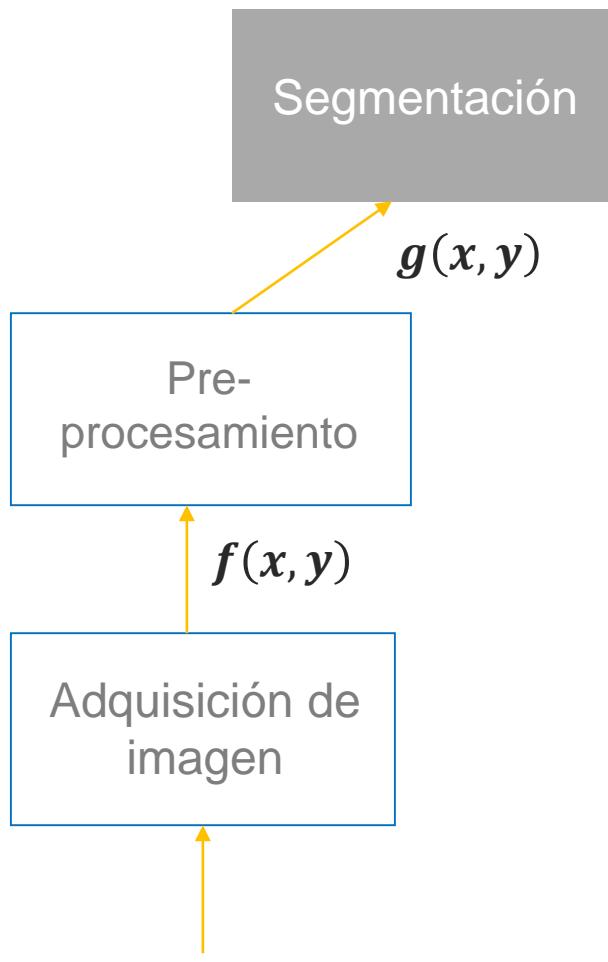
UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



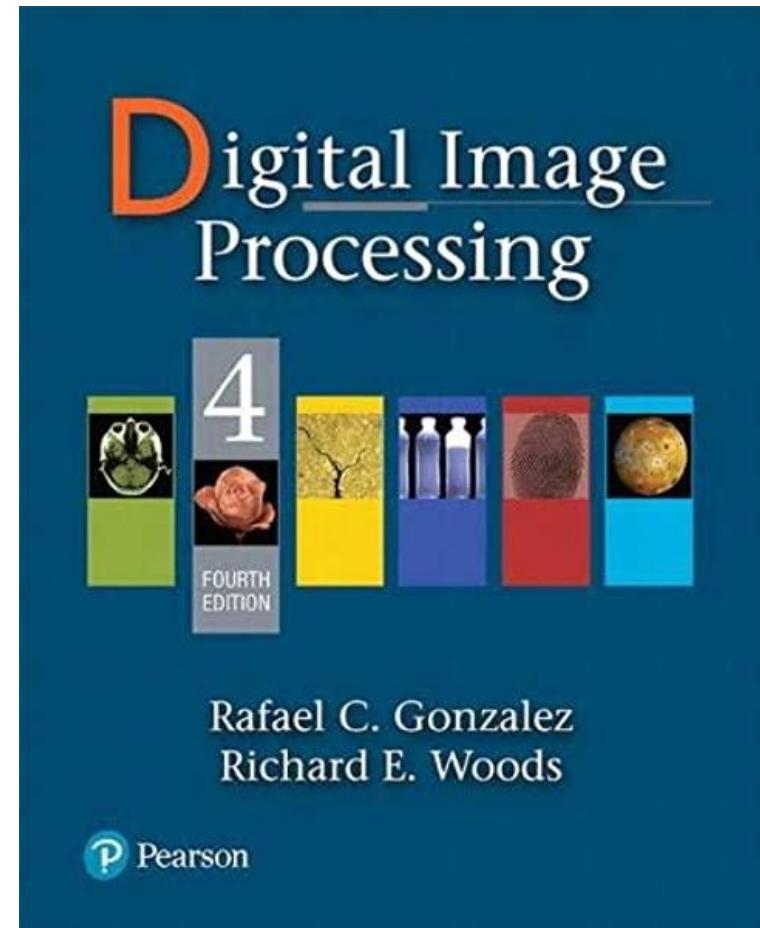
Gonzalez and Woods



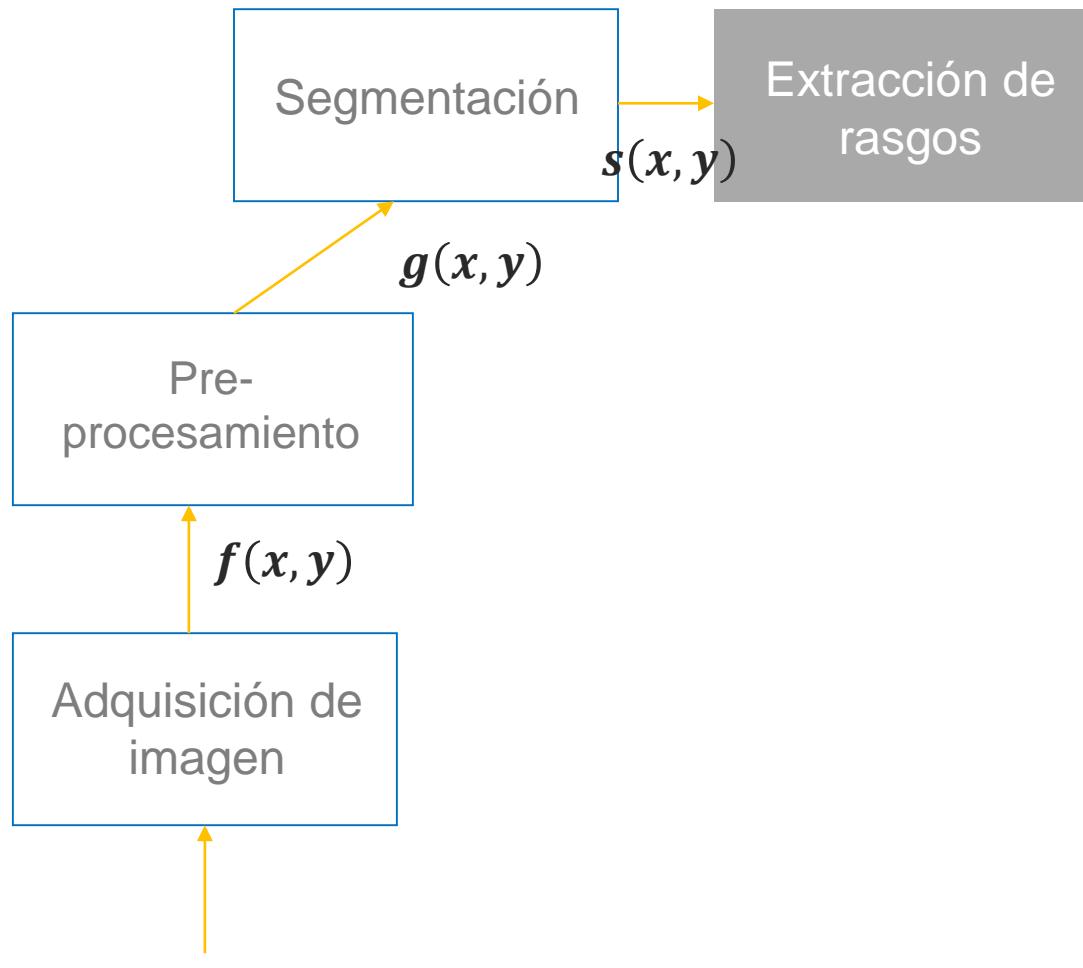
UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



Gonzalez and Woods

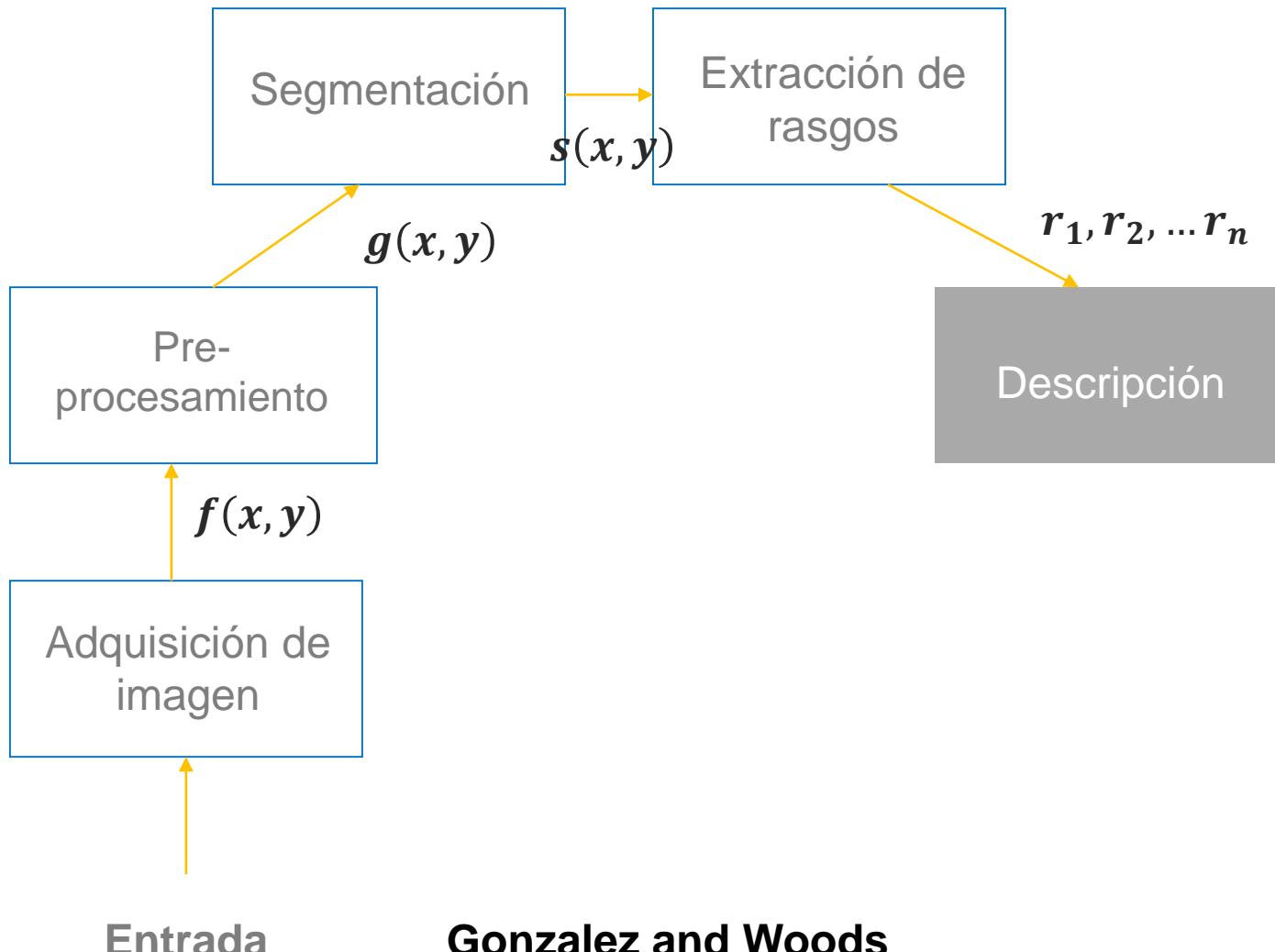


UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS

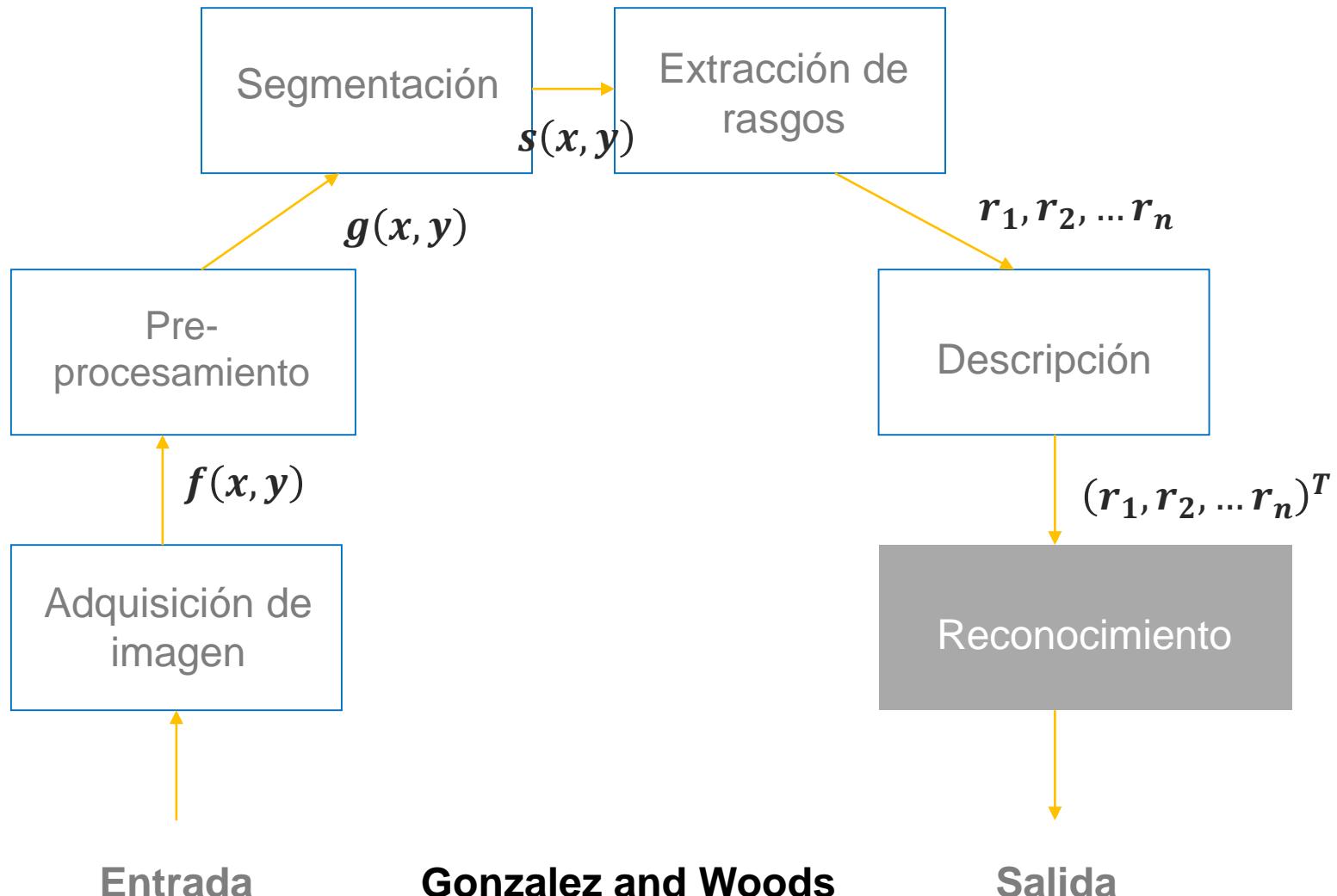


Gonzalez and Woods

UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS

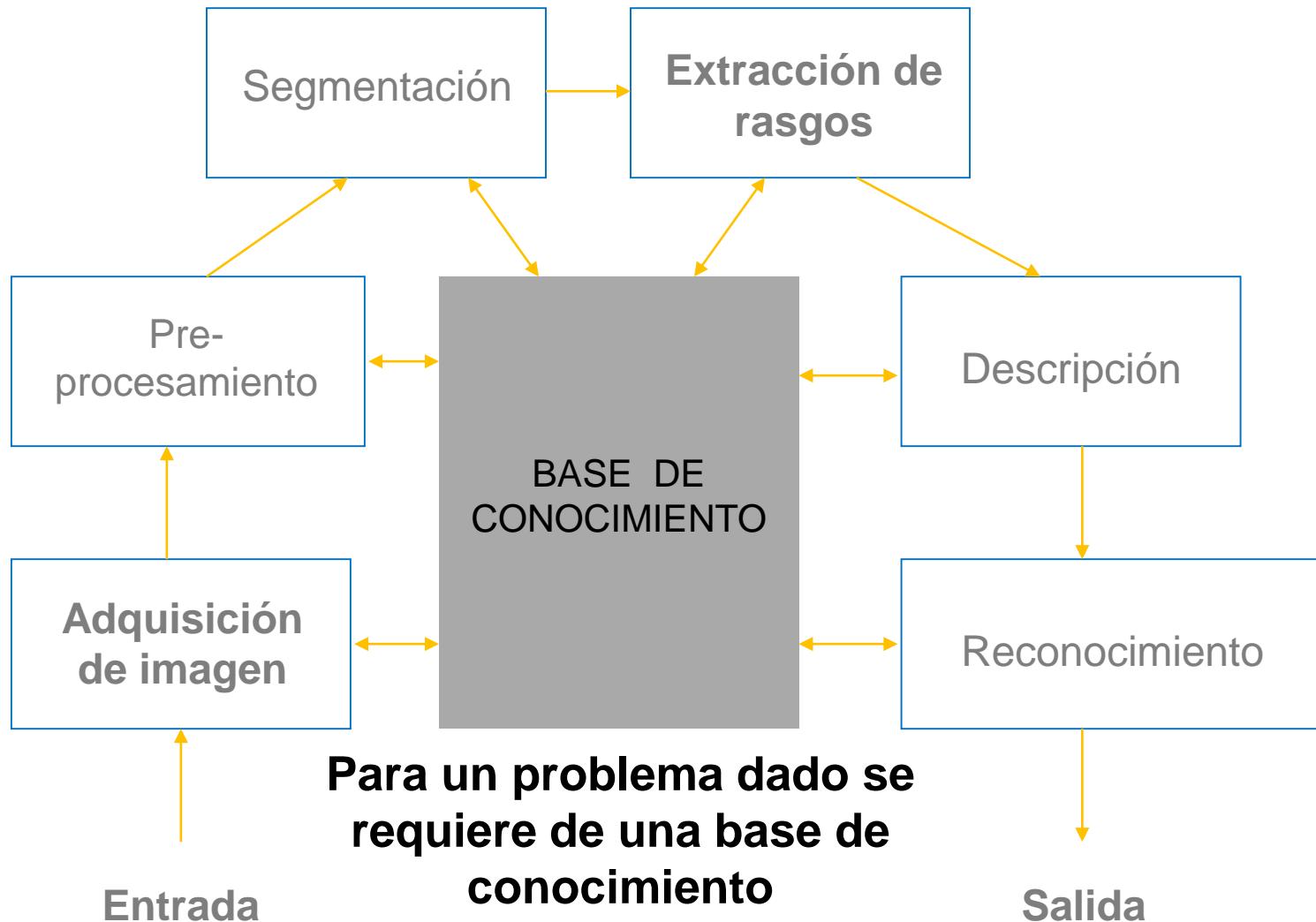


UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS

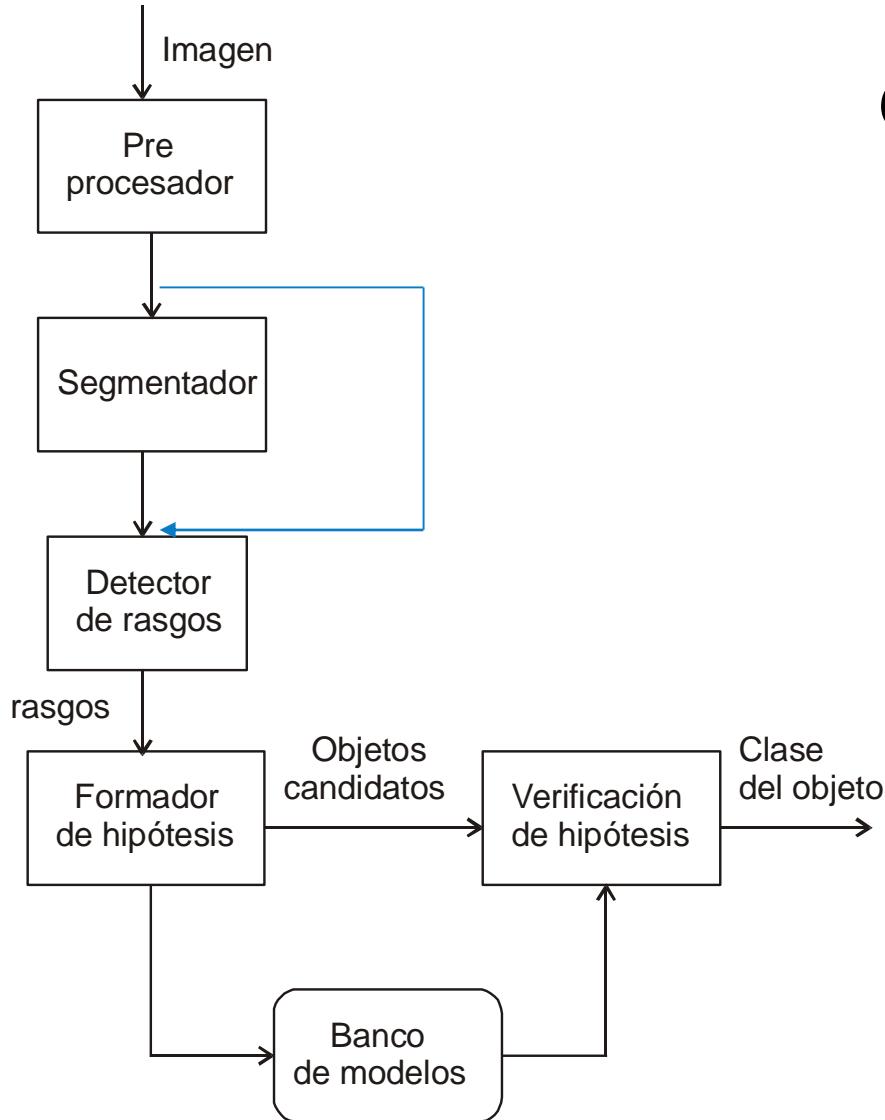


Gonzalez and Woods

UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



OTRO PARADIGMA



Componentes de un sistema para el reconocimiento de objetos:

Este paradigma es útil para problemas que involucran muchas clases de objetos

**1000 clases – 10 clases
(reduce el espacio de búsqueda)**

Verificar si son estas 10

Fundamentos

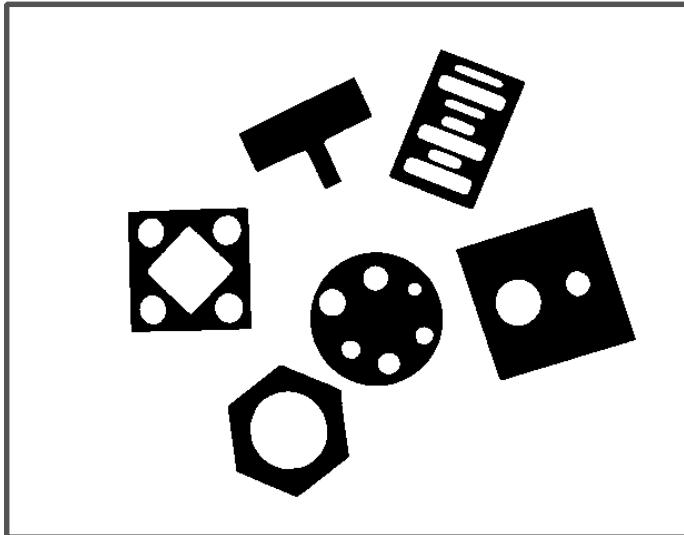
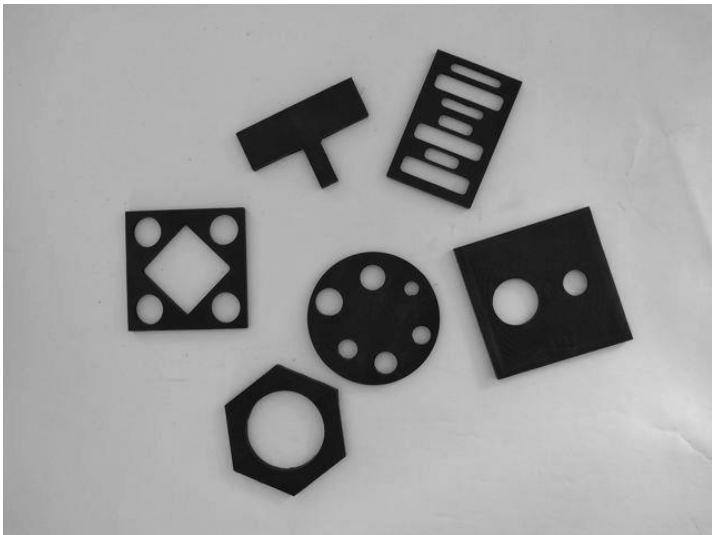
FUNDAMENTOS

IMAGEN DIGITAL E IMAGEN BINARIA

Definición 2.2. Sean Z el conjunto de los enteros. Una imagen digital denotada como $f(x, y)$, es un arreglo bidimensional $\in Z \times Z$ si (x, y) son enteros de $Z \times Z$ y es una f función que asigna a cada par (x, y) un número de Z .

Definición 2.3. Una imagen binaria denotada como $b(x, y)$ es una imagen digital $f(x, y)$ que ha sido cuantizada a dos niveles de intensidad, 0 y 1 por ejemplo.

Ejemplos:



1 2 3 4 5 6

1						
2						
3						
4						
5						
6						

$f(x, y)$



1 2 3 4 5 6

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

1 2 3 4 5 6

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

$b(x, y)$



1 2 3 4 5 6

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

FUNDAMENTOS

AL IGUAL QUE UNA SEÑAL COTÍNUA PUEDE SER DIGITALIZADA PARA OBTENER UNA SEÑAL DIGITAL, UNA IMAGEN DIGITAL PUEDE SER OBTENIDA MEDIANTE UN PROCESO DE DIGITALIZACIÓN:

UNO DE DISCRETIZACIÓN

UNO DE CUANTIZACIÓN

DIGITALIZACIÓN

EL PROCESO DE DIGITALIZACIÓN

Una señal continua $x(t)$ puede ser muestreada mediante un tren de impulsos o función **comb**:

$$p(t) = \sum_{m=-\infty}^{\infty} \delta(t - mT_s) = \sum_{m=-\infty}^{\infty} \delta(t - \frac{m}{f_s})$$

T_s es el periodo de muestreo.

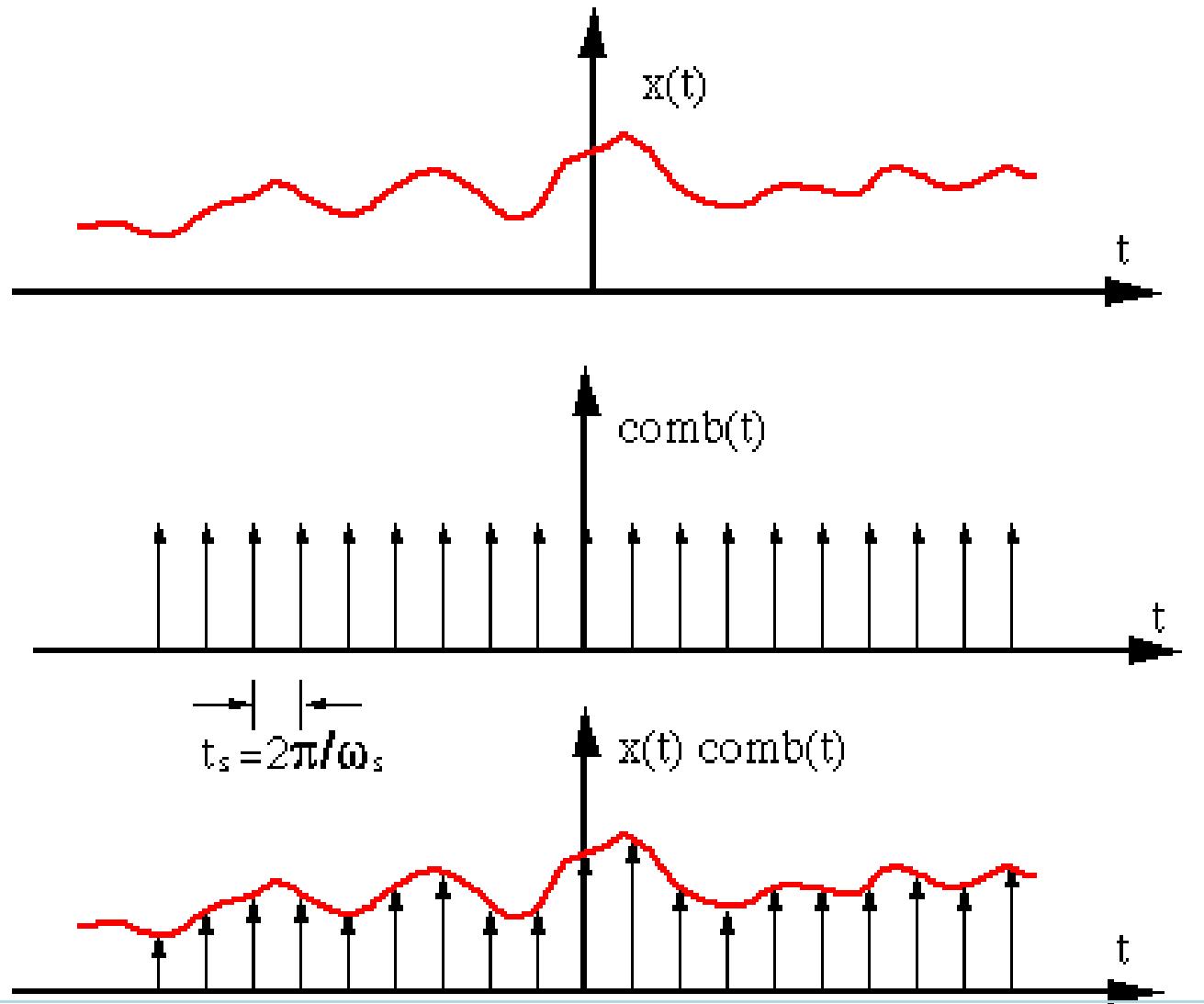
$f_s \triangleq 1/T_s$ tasa de muestreo o frecuencia de muestreo.

El proceso de muestreo viene dado por:

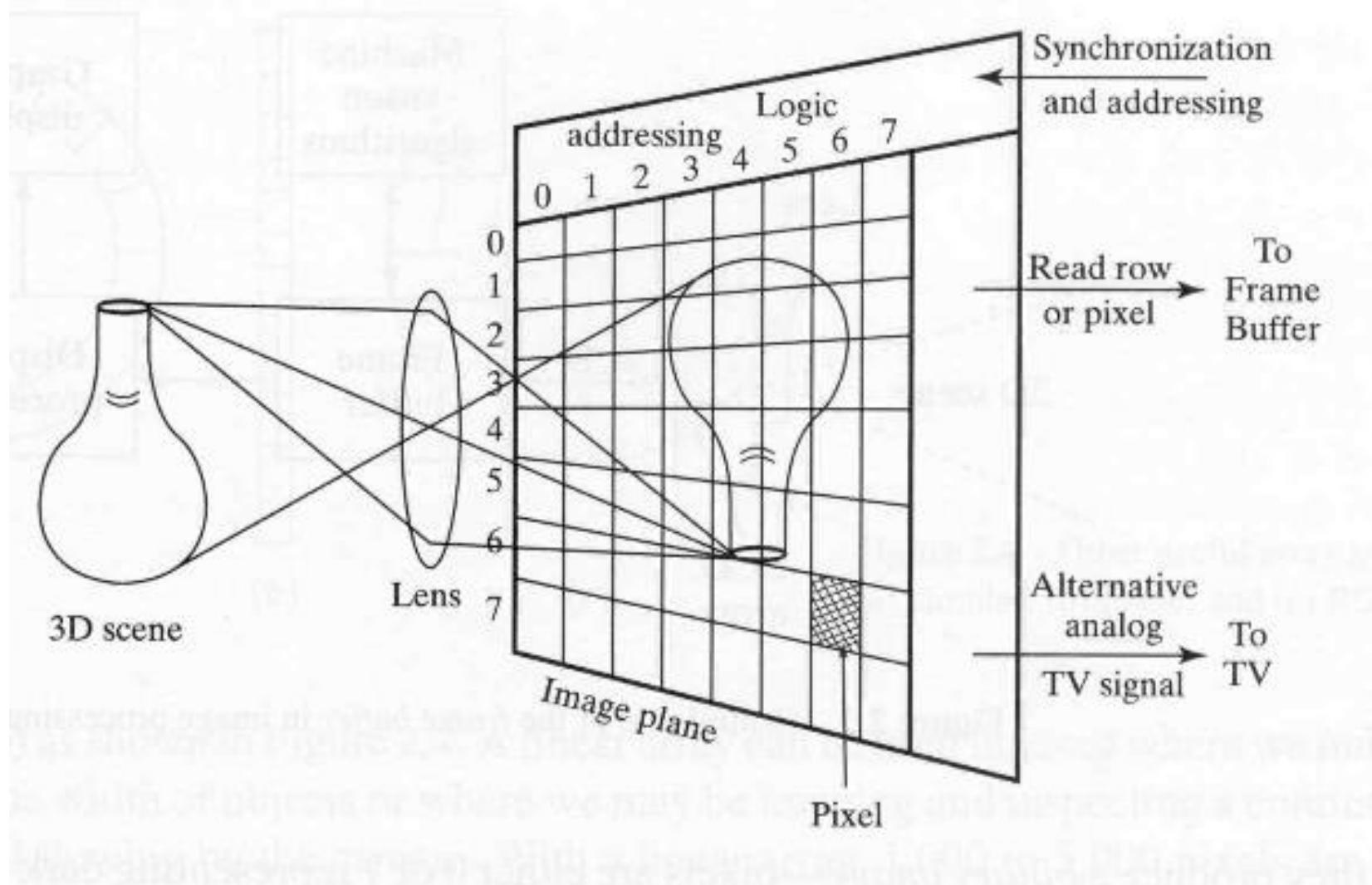
$$x_s(t) = x(t) p(t) = x(t) \sum_{m=-\infty}^{\infty} \delta(t - mT_s) = \sum_{m=-\infty}^{\infty} x(mT_s) \delta(t - mT_s)$$

$x_s(t)$ es el tren de pulsos modulado por la señal $x(t)$.

En una dimensión:

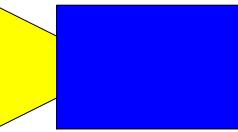
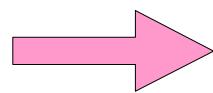
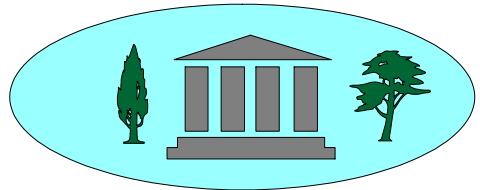


En dos dimensiones:

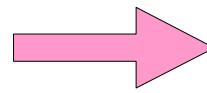


EL PROCESO DE ADQUISICIÓN

ESCENA

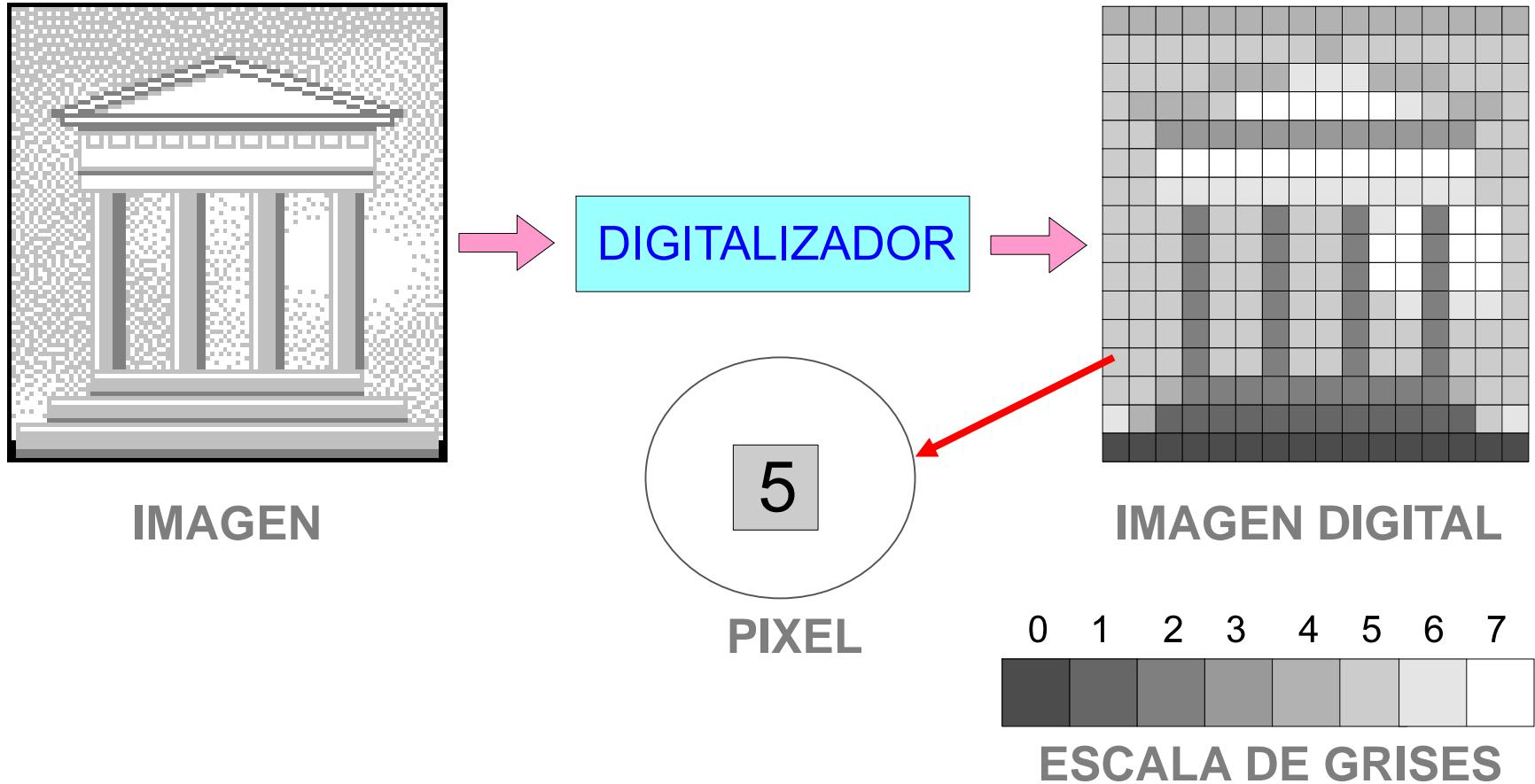


CÁMARA



IMAGEN

EL PROCESO DE DIGITALIZACIÓN



Efecto de la resolución:



(a)



32 X 32

(b)



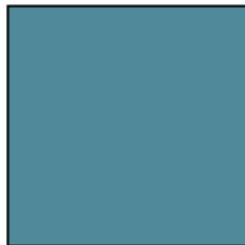
16 X 16

(c)

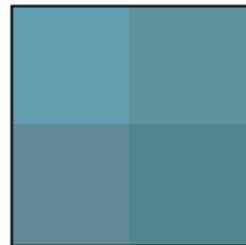
(a) Una imagen monocromática. (b) versión muestreada de (a) a 32×32 muestras, y (c) versión muestreada de (a) a 16×16 muestras.

Efectos del cambio de resolución y de niveles de cuantización:

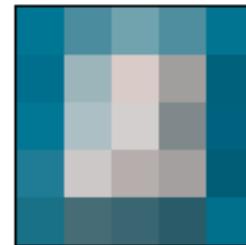
1 x 1



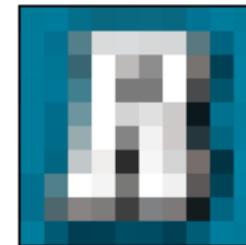
2 x 2



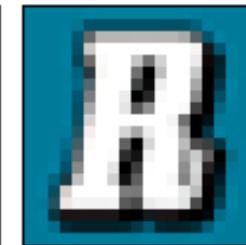
5 x 5



10 x 10



20 x 20



50 x 50



100 x 100



N=64



N=32



N=16



N=8



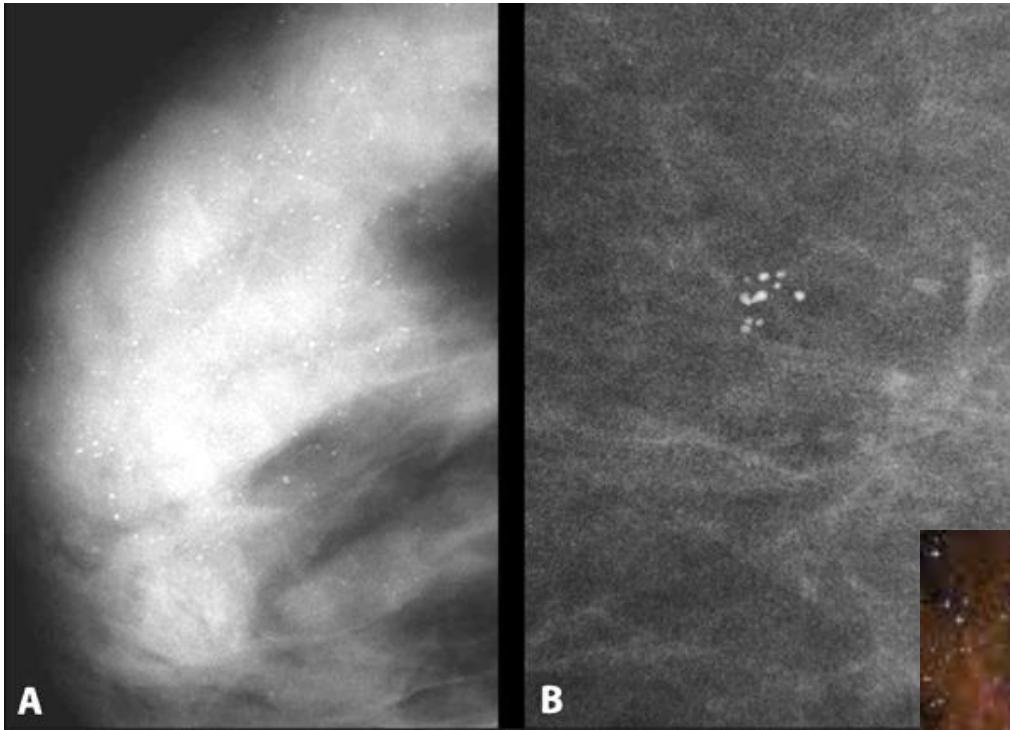
N=4



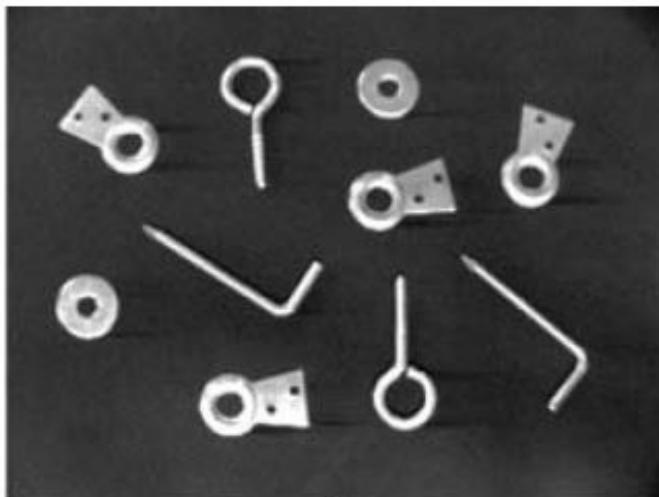
N=2

La calidad de una imagen para una aplicación dada viene dada por el nivel de resolución o discretización de la imagen y el nivel de cuantización:

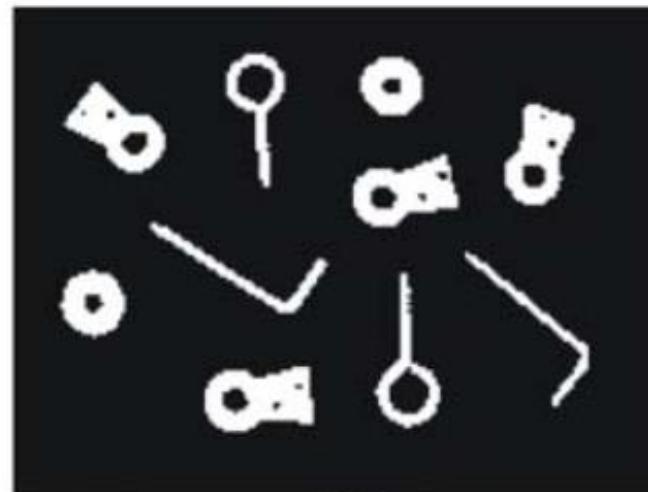




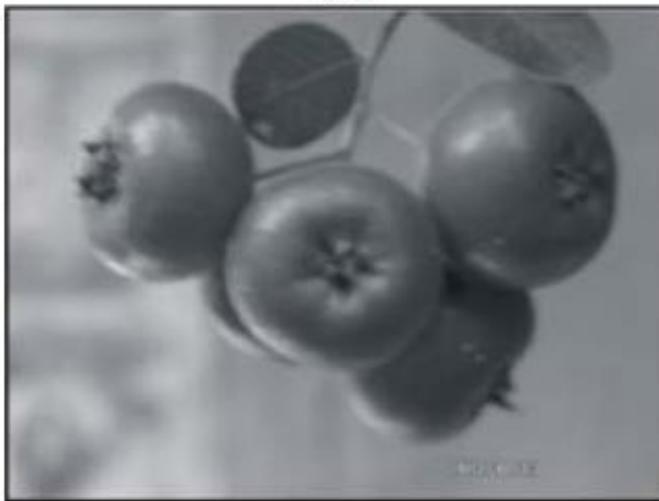
DOS IMÁGENES DIGITALES Y SUS CORRESPONDIENTES IMÁGENES BINARIAS



(a)



(b)



(c)



(d)

Se preservó la forma de los objetos

Se perdió la forma de los objetos

HISTOGRAMA DE UNA IMAGEN

Definición 2.12. El histograma de una imagen $f(x, y)$ con L niveles de intensidad o de gris en el rango $[0, L - 1]$, denotado como $h(r_k)$ es una función discreta

$$h(r_k) = \frac{n_k}{N} \quad (2.2)$$

donde r_k es el k -ésimo nivel de gris, n_k es el número de píxeles en la imagen con el nivel de intensidad r_k y N el número total de píxeles en la imagen.

$$h(r_1) + \cdots + h(r_{L-1}) = 1.0$$

HISTOGRAMA DE UNA IMAGEN

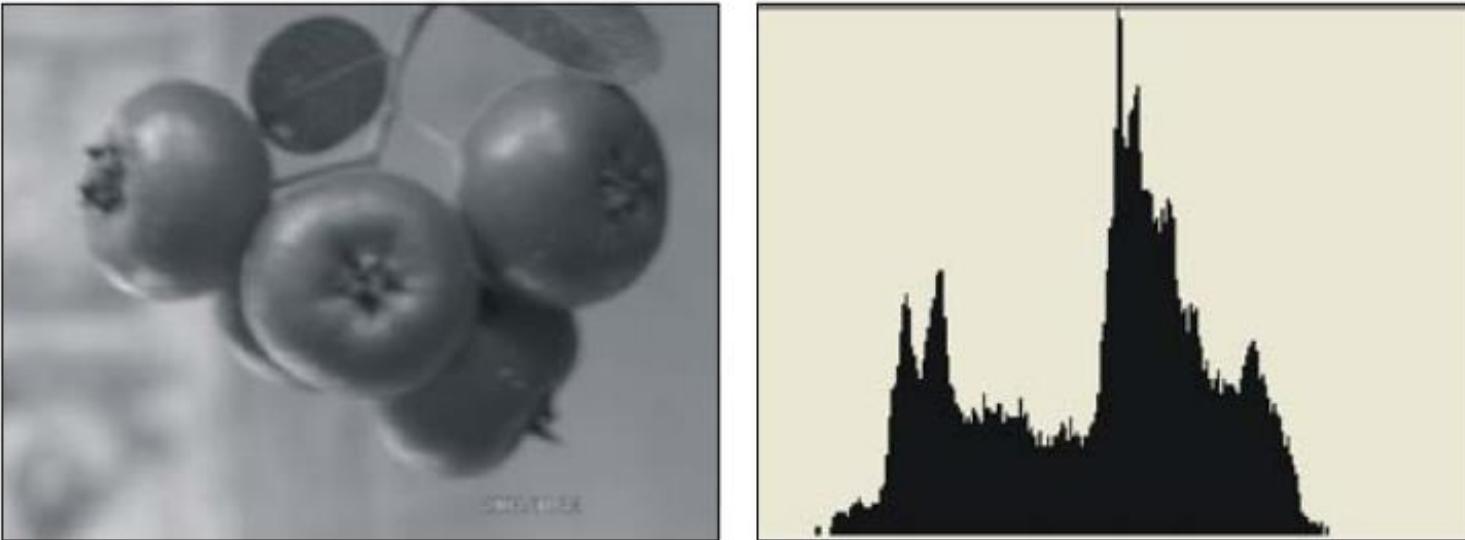


Figura 2.8. Una imagen y su histograma.

En la figura 2.8 se muestra una imagen y su histograma. La función $h(r_k)$ proporciona un estimado de la probabilidad de ocurrencia de un nivel de gris dado n_k . De igual forma, la función $h(r_k)$ también habla de manera global acerca de la apariencia de una imagen.

```
% Programa para obtener el
% histograma de una imagen de entrada f(x,y)

clc; % limpia pantalla
clearvars; %remueve todas las variables.
close all; % cierra gráficas, archivos y recursos abiertos

f=imread('6 objetos.JPG'); % se lee imagen de prueba f(x,y)

figure(1) % se abre una ventana auxiliar.
imshow(f) % se muestra imagen f(x,y) .
title('Imagen original') % se asigna título de esta imagen.

[n,m]=size(f); % dimensiones de la imagen f(x,y)
disp(n); % se muestra valor de n.
disp(m); % se muestra valor de m.
```

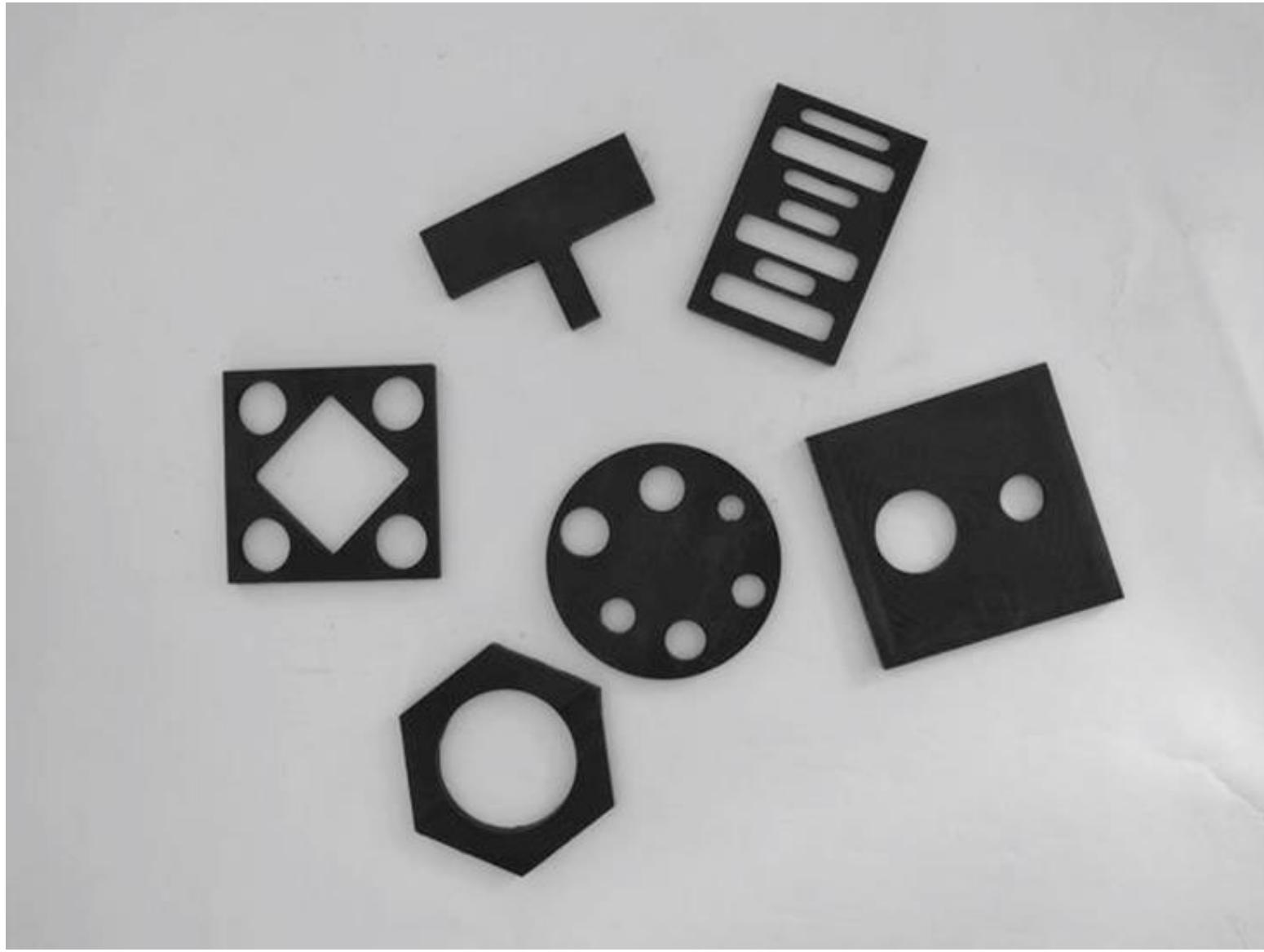


Imagen original

```

c=zeros(256,1); % inicializa en ceros el contador c(i).
niveles=0:255; % niveles de gris.

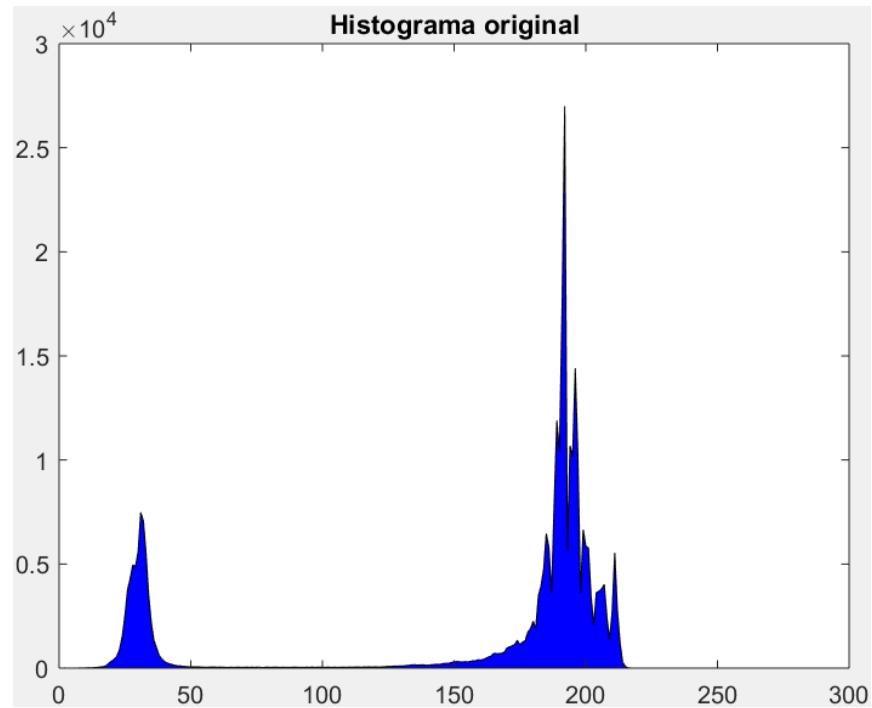
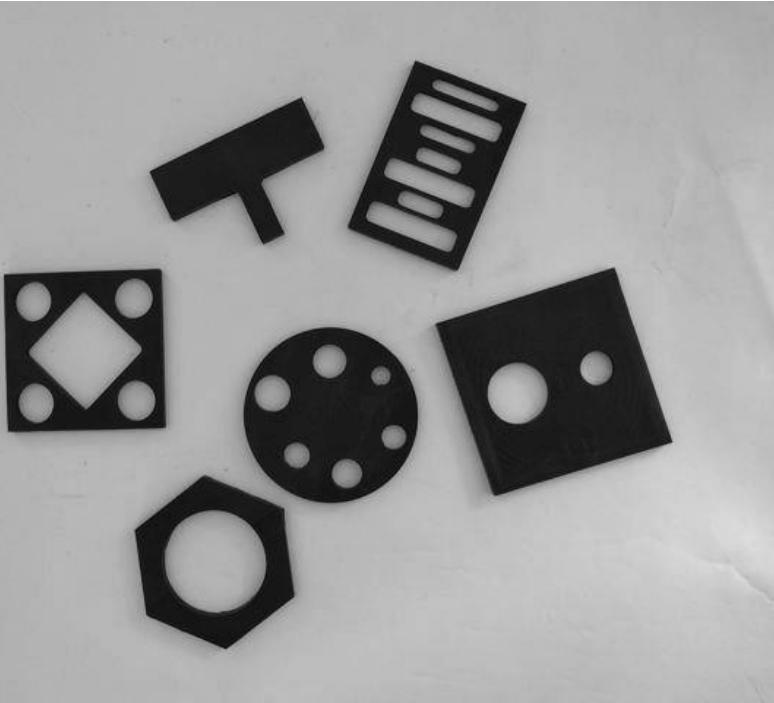
for x=1:n % se barre imagen en renglones.
    for y=1:m % se barre imagen en columnas.
        i=f(x,y); % se obtiene el nivel de gris de f(x; y).
        c(i+1)=c(i+1)+1; % se incrementa contador.
    end
end

figure (2) % se abre una ventana auxiliar.
fill(niveles,c,'b'); % dibuja histograma h(ri) de imagen.
% checar esta función fill, ya que histograma queda movido
hacia arriba sino no
% empieza con 0.
title('Histograma original'); % se asigna título de esta
imagen.
%disp(c);

```

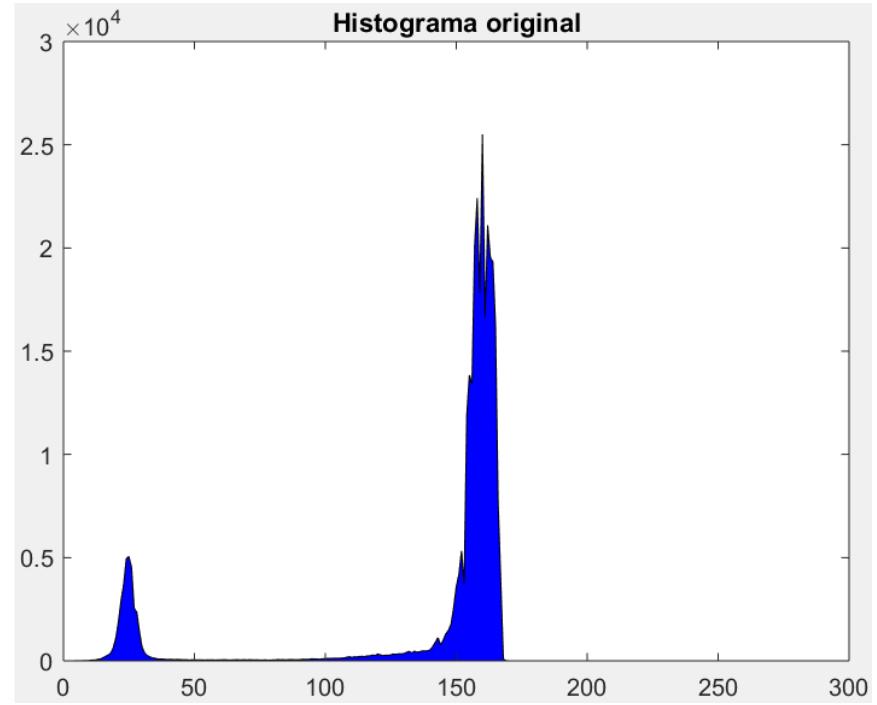
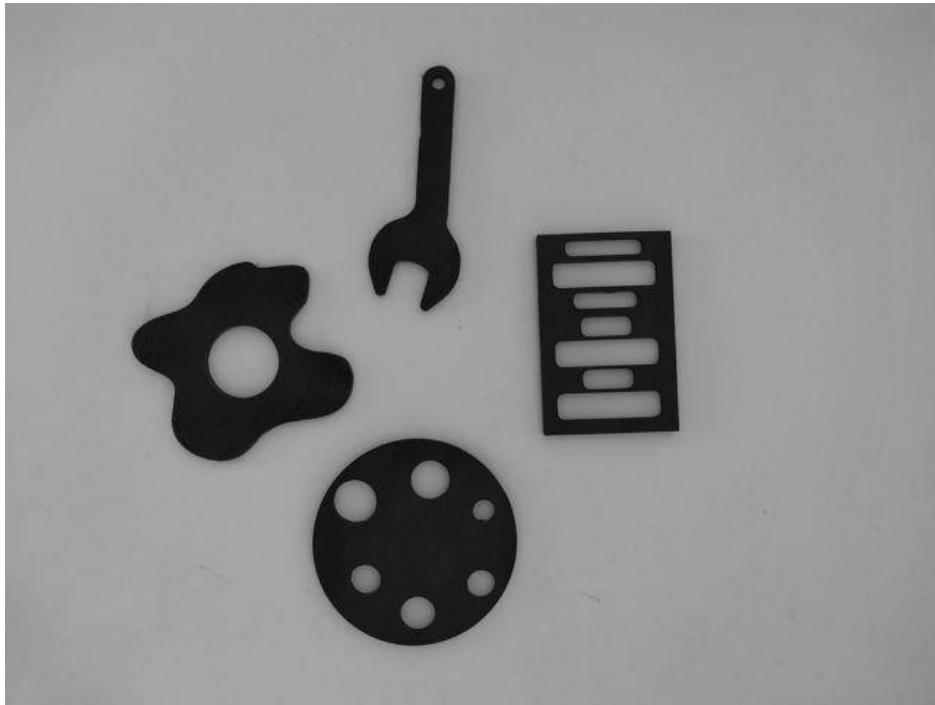
EJEMPLO DE HISTOGRAMA DE UNA IMAGEN:

Dos modas



EJEMPLO DE HISTOGRAMA DE UNA IMAGEN:

Dos modas

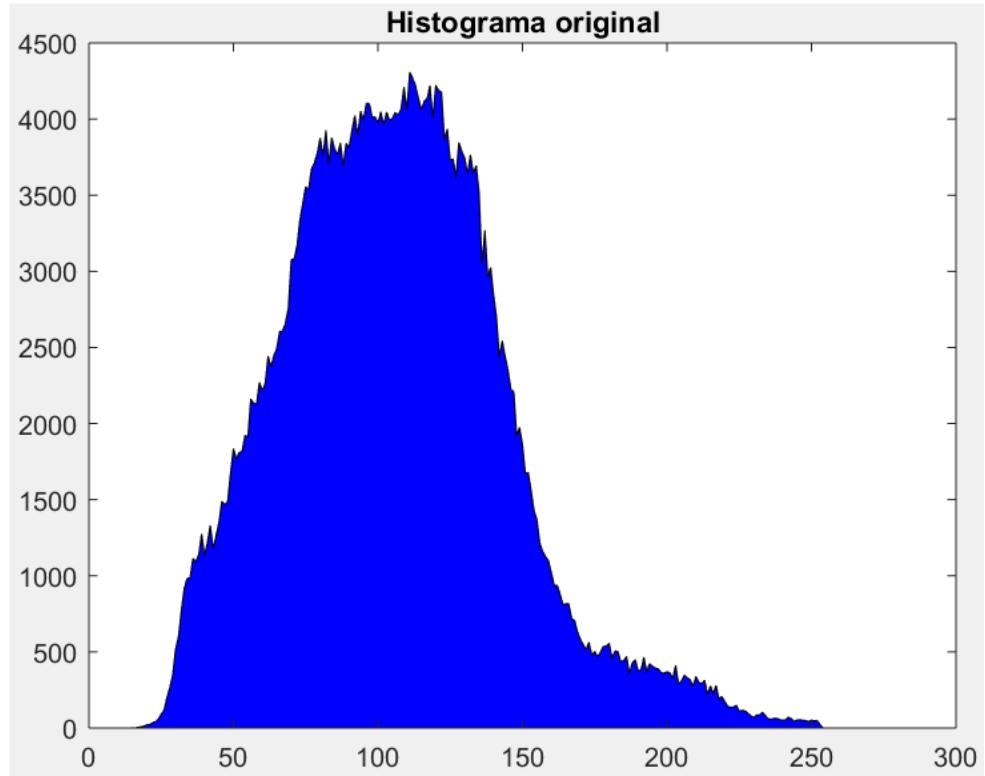


EJEMPLO DE HISTOGRAMA DE UNA IMAGEN:

Una sola moda



Ricky

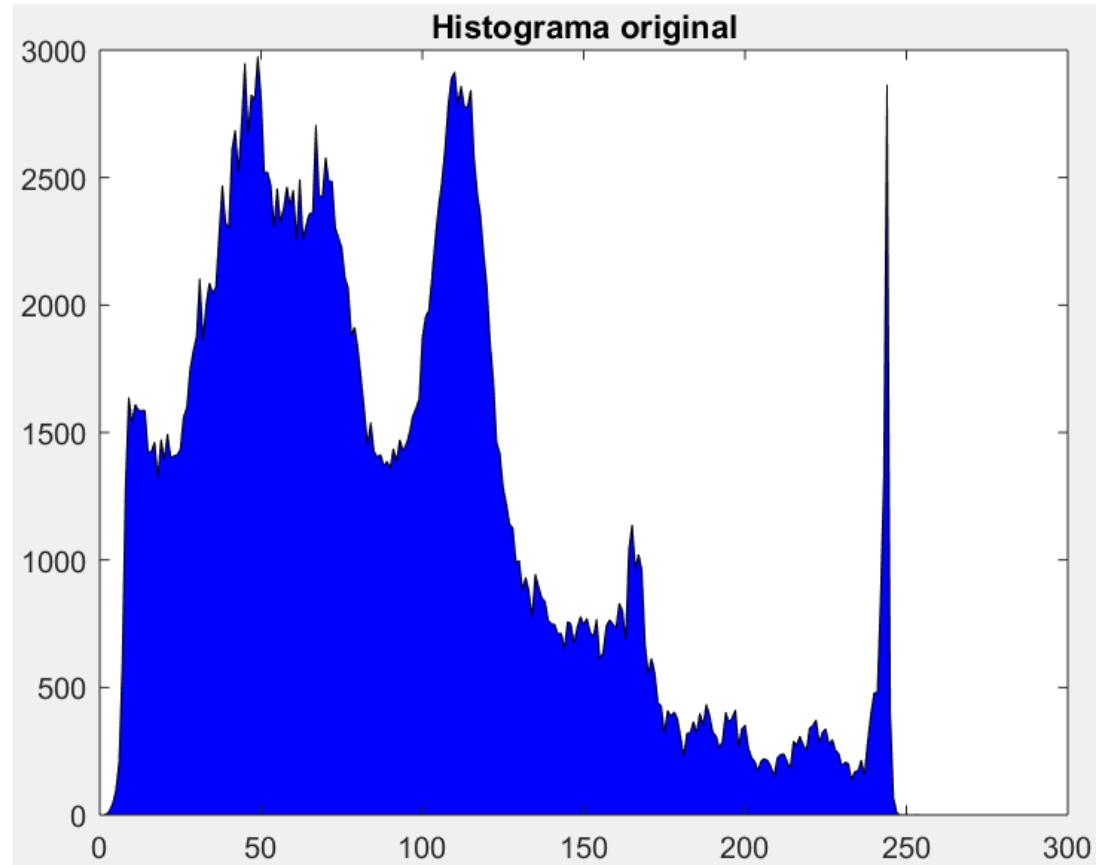


EJEMPLO DE HISTOGRAMA DE UNA IMAGEN:

Varias modas

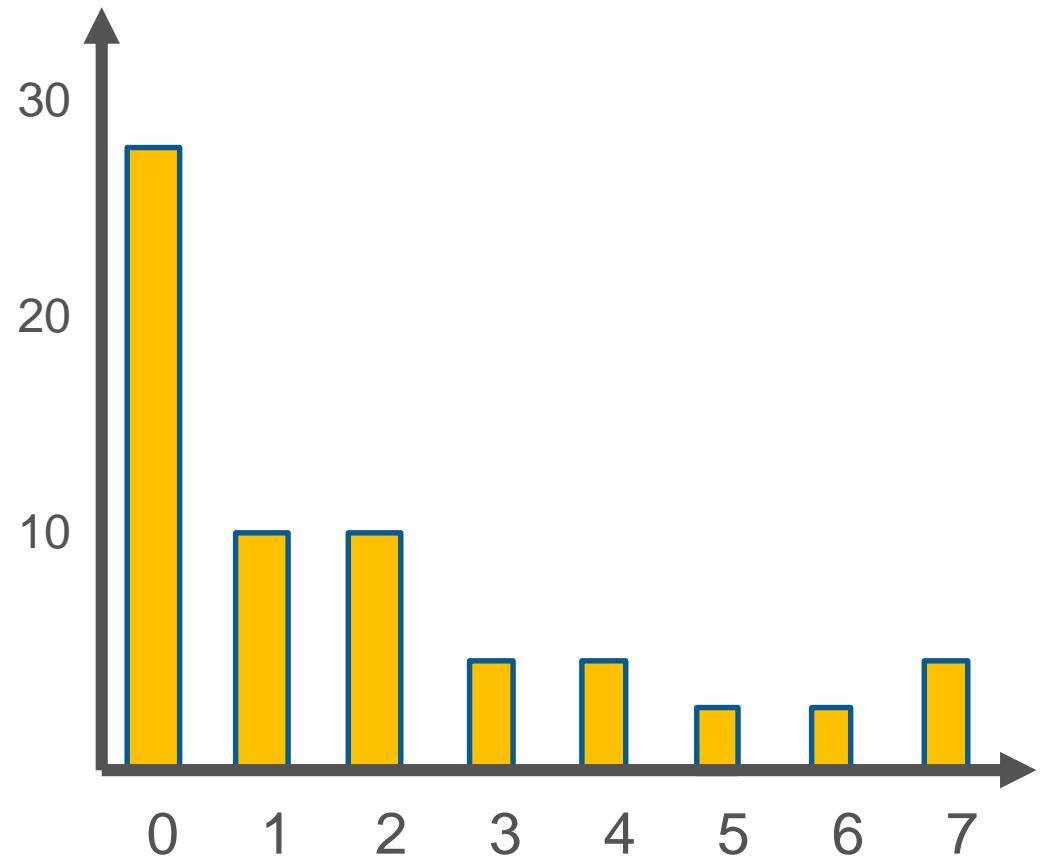


Cleo



Ejemplo: $L = 8$

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

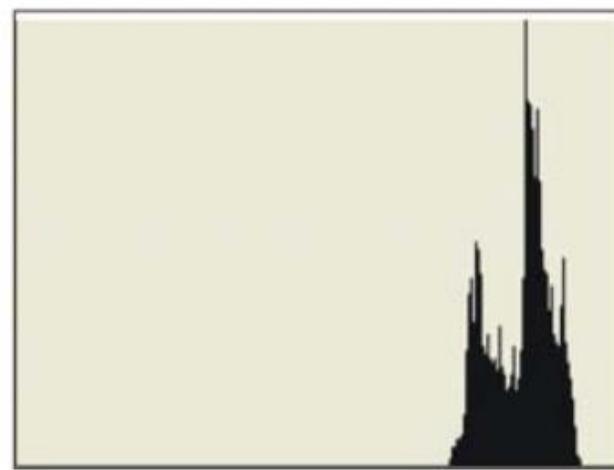


La suma de los tamaños de todas las barras es igual a N
Si el histograma está normalizado, la suma es igual a 1.0

HISTOGRAMA DE UNA IMAGEN



(a)



(b)

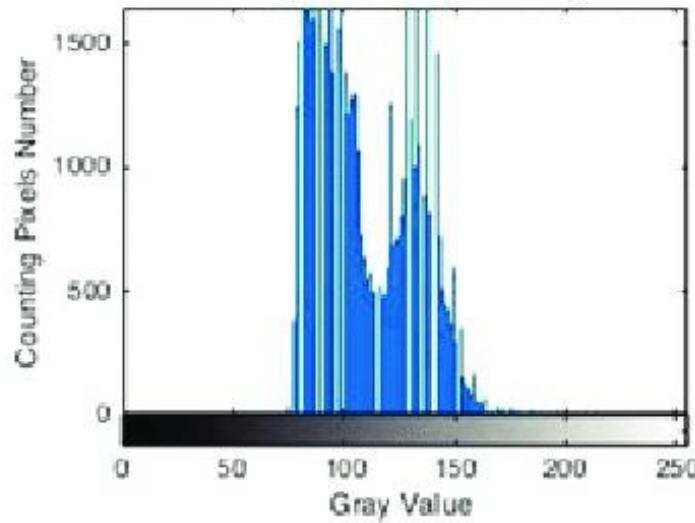
Figura 2.9. (a) Una imagen poco contrastada, obscura y su histograma. (b) Una imagen con poco contraste, clara y su histograma.

Ecualización de una imagen

Original Image



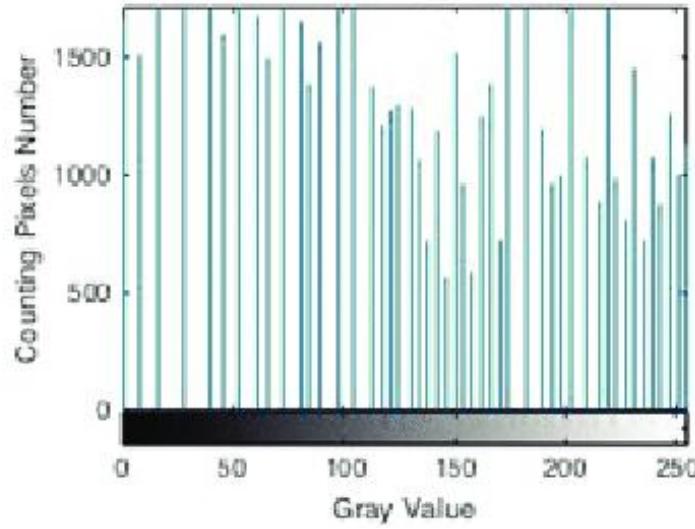
Histogram of Original Image



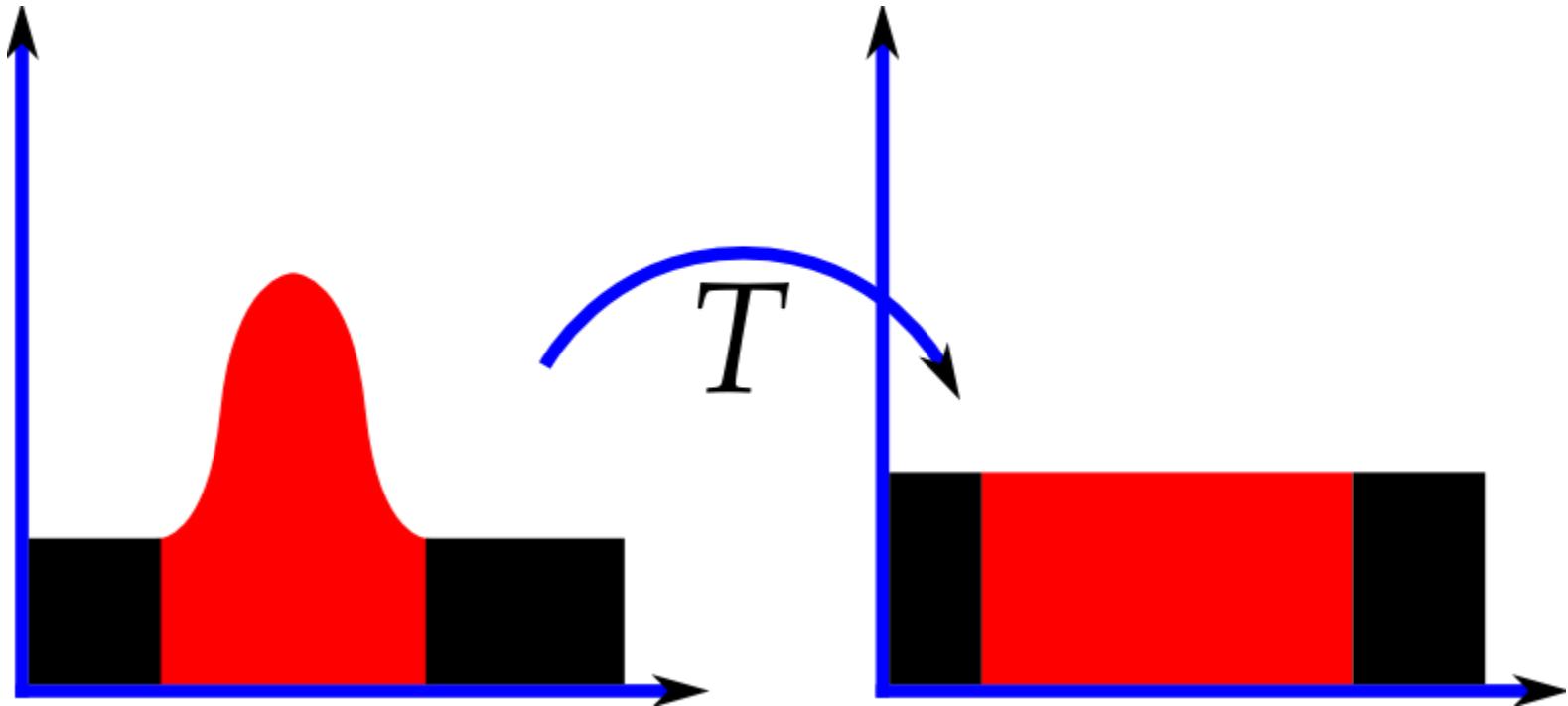
New Image



Histogram of New Image



Ecualización de una imagen

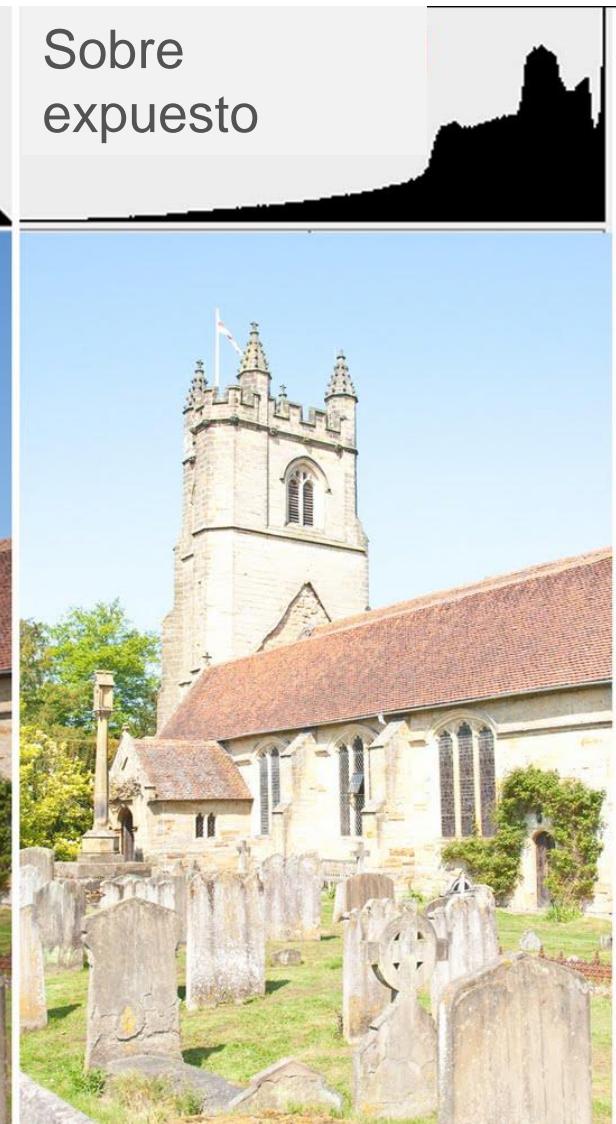


Otros ejemplos:

Bajo expuesto

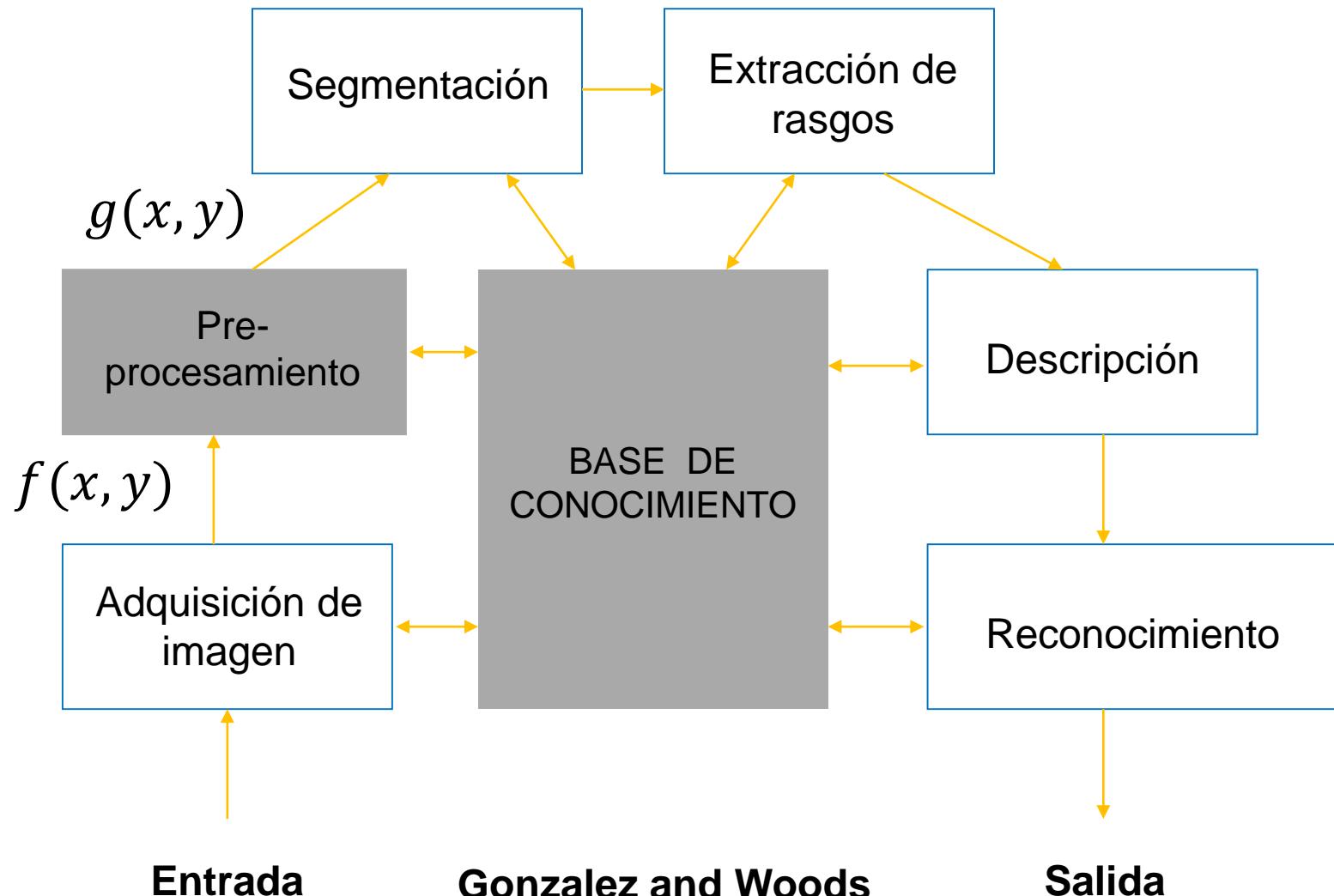


Sobre expuesto

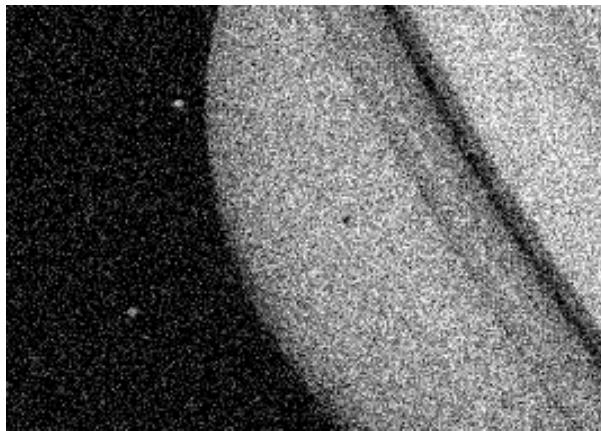


Pretratamiento o acondicionado de una imagen

UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



ACONDICIONADO DE UNA IMAGEN



ACONDICIONADO DE UNA IMAGEN

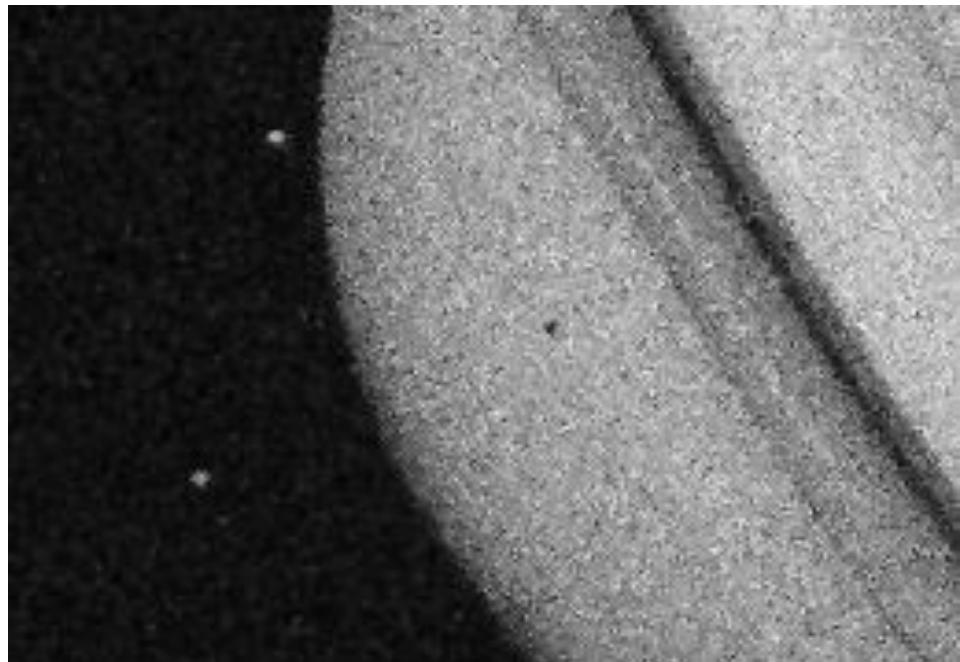
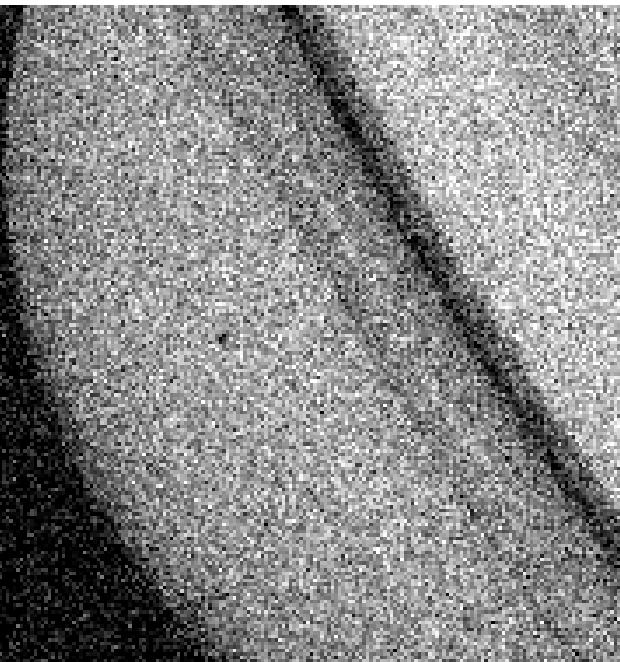
PREPROCESAMIENTO: PROCESAR UNA IMAGEN DE ENTRADA DE TAL FORMA QUE LA NUEVA IMAGEN TENGA UNA MEJOR APARIENCIA VISUAL O PERMITA A UN SISTEMA PARA EL RECONOCIMIENTO DE OBJETOS ENTREGAR MEJORES RESULTADOS.

ACONDICIONADO DE UNA IMAGEN: PROCESAR UNA IMAGEN DE ENTRADA CON EL OBJETIVO DE QUE LOS SIGUIENTES MÓDULOS DE UN SISTEMA PARA EL ANÁLISIS DE IMÁGENES OFREZCAN MEJORES RESULTADOS.

ACONDICIONADO DE UNA IMAGEN



ACONDICIONADO DE UNA IMAGEN



FILTRADO

RUIDO EN IMÁGENES



$$f(x, y)$$

- La óptica y electrónica asociadas al sistema de adquisición,
- Las condiciones de iluminación,
- Las sombras creadas entre objetos,
- Las condiciones atmosféricas,
- Etcétera.



RUIDO EN IMÁGENES

El ruido en una imagen es cualquier entidad o resultado intermedio no interesante para los propósitos del cómputo principal. Pudiera definirse el ruido también como todo aquello que tiende a entorpecer (en cualquiera de sus etapas) el desempeño del SRAO.

$$\tilde{f} = f + r$$

RUIDO DE ADQUISICIÓN

estadístico. Para esto se adquieren n imágenes f_1, f_2, \dots, f_n de la misma escena en las mismas condiciones de iluminación calculando para cada $x, y = 1, \dots, N$:

$$\bar{f}(x, y) = \frac{1}{n} \sum_{k=1}^n f_k(x, y) \quad (3.1)$$

$$\sigma(x, y) = \left(\frac{1}{n-1} \sum_{k=1}^n (\bar{f}(x, y) - f_k(x, y))^2 \right)^{\frac{1}{2}} \quad (3.2)$$

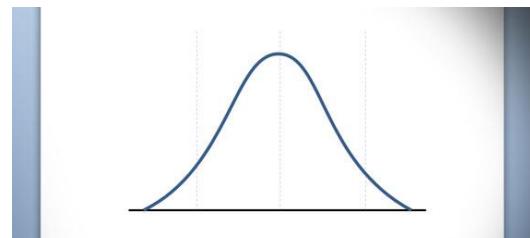
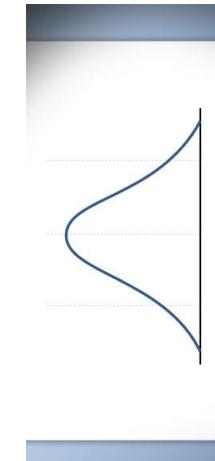
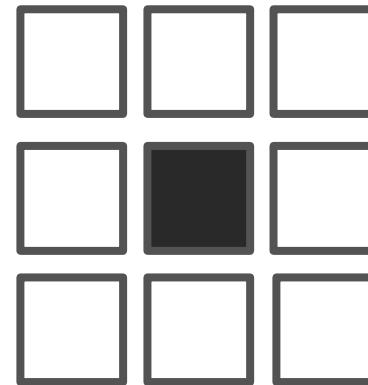
Cualquiera de las imágenes la podemos como una variable aleatoria.

La imagen $\sigma(x, y)$ debería ser oscura.

Si tomamos todos los valores de intensidad de $\sigma(x, y)$, los elevamos al cuadrado y los sumamos, obtendremos un valor superior a cero.

RUIDO DE ADQUISICIÓN

Entrecruce de información



OTRAS CAUSAS DEL RUIDO

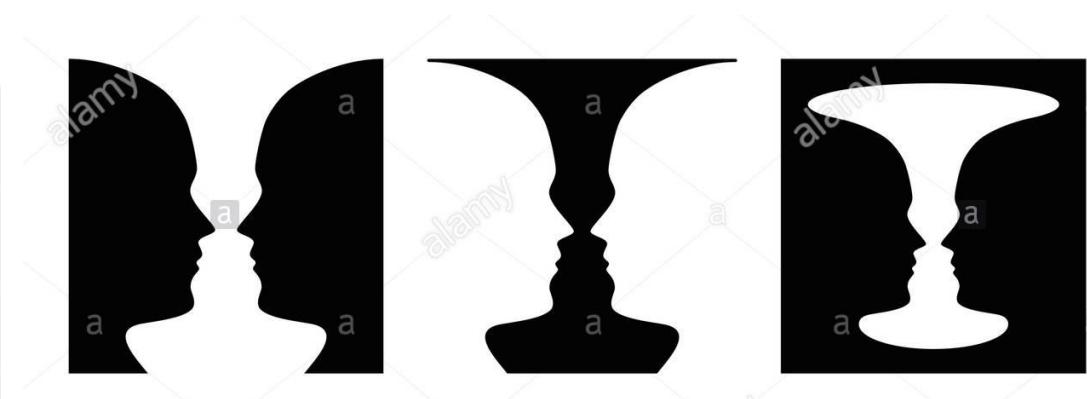
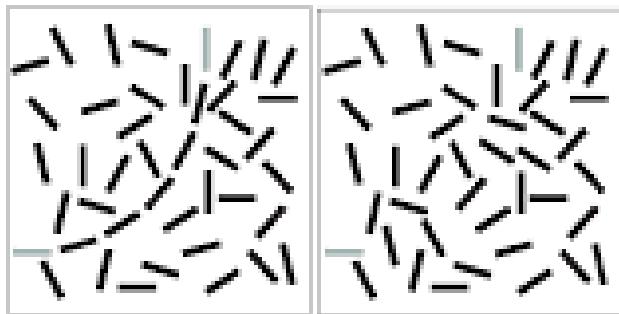
- 1) En detección de bordes o líneas, el ruido puede consistir en fluctuaciones espurias de los niveles de gris de los píxeles introducidas por el sistema de adquisición.
- 2) En el caso de algoritmos recibiendo como entrada resultados de cómputo numéricos intermedios, el ruido puede deberse a fluctuaciones aleatorias o inexactitudes en los datos, precisión limitada del equipo de cómputo usado o redondeo.
- 3) En operaciones de agrupamiento perceptual, puede tratarse de contornos o líneas que no pertenecen a ningún objeto en la imagen.

OTRAS CAUSAS DEL RUIDO

a

<i>Similarity (color)</i>	
<i>Similarity (shape)</i>	
<i>Proximity</i>	
<i>Uniform connectedness</i>	
<i>Element connectedness</i>	

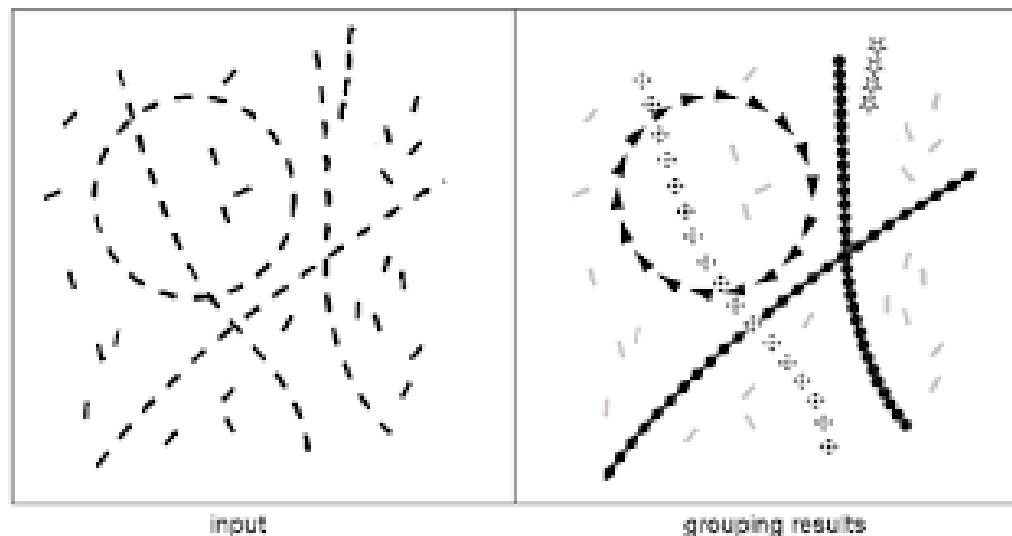
b



a alamy stock photo

PA6P1Y

www.alamy.com



TIPOS DE RUIDO

1. **Ruido aditivo.** $\tilde{f}(x, y)$ es obtenida a partir de $f(x, y)$, en este caso, al adicionar al azar a cada píxel de $f(x, y)$, un real c , tal que $\tilde{f}(x, y) = f(x, y) + c$.
2. **Ruido subtractivo.** $\tilde{f}(x, y)$ es obtenida a partir de $f(x, y)$, en este caso, al restar al azar a cada píxel de $f(x, y)$ un real c , tal que $\tilde{f}(x, y) = f(x, y) - c$.
3. **Ruido mezclado.** $\tilde{f}(x, y)$ es obtenida a partir de $f(x, y)$, en este caso, al sumar o restar al azar a cada pixel de $f(x, y)$ un real c , tal que $\tilde{f}(x, y) = f(x, y) + c$ (ruido aditivo) o $\tilde{f}(x, y) = f(x, y) - c$ (ruido subtractivo).

MANERAS DE GENERAR IMÁGENES CON RUIDO

Una manera de generar una versión ruidosa $\tilde{f}(x, y)$ de una imagen $f(x, y)$, es la siguiente:

$$\tilde{f}(x, y) = \begin{cases} f(x, y) & a < l \\ f(x, y) + b \cdot c & a \geq l \end{cases} \quad (3.3)$$

donde $f(x, y)$ es la imagen original, $a \in [0,1]$ es una variable aleatoria uniformemente distribuida, l es el parámetro que controla cuanto del patrón $f(x, y)$ es contaminado y c que tan severo es el ruido. Cuando $b=1$ el ruido adherido al patrón es de tipo aditivo. Inversamente, cuando $b = -1$ ruido adherido al patrón es de tipo subtractivo.

MANERAS DE GENERAR IMÁGENES CON RUIDO

Finalmente, cuando para cada $f(x, y)$, $b = 1$ ó $b = -1$, el ruido adherido es mezclado. En el caso de imágenes con valores en el rango $[0, L-1]$, el valor de c ($c \in \mathbf{Z}$) substraído o adicionado a cada píxel será tal que:

$$0 \leq f(x, y) \pm c \leq (L-1) \quad (3.4)$$

TIPOS DE RUIDO

3.2.1. Ruido Gaussiano.

En ausencia de información acerca del ruido presente en una imagen, se asume que dicho ruido sigue un comportamiento Gausiano. En otras palabras la variable c toma valores según la siguiente relación:

$$c = (L - 1) \cdot \exp\left(\frac{-z^2}{2\sigma^2}\right) \quad (3.5)$$

TIPOS DE RUIDO

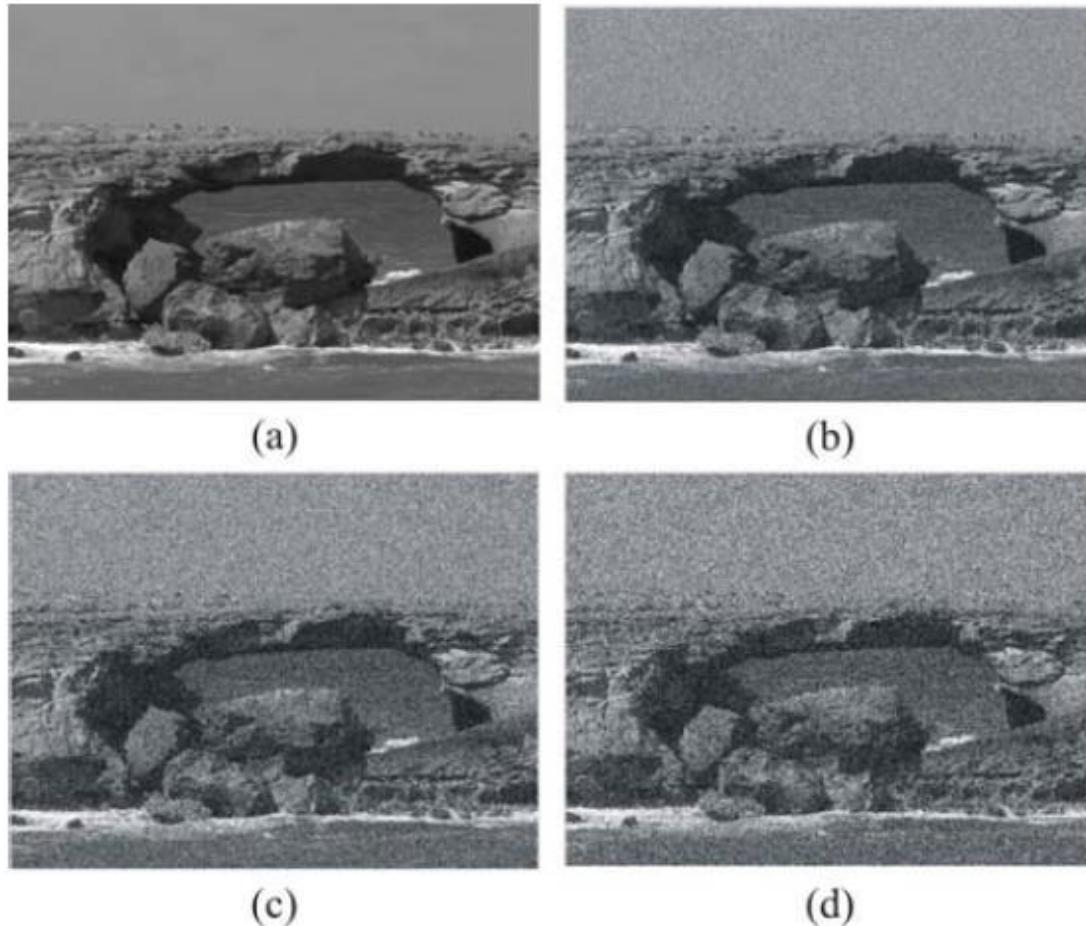


Figura 3.1. (a) Una imagen, y (b-d) tres versiones contaminadas con ruido Gausiano de la misma imagen con $\sigma = 0.01$, $\sigma = 0.03$ y $\sigma = 0.05$.

```
% Programa para agregar ruido Gaussiano a una imagen.  
  
clc % limpia pantalla.  
clearvars; % remueve todas las variables del espacio  
actual de trabajo.  
close all;% cierra gráficas, archivos y recursos abiertos.  
  
f=imread('Gato 01.JPG'); % lee imagen de prueba f(x,y).  
figure(1)  
imshow(f) % muestra imagen de prueba.  
title('Imagen original')  
[n,m]=size(f); % dimensiones de la imagen f(x,y).
```

Imagen original sin ruido:



Imagen original

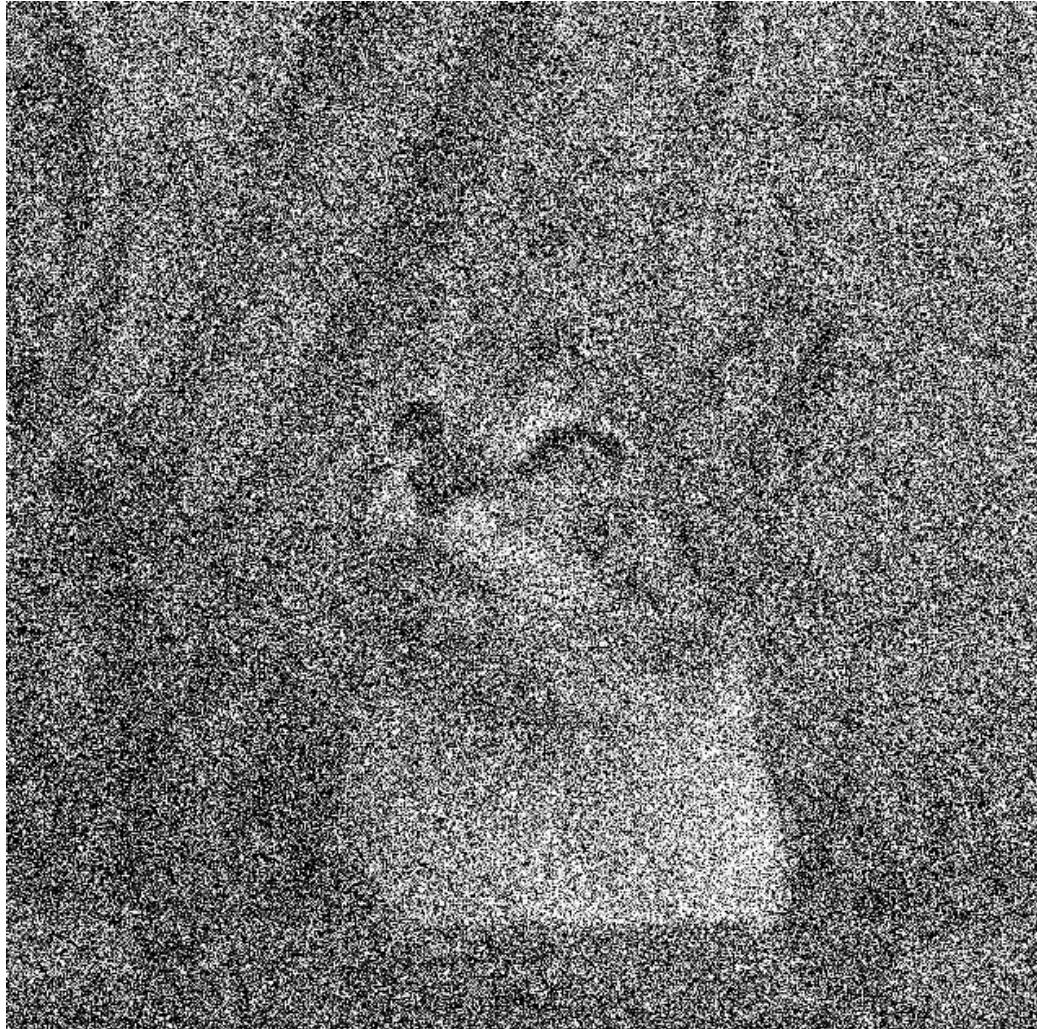
```
fsp = imnoise(f, 'gaussian', 0, 0.2); % permite agregar ruido  
% con una media y una sigma.  
figure(2)  
imshow(fsp) % imagen con ruido.  
title('Imagen con ruido Gaussiano')
```

Imagen con ruido agregado:



$$\sigma = 0.2$$

Imagen con ruido agregado:



$$\sigma = 0.4$$

Imagen con ruido agregado:



$$\sigma = 0.6$$

TIPOS DE RUIDO

3.2.2. Ruido de sal y pimienta.

Un tipo de ruido que algunas veces acompaña a las imágenes de entrada es el llamado *ruido tipo sal y pimienta*. Este tipo de ruido se caracteriza por saturar el valor de una o más píxeles de la imagen hacia el límite positivo (sal) o hacia el límite inferior (pimienta).

TIPOS DE RUIDO



(a)



(b)



(c)

Figura 3.2. (a-c) tres versiones ruidosas con ruido sal y pimienta de la imagen mostrada en la Figura 3.1(a). En este caso $l = 0.02$, $l = 0.07$ y $l = 0.12$.

```
% Programa para agregar ruido sal y pimienta.

clc % limpia pantalla.
clearvars; % remueve todas las variables del espacio actual de trabajo.
close all;% cierra grácas, archivos y recursos abiertos.

f=imread('Gato 01.JPG'); % lee imagen de prueba f(x,y) .
figure(1)
imshow(f) % muestra imagen de prueba.
title('Imagen original')
[n,m]=size(f); % dimensiones de la imagen f(x,y) .
%g=zeros(n,m);
%g=f;
%disp(n); % filas
%disp(m); % columnas
```

Imagen original sin ruido:



Imagen original

```
fsp = imnoise(f,'salt & pepper',0.2);
figure(2)
imshow(fsp) % imagen con ruido.
title('Imagen con ruido sal y pimienta')
```

Imagen con ruido agregado:



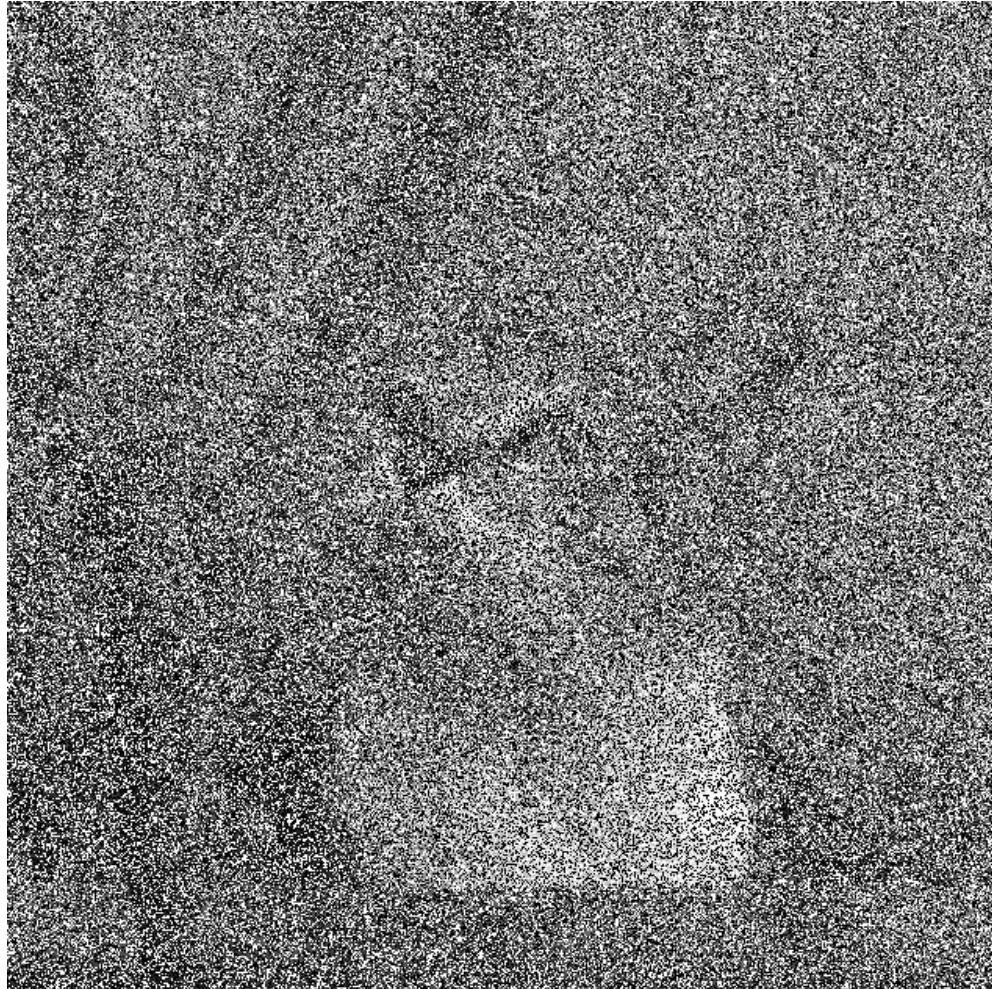
$$l = 0.2$$

Imagen con ruido agregado:



$$l = 0.4$$

Imagen con ruido agregado:



$$l = 0.6$$

Limpieza de una imagen:

En el dominio de la frecuencia.

En el dominio de los píxeles.

Filtrado de el dominio de la frecuencia:

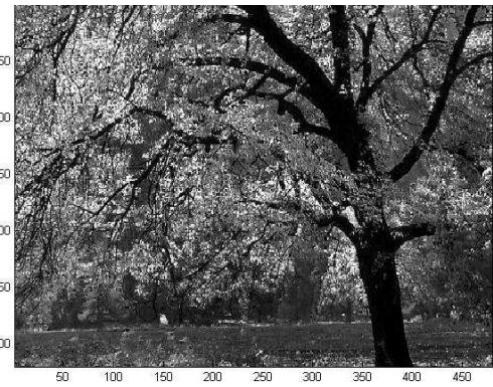
Calcular la transformada de Fourier $F(u,v)$ de $f(x,y)$.

Diseñar filtro $G(u,v)$.

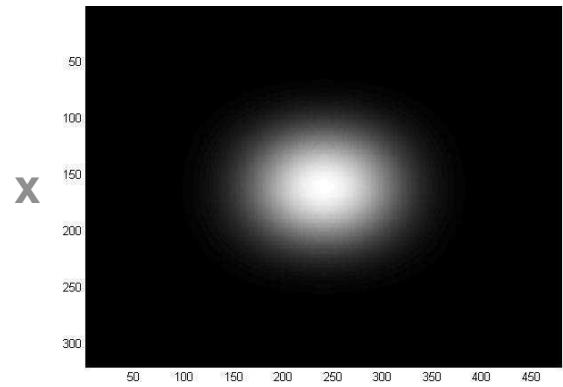
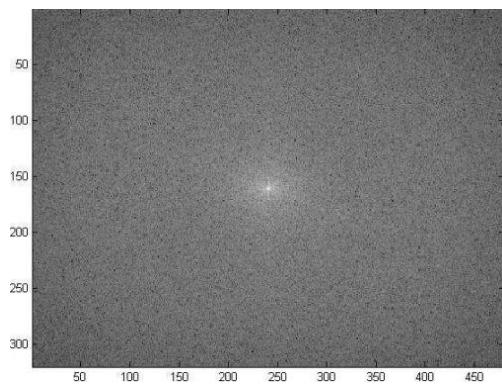
Multiplicar $F(u,v)$ por el filtro $G(u,v)$.

Calcular la transformada de Fourier inversa $f^*(x,y)$.

Obtener la parte real $f(x,y)$.

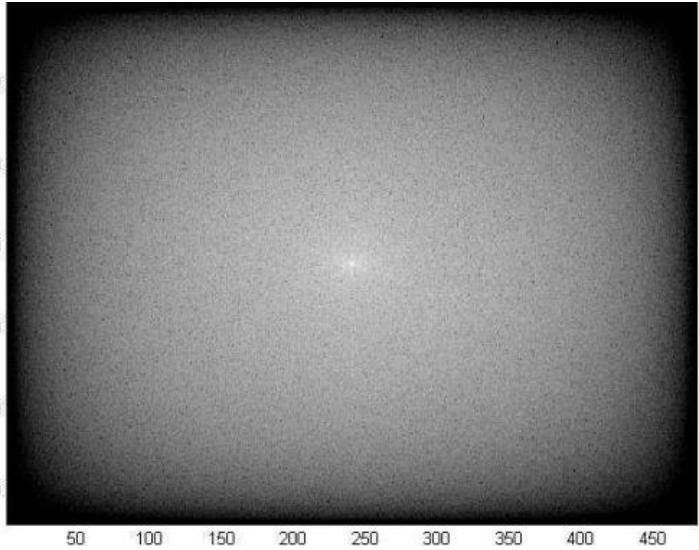


F(u,v)

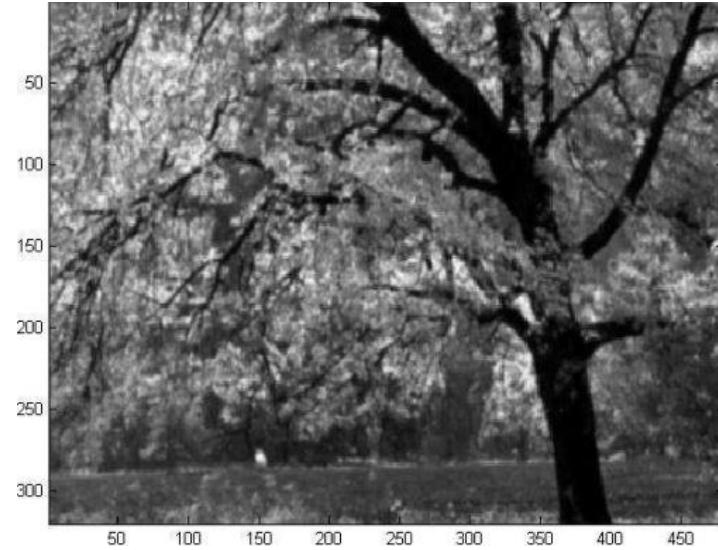


X

=

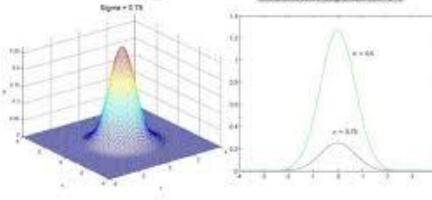


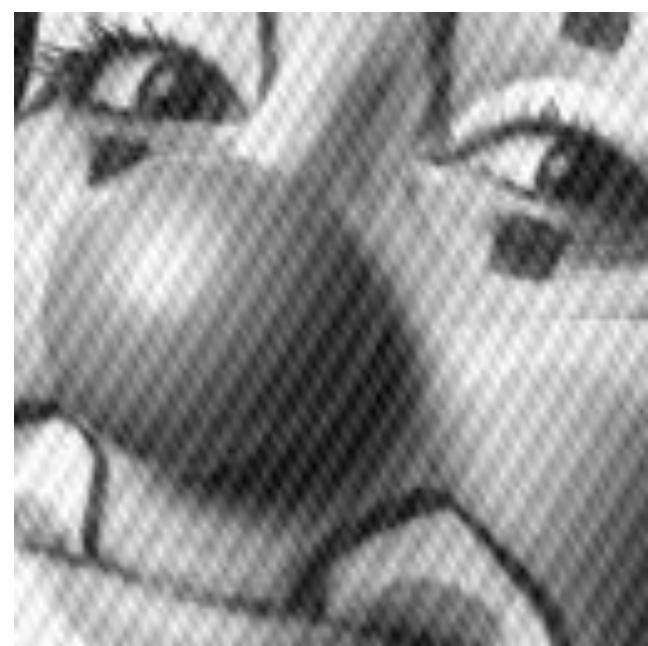
**$f^*(x,y)$ y parte
real**



Filtro Gaussiano (II)

$$w(s,t) = \frac{e^{-\frac{s^2+t^2}{\sigma^2}}}{2\pi\sigma^2}$$



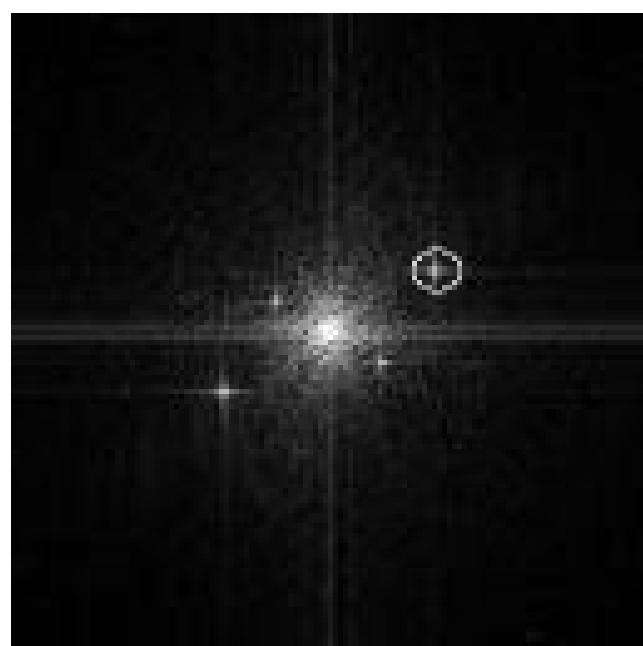


$F(u,v)$

**Filtrado local
de la FFT**

**Picos de ruido
removidos**

**Transformada
inversa**



Aplicación de la FFT:

Filtrado en el dominio frecuencias: Convolución:

Imagen:

$$I[m, n]$$


Transformada de Fourier:

$$F[u, v]$$

Diseño de filtro:

$$H[u, v]$$


Aplicación de filtro:

$$H[u, v] \cdot F[u, v]$$


Transformada de

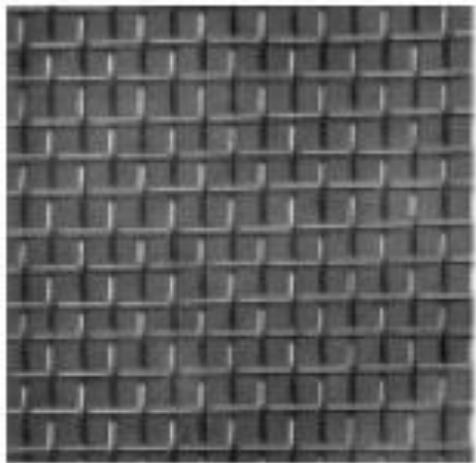
Fourier inversa:

$$\text{FFT}^{-1}\{H[u, v] \cdot F[u, v]\}$$

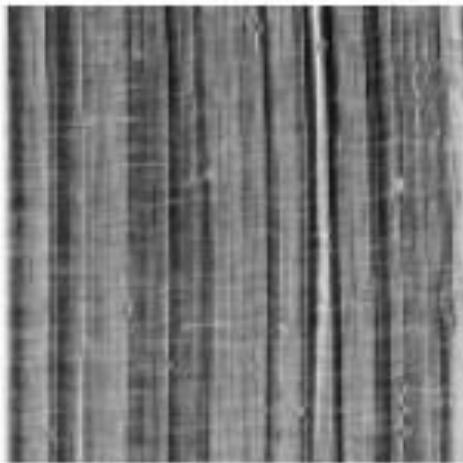

Imagen mejorada:

$$I'[m, n]$$

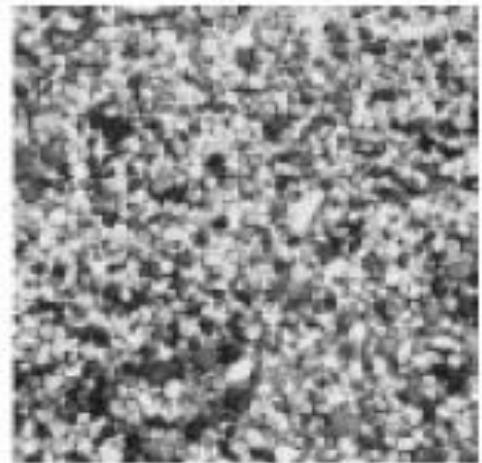
Reconocimiento de texturas:



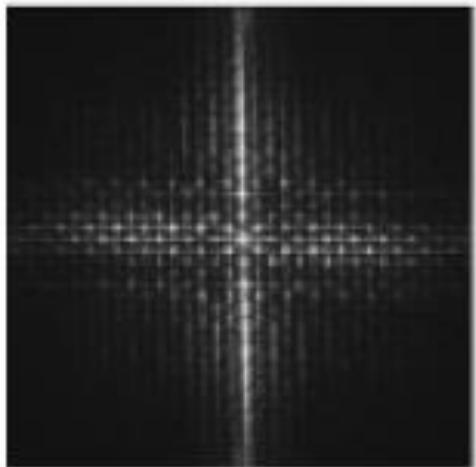
(a) A periodic texture
(D1).



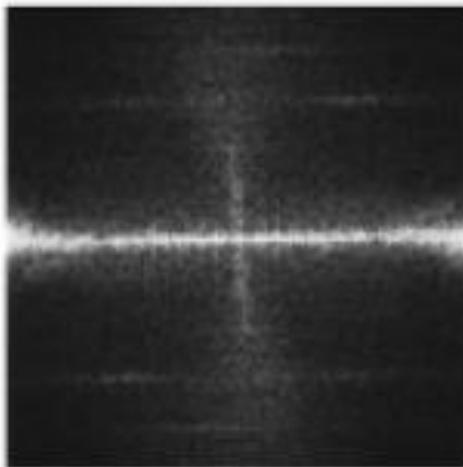
(b) A directional texture
(D106).



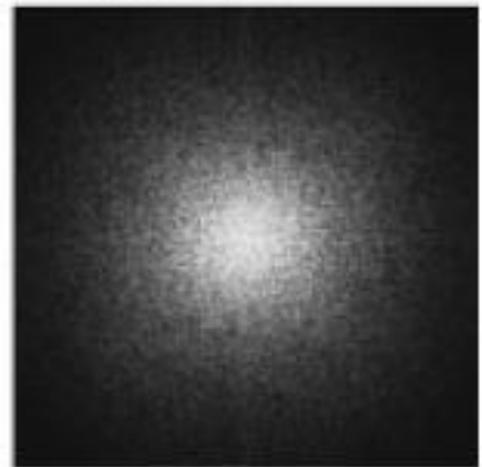
(c) A random texture
(D106).



(d) The Fourier spectrum
of (a).



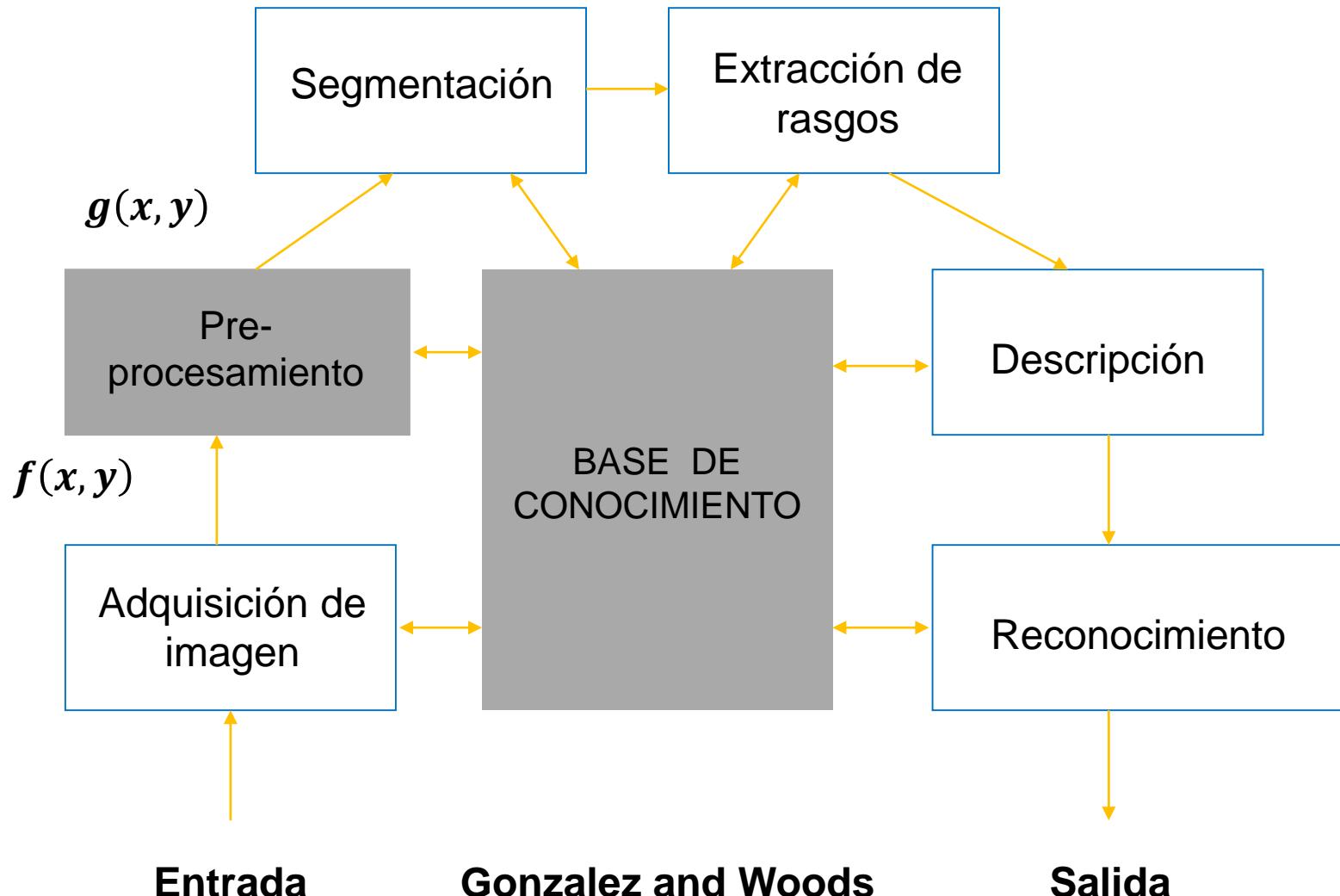
(e) The Fourier spectrum
of (b).



(f) The Fourier spectrum
of (c).

Pre-tratamiento o acondicionado de una imagen:

UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



FILTRADO EN EL DOMINIO DE LOS PÍXELES

FILTRADO LINEAL

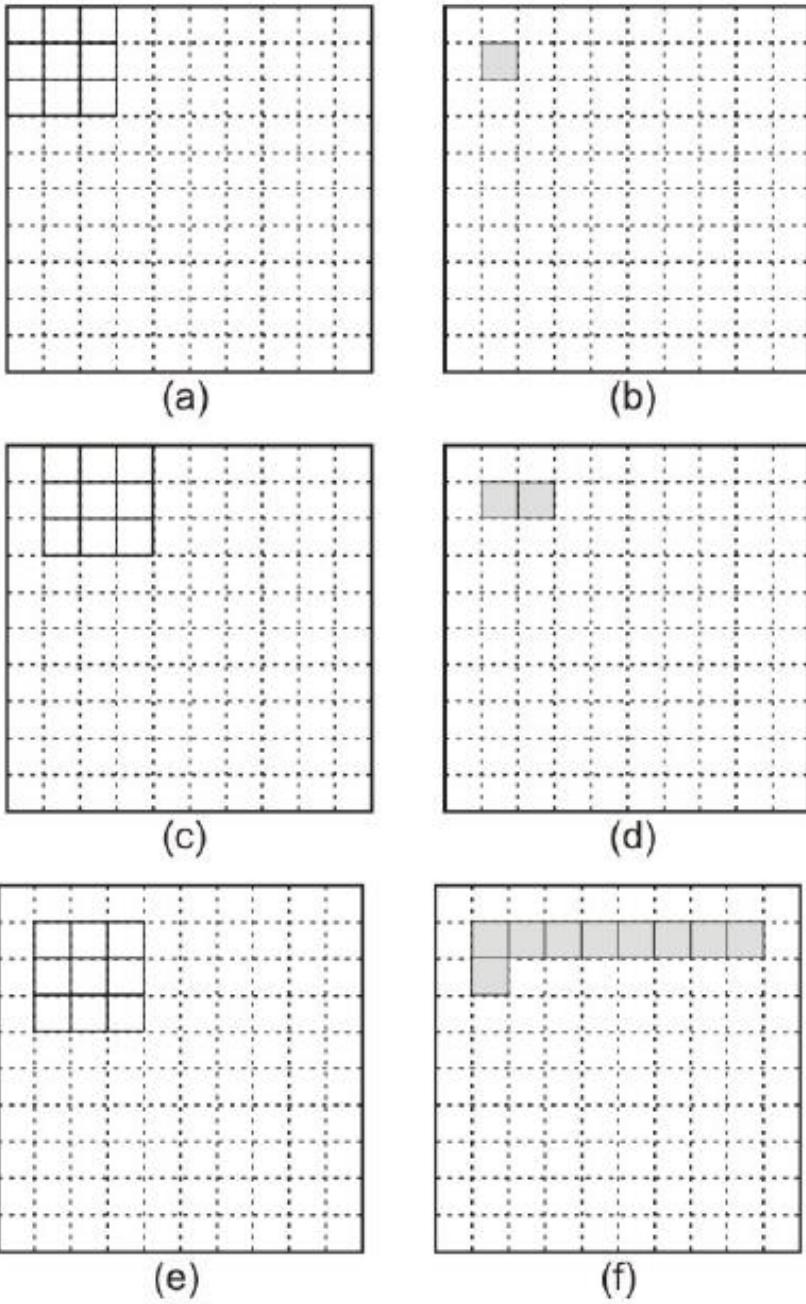
$$s(x, y) = T[f(x, y)]$$



FILTRADO LINEAL

PRINCIPIO DE APLICACIÓN DE UNA MÁSCARA DE FILTRADO

$$\begin{aligned}(f * g_N)[n] &= \sum_{m=0}^{N-1} f[m]g_N[n-m] \\&= \sum_{m=0}^n f[m]g[n-m] + \sum_{m=n+1}^{N-1} f[m]g[N+n-m] \\&= \sum_{m=0}^{N-1} f[m]g[(n-m) \bmod N] \triangleq (f *_N g)[n]\end{aligned}$$



FILTRADO LINEAL

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

El operador T puede tomar muchas formas. Una de ellas puede ser el producto interno de los pesos o valores de la máscara y los valores "justo abajo" de la imagen:

$$s = p_1 f_1 + p_2 f_2 + \cdots + p_n f_n = \sum_{k=1}^n p_k f_k$$

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

1	2	1
2	3	2
1	2	1

(a)

	2	1	5	
	4	5	3	
	2	3	3	

(b)

				49

(c)

- (a) Pesos de la máscara asociada al operador T .
(b) Valores debajo de esta máscara.
(c) Resultado.

2	1	1
1	3	1
1	1	2

Padding (rellenado)=1

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

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

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

$$(-1 \times 0) + (0 \times 0) + (1 \times 0) + \\ (-1 \times 0) + (0 \times 2) + (1 \times 1) + \\ (-1 \times 0) + (0 \times 1) + (1 \times 3) = 4$$

Stride (paso)=1

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

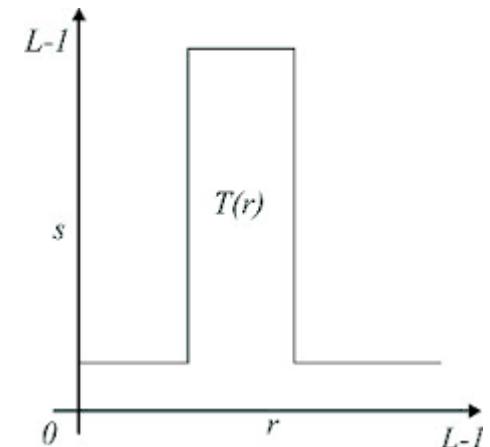
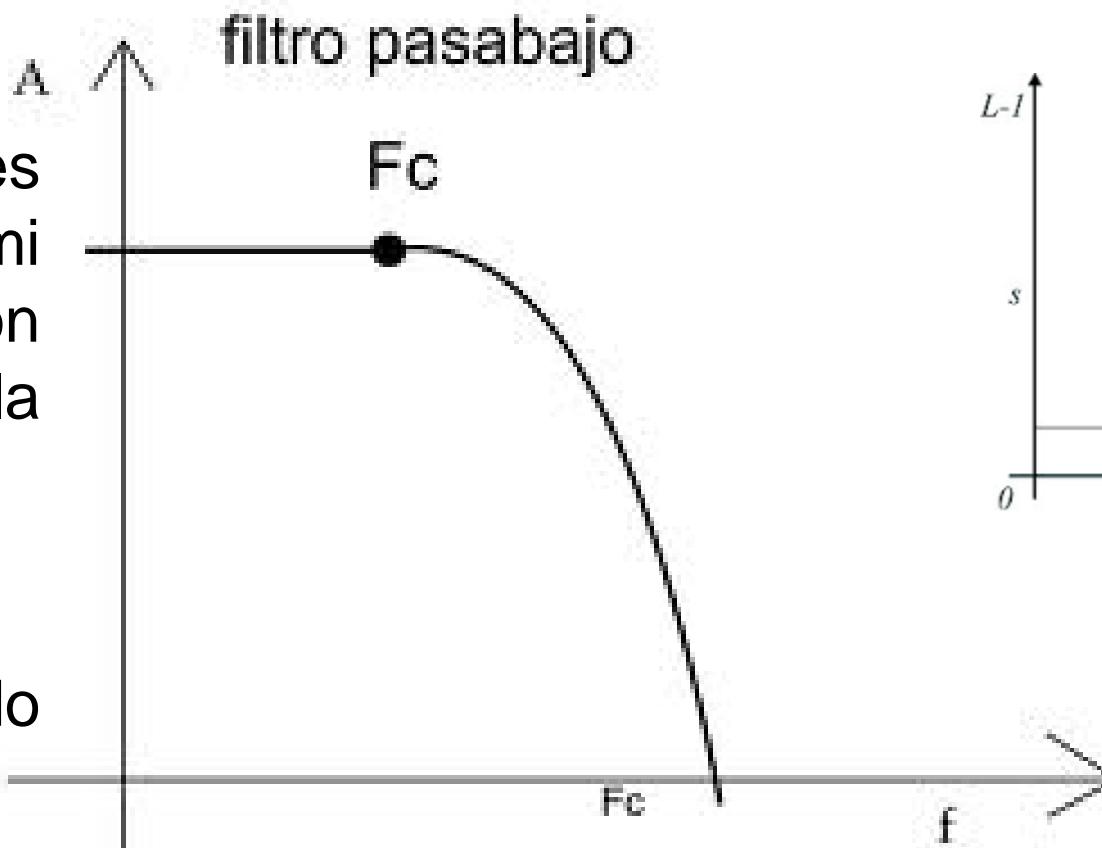
0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

$$(-1 \times 0) + (0 \times 0) + (1 \times 0) + \\ (-1 \times 2) + (0 \times 1) + (1 \times 1) + \\ (-1 \times 1) + (0 \times 3) + (1 \times 1) = -1$$

4	-1	X
X	X	X
X	X	X

Filtrado pasa bajas:

Si los valores de mi máscara son **positivos**, la máscara estará ejecutando un filtrado pasa-bajo.



En una imagen sería reducir la influencia del ruido asociado a las altas frecuencias de la señal.

FILTRADO PROMEDIO ARITMÉTICO:

$$f_{pa} = \frac{1}{n^2} \sum_{(x,y) \in V} f(x, y) \quad (3.7)$$

con n^2 el número de píxeles dentro de la ventana V .

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

$$s = \frac{1}{9} (2 + 1 + 1 + 1 + 3)$$

0	0	0	0	0
0	x	x	X	0
0	X	1	X	0
0	x	X	x	0
0	0	0	0	0

FILTRADO PROMEDIO ARITMÉTICO:

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

*

=

1	1	1
1	1	1
1	1	1

0	0	0	0	0
0	x	x	X	0
0	X	1	X	0
0	x	x	x	0
0	0	0	0	0

s

$$= \frac{1}{9} (1 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 3 + 1 \times 1 + 1 \times 1 + 1 \times 1)$$



(a)



(b)



(c)



(d)



(e)



(f)

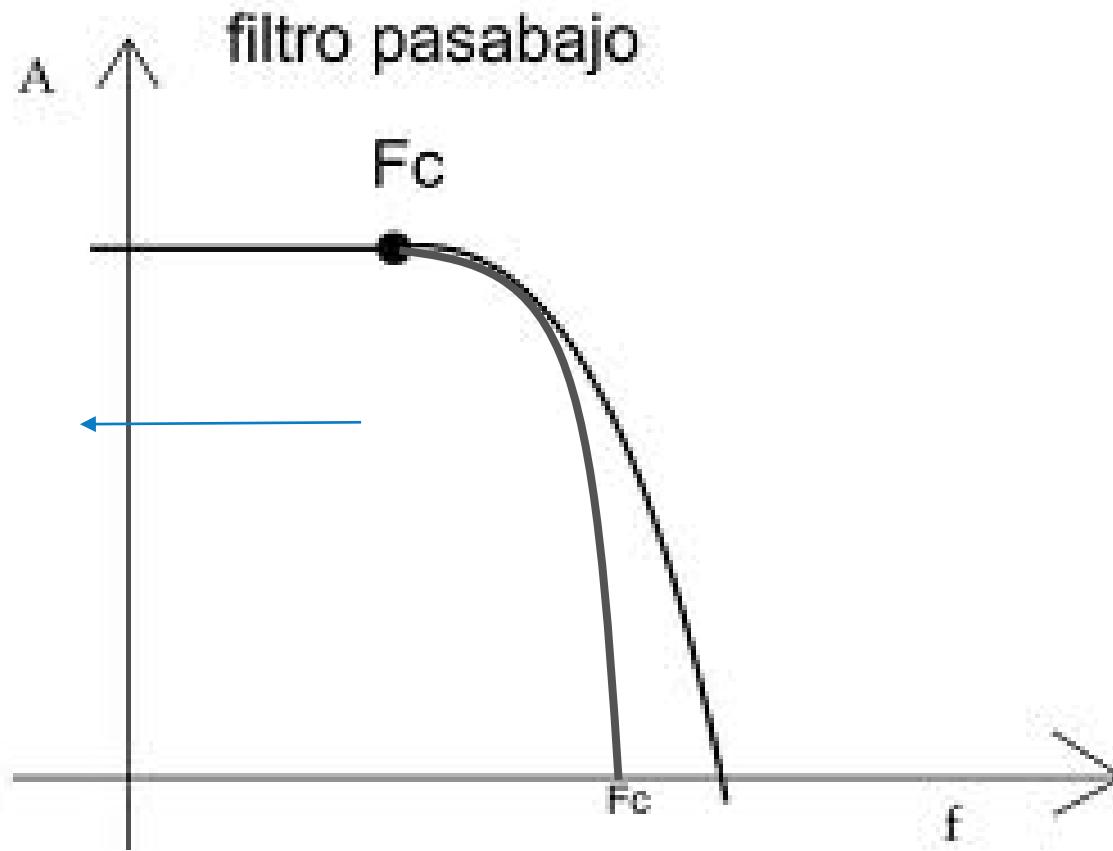
Efecto de aplicar el filtro de promediado aritmético de varios tamaños sobre una imagen. (a) 3x3, (b) 5x5, (c) 7x7, (d) 9x9, (e) 11x11 y (f) 13x13.

¿Qué notamos?

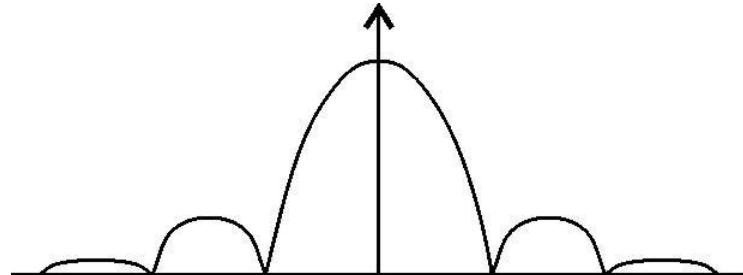
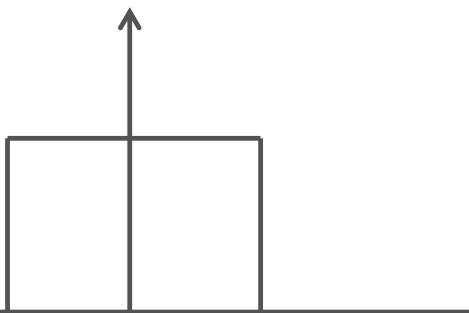
R: La imagen se va haciendo más difusa (desenfocada) conforme el tamaño de la máscara crece.

¿A qué equivale en el dominio de la frecuencia crecer la máscara?

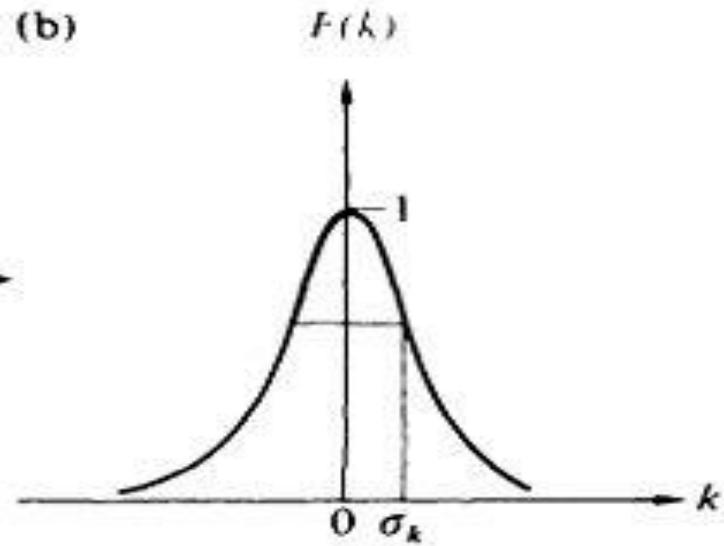
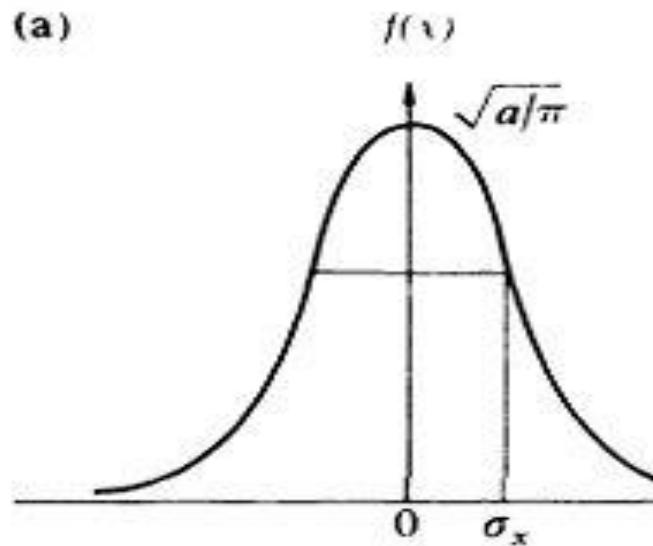
Conforme se crece el tamaño de la máscara, es como si la frecuencia de corte F_c del filtro se moviera hacia la izquierda.



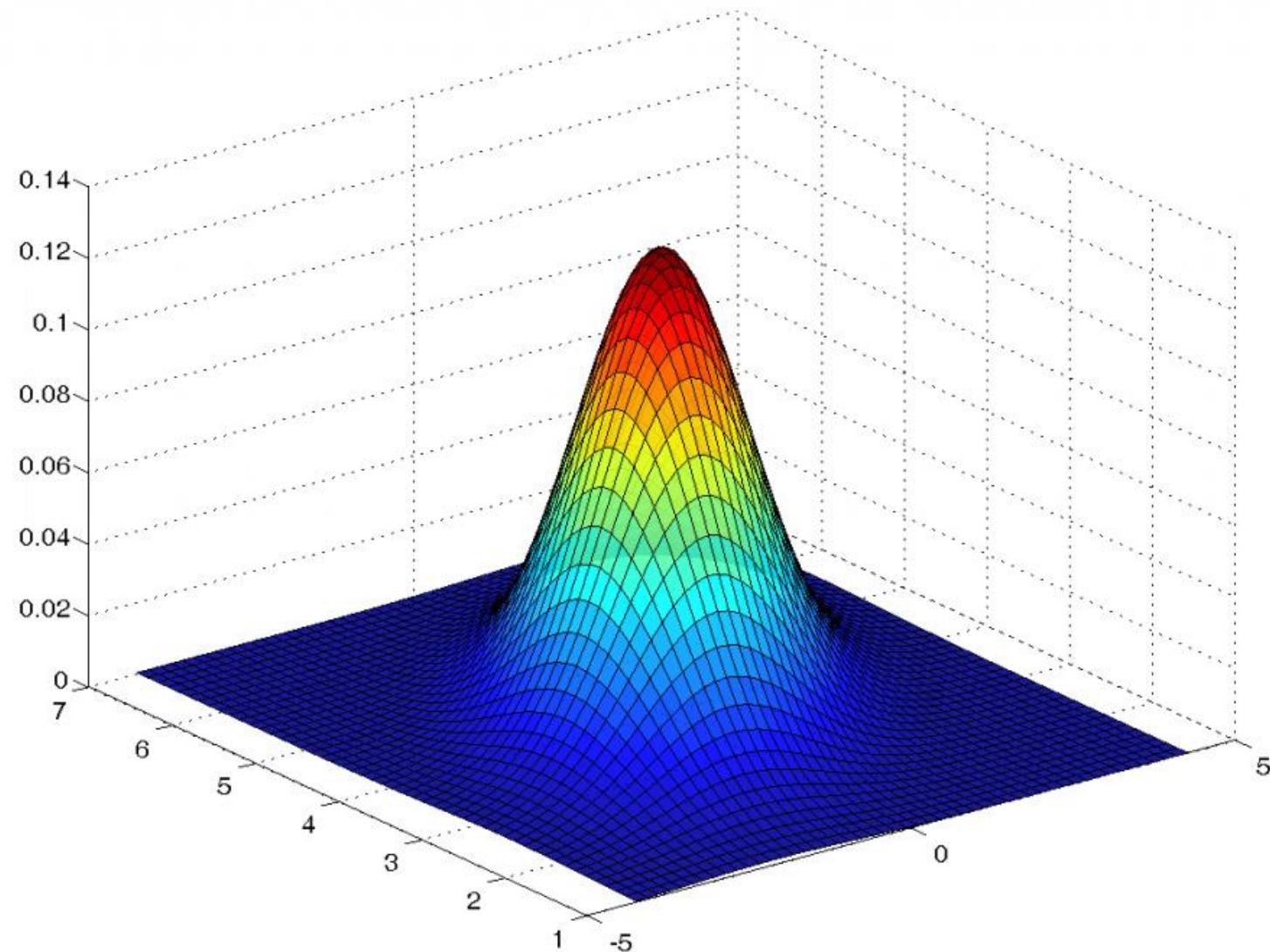
Se deja pasar menos información de alta frecuencia (detalle).



La transformada de Fourier del núcleo promedio aritmético 1-D viene dada por la parte real de la función SINC.



La transformada de Fourier del núcleo Gaussiano 1-D es otra Gaussiana



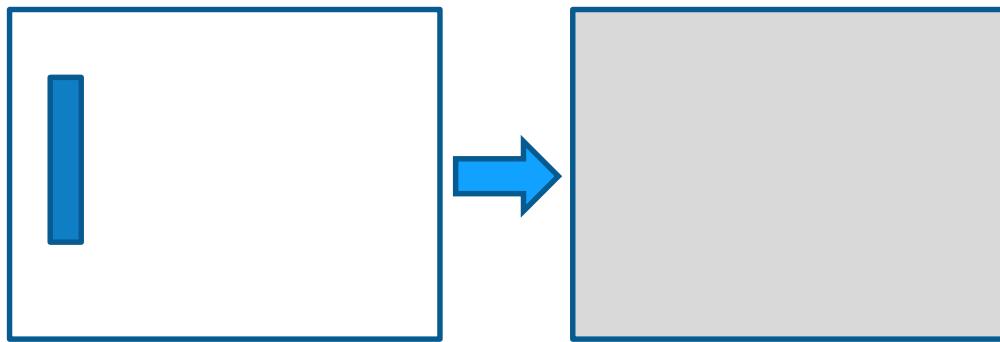
Separabilidad lineal del filtro Gaussiano:

$$\begin{aligned}f_G &= G * f \\&= \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} G(h, k) f(i - h, j - k) \\&= \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{h^2+k^2}{2\sigma^2}} f(i - h, j - k) \\&= \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{h^2}{2\sigma^2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{k^2}{2\sigma^2}} f(i - h, j - k)\end{aligned}$$

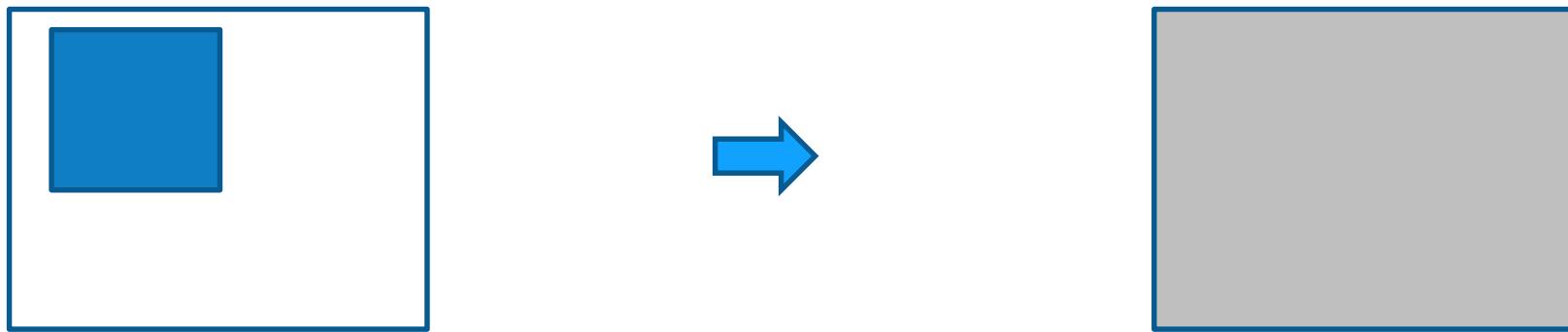
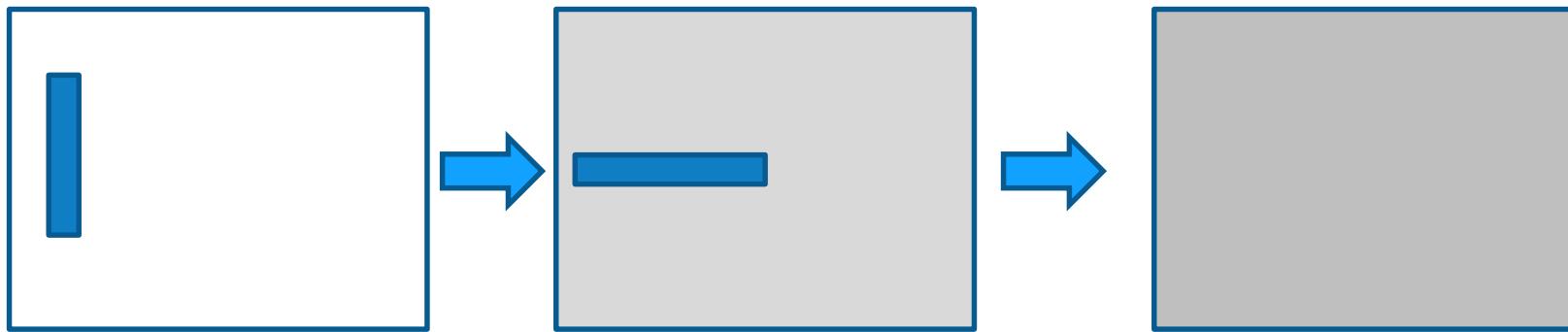
Separabilidad lineal del filtro Gaussiano:



Separabilidad lineal del filtro Gaussiano:

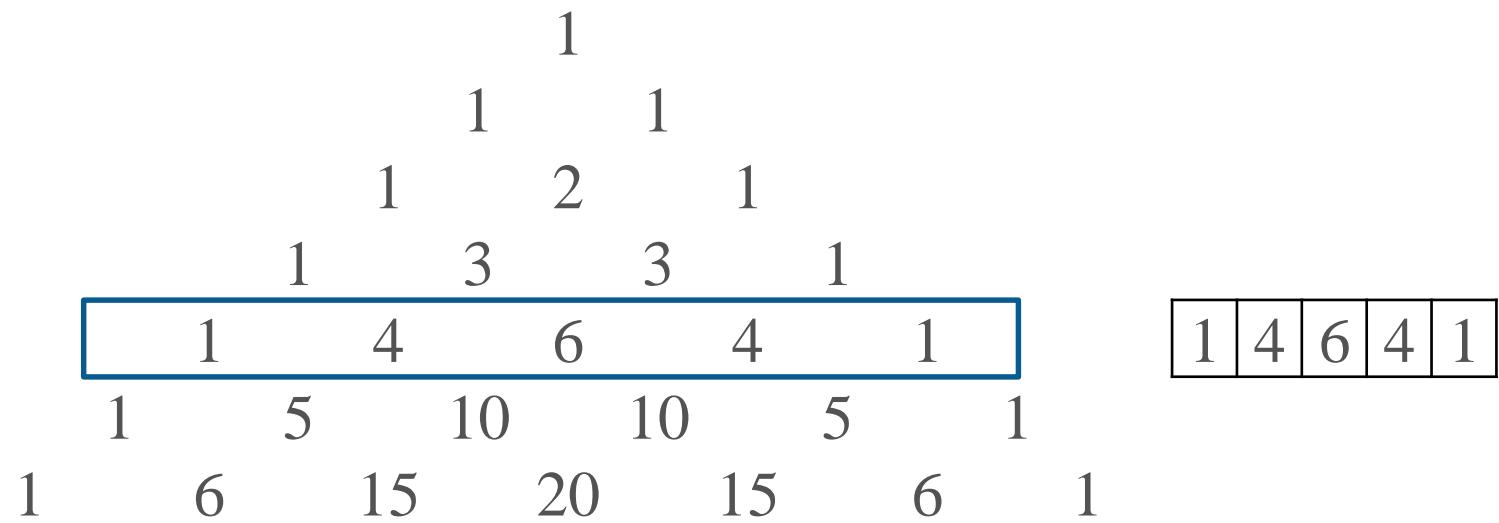


Separabilidad lineal del filtro Gaussiano:



Diseño de núcleos Gaussianos:

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{n}x^n$$



	-3	-2	-1	0	1	2	3
-3	0.011	0.039	0.082	0.105	0.082	0.039	0.011
-2	0.039	0.135	0.287	0.368	0.287	0.135	0.039
-1	0.082	0.287	0.606	0.779	0.606	0.287	0.082
0	0.105	0.368	0.779	1.0	0.779	0.368	0.105
1	0.082	0.287	0.606	0.779	0.606	0.287	0.082
2	0.369	0.135	0.387	0.368	0.387	0.135	0.039
3	0.011	0.039	0.082	0.105	0.182	0.039	0.011

$$G(x, y) = ce^{-\frac{x^2+y^2}{2\sigma^2}}$$

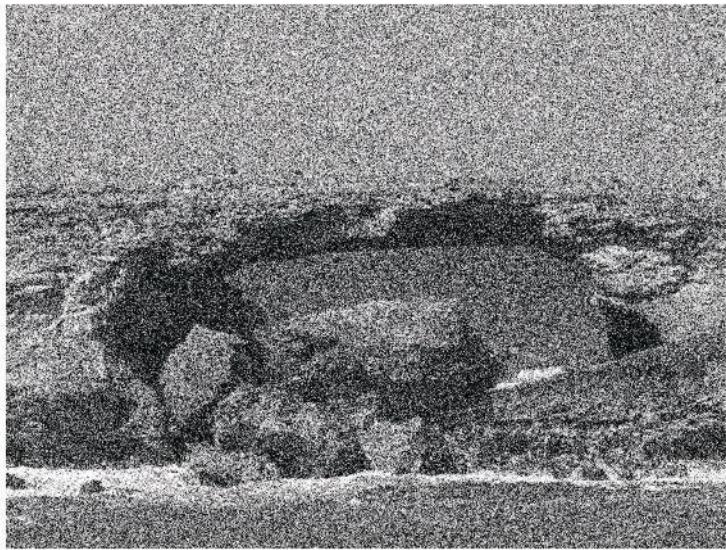
$$c = \sum_{h=-\frac{n}{2}}^{\frac{n}{2}} \sum_{k=-\frac{n}{2}}^{\frac{n}{2}} G(h, k) = 1115$$

1	4	7	10	7	4	1
4	12	26	33	26	12	4
7	26	55	71	55	26	7
10	33	71	91	71	33	10
7	26	55	71	55	26	7
4	12	26	33	26	12	4
1	4	7	10	7	4	1

$$\sigma^2 = 2$$

$$n = 7$$

	1	4	7	10	7	4	1
	4	12	26	33	26	12	4
	7	26	55	71	55	26	7
1115	10	33	71	91	71	33	10
	7	26	55	71	55	26	7
	4	12	26	33	26	12	4
	1	4	7	10	7	4	1



(a) Una imagen contaminada con ruido Gaussiano. (b) Imagen filtrada con un filtro promedio aritmético. (c) Imagen filtrada con núcleo Gaussiano.

(a)



(b)



(c)

```
% Programa para agregar ruido Gaussiano a una imagen y  
% para limpiar imagen mediante un filtro Gaussiano.  
  
clc % limpia pantalla.  
clearvars; % remueve todas las variables.  
close all;% cierra grácas, archivos y recursos abiertos.  
  
f=imread('Gato 01.JPG'); % lee imagen de prueba f(x,y).  
figure(1)  
imshow(f) % muestra imagen de prueba.  
title('Imagen original')  
[n,m]=size(f); % dimensiones de la imagen f(x,y).
```



Imagen original

```
fsp = imnoise(f,'gaussian',0,0.2); % permite agregar ruido  
% con una media y una sigma  
figure(2)  
imshow(fsp) % imagen con ruido.  
title('Imagen con ruido Gaussiano')  
  
H = fspecial('gaussian',9,2);  
blurred = imfilter(fsp,H,'replicate');  
figure(3);  
imshow(blurred);  
title('Imagen filtrada con promedio gaussiano')  
  
h=9; %tamaño del filtro promedio aritmético  
H = fspecial('average',h);  
blurred = imfilter(fsp,H,'replicate');  
figure(4);  
imshow(blurred);  
title('Imagen filtrada con promedio aritmético')
```

Filtrado Gaussiano comparado con filtrado promedio aritmético:



$\sigma = 0.2$



$tamaño = 9, \sigma = 2$



$tamaño = 9$

OTROS FILTROS LINEALES

FILTRO PROMEDIO ARMÓNICO.

FILTRO PROMEDIO CONTRA-ARMÓNICO.

FILTRO PROMEDIO GEOMÉTRICO.

FILTRO YP.

FILTRO PROMEDIO ARMÓNICO:

$$f_{ph} = \frac{n^2}{\sum_{(x,y) \in V} \frac{1}{f(x, y)}}$$

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

$$s = \frac{9}{\frac{1}{2} + \frac{1}{1} + \dots + \frac{1}{2}} = ?$$

Efecto de aplicar el filtro **promedio armónico** sobre una imagen contaminada con diversos tipos de ruido: sal y pimienta.



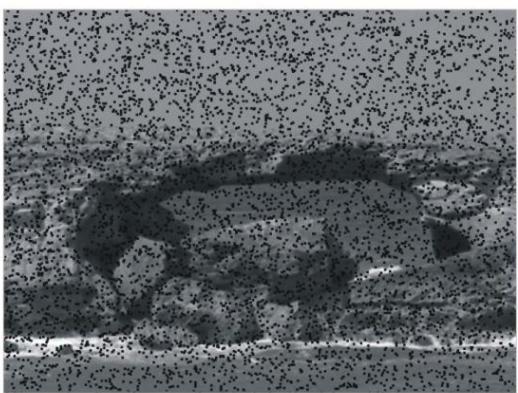
(a)



(d)



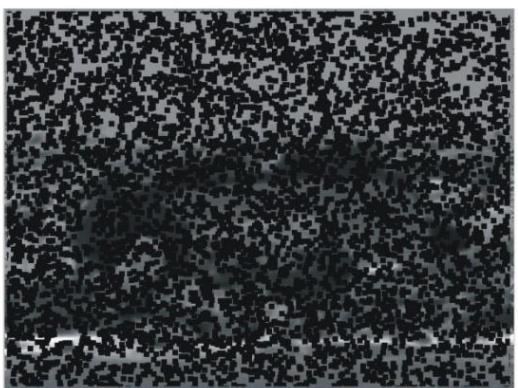
(b)



(e)



(c)



(f)

3×3

5×5

FILTRO PROMEDIO CONTRA-ARMÓNICO:

$$f_{ca} = \frac{\sum_{(x,y) \in V} f(x, y)^R}{\sum_{(x,y) \in V} f(x, y)^{R-1}}$$

$R < 0$, el filtro elimina ruido tipo sal.

$R > 0$, el filtro elimina ruido tipo pimienta.



(a)



(d)



(b)



(e)



(c)



(f)

Efecto de aplicar el filtro **contra-armónico** sobre una imagen contaminada ya sea con ruido tipo sal o tipo pimienta.

FILTRO PROMEDIO GEOMÉTRICO:

$$f_{pg} = \prod_{(x,y) \in V} [f(x, y)]^{\frac{1}{n^2}}$$

0	0	0	0	0
0	2	1	1	0
0	1	3	1	0
0	1	1	2	0
0	0	0	0	0

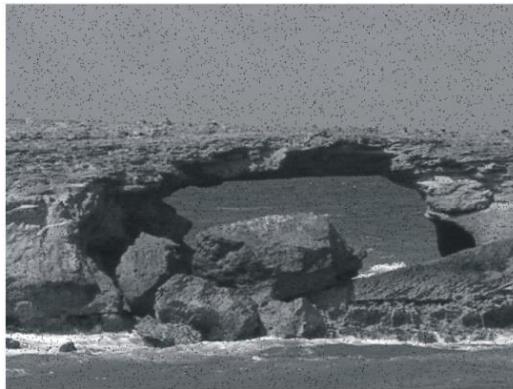
$$s = 2^{\frac{1}{9}} \times 1^{\frac{1}{9}} \times \cdots \times 2^{\frac{1}{9}}$$

¿Qué pasa si uno de los valores de intensidad es “0”?

R: 0.



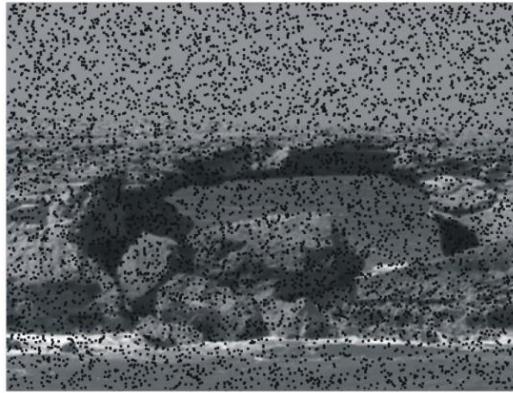
(a)



(d)



(b)



(e)



(c)



(f)

Efecto de aplicar el filtro **promedio geométrico** sobre una imagen contaminada con diversos tipos de ruido.

3×3

5×5

FILTRO Y_p:

$$f_{Y_p} = \left[\sum_{(x,y) \in V} \frac{f(x,y)^p}{n^2} \right]^{\frac{1}{p}}$$

$p < 0$, el filtro remueve ruido tipo sal.

$p > 0$, el filtro remueve ruido tipo pimienta.



(a)



(d)



(b)



(e)



(c)



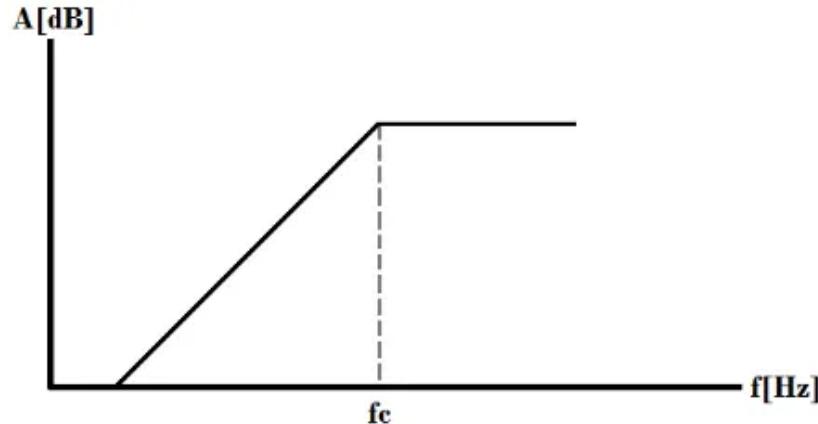
(f)

Efecto de aplicar el filtro Y_p sobre una imagen contaminada con diversos tipos de ruido: sal y pimienta.

3×3

5×5

Filtros resaltadores:



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

1	-2	1
-2	5	-2
1	-2	1



(a)

(b)

(c)

Las máscaras contienen valores positivos, negativos y “0” y además la suma de los pesos es ligeramente positivo.

Filtros enfatizadores de altas frecuencias "High-Boost":

$$f_a = f - f_b$$

$A = 1$, da como resultado la versión pasa-alto estándar.

$$f_e = A \cdot f - f_b$$

$A > 1$, parte de la información de la imagen original es adicionada a la imagen filtrada en sus altas frecuencias.

$$\begin{aligned}f_e &= A \cdot f - f_b \\&= (A-1)f + f - f_b \\&= (A-1)f + f_a\end{aligned}$$

Filtros enfatizadores de altas frecuencias "High-Boost":

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

-1	-1	-1
-1	P	-1
-1	-1	-1

$$P = 9 \cdot A - 1$$

$$\begin{aligned}f_e &= A \cdot f - f_b \\&= (A - 1)f + f - f_b \\&= (A - 1)f + f_a\end{aligned}$$

Al deslizar la mascara, si su suma es “0” y si los valores en la imagen son semejantes, entonces, el resultado tiende a ser “0”.

Si $A = 1$, $P = 8$.

Efecto de aplicar el filtro enfatizador sobre una imagen:

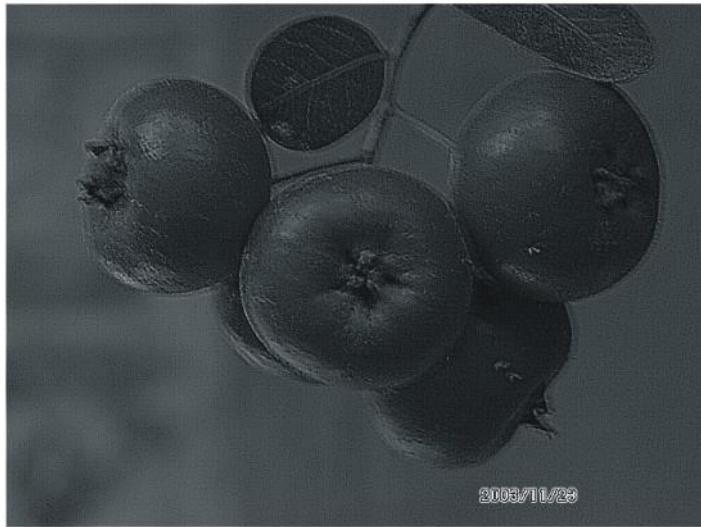
$$A = 1.01,$$
$$A = 1.05,$$
$$A = 1.1.$$



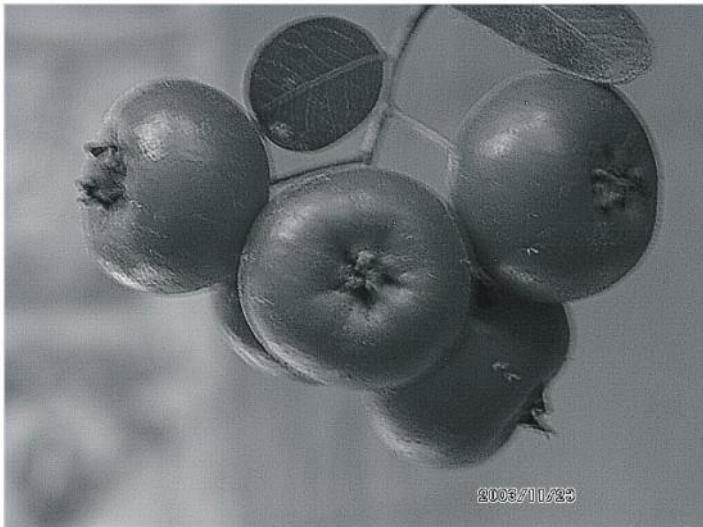
(a)



(b)

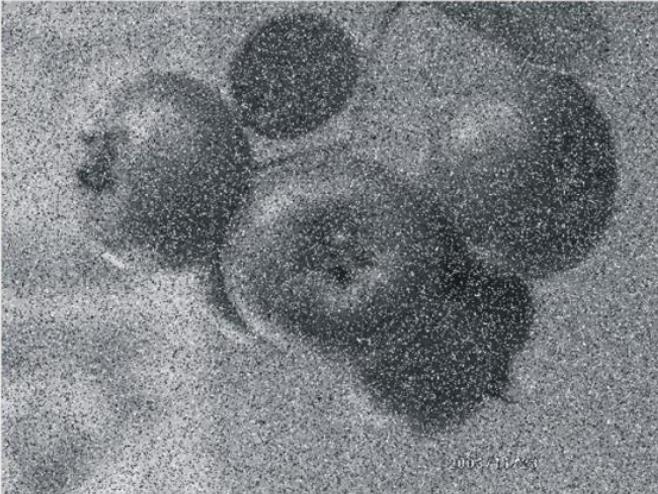


(c)



(d)

Los filtros lineales **no** son útiles para el caso de ruido no mezclado:



(a)



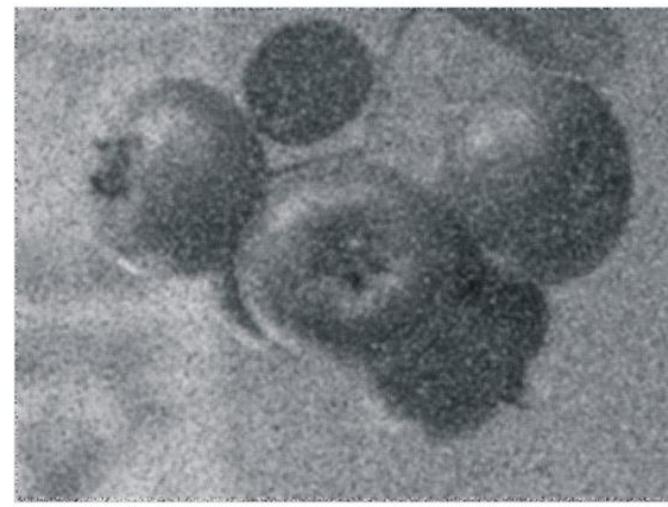
(b)

3×3



5×5

(c)



(d)

FILTROS NO LINEALES DE ORDEN

Principio general de operación:

- 1. ORDENAMIENTO DE LOS VALORES DENTRO DE LA VENTANA de valores desordenados**
- 2. ELEGIR UNA REGLA PARA SELECCIÓN UN VALOR.**
- 3. ASIGNACIÓN EL NUEVO VALOR EN LA IMAGEN DE SALIDA.**

$$f_1 \leq f_2 \leq f_3 \leq \cdots \leq f_{N^2}$$

$$\left\lfloor f_1, f_2, f_3, \dots, f_{N^2} \right\rfloor$$

EJEMPLO: FILTRO MEDIANA

5	10	8
11	15	13
42	10	12

5 10 8 11 15 13 42 10 12

EJEMPLO: FILTRO MEDIANA

5	10	8
11	15	13
42	10	12

5 10 8 11 15 13 42 10 12



(5, 8, 10, 10, 11, 12, 13, 15, 42)

EJEMPLO: FILTRO MEDIANA

5	10	8
11	15	13
42	10	12

5 10 8 11 15 13 42 10 12



(5, 8, 10, 10, 11, 12, 13, 15, 42)

EJEMPLO: FILTRO MEDIANA

5	10	8
11	15	13
42	10	12

5 10 8 11 15 13 42 10 12



(5, 8, 10, 10, 11, 12, 13, 15, 42)



(5, 8, 10, 10, 11, 12, 13, 15, 42)



11

5 10 8 11 15 13 42 10 12



(5, 8, 10, 10, 11, 12, 13, 15, 42)



(5, 8, 10, 10, [11,] 12, 13, 15, 42)



11

5	10	8
11	15	13
42	10	12



X	X	X
X	11	X
X	X	X

23	14	15	28	11
4	35	6	2	14
45	56	3	5	7
12	7	8	32	45
45	86	2	5	13



0	0	0	0	0
0	15	x	x	0
0	x	x	X	0
0	x	x	x	0
0	0	0	0	0

3 4 6 14 15 23 35 45 56

23	14	15	28	11
4	35	6	2	14
45	56	3	5	7
12	7	8	32	45
45	86	2	5	13



0	0	0	0	0
0	15	14	x	0
0	x	x	X	0
0	x	x	x	0
0	0	0	0	0

3 4 6 14 15 23 35 45 56

2 3 5 6 14 15 28 35 56

23	14	15	28	11
4	35	6	2	14
45	56	3	5	7
12	7	8	32	45
45	86	2	5	13



0	0	0	0	0
0	15	14	7	0
0	x	x	X	0
0	x	x	x	0
0	0	0	0	0

3 4 6 14 15 23 35 45 56

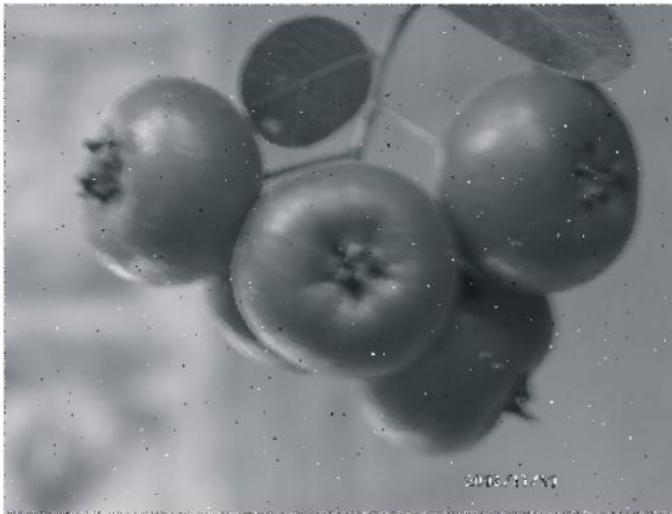
2 3 5 6 14 15 28 35 56

2 3 4 6 7 11 14 15 28

¿Porqué se obtuvo este resultado?



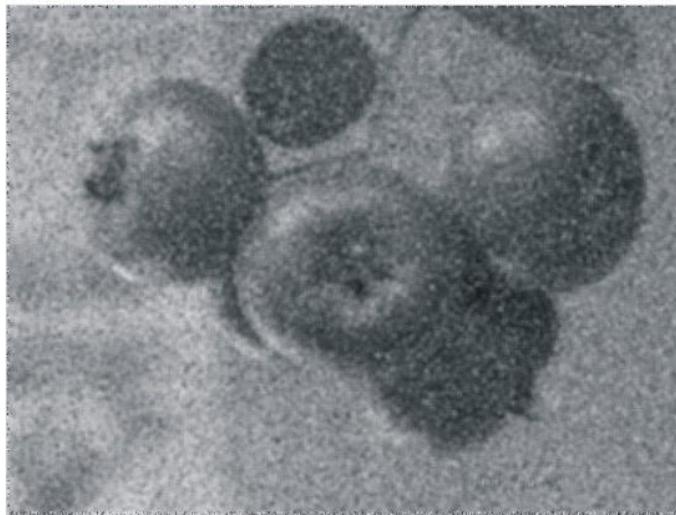
(a)



(b)



(c)



(d)

Efecto de aplicar el filtro mediana sobre una imagen contaminada con ruido sal y pimienta

% Programa para agregar ruido sal y pimienta y para
% filtrar mediante un filtro mediano.

clc % limpia pantalla.

clearvars; % remueve todas las variables.

close all;% cierra grácas, archivos y recursos abiertos.

f=imread('6 objetos.JPG'); % lee imagen de prueba

f(x,y).

figure(1)

imshow(f) % muestra imagen de prueba.

title('Imagen original')

[n,m]=size(f); % dimensiones de la imagen f(x,y).

%g=zeros(n,m);

%g=f;

%disp(n); % filas

%disp(m); % columnas

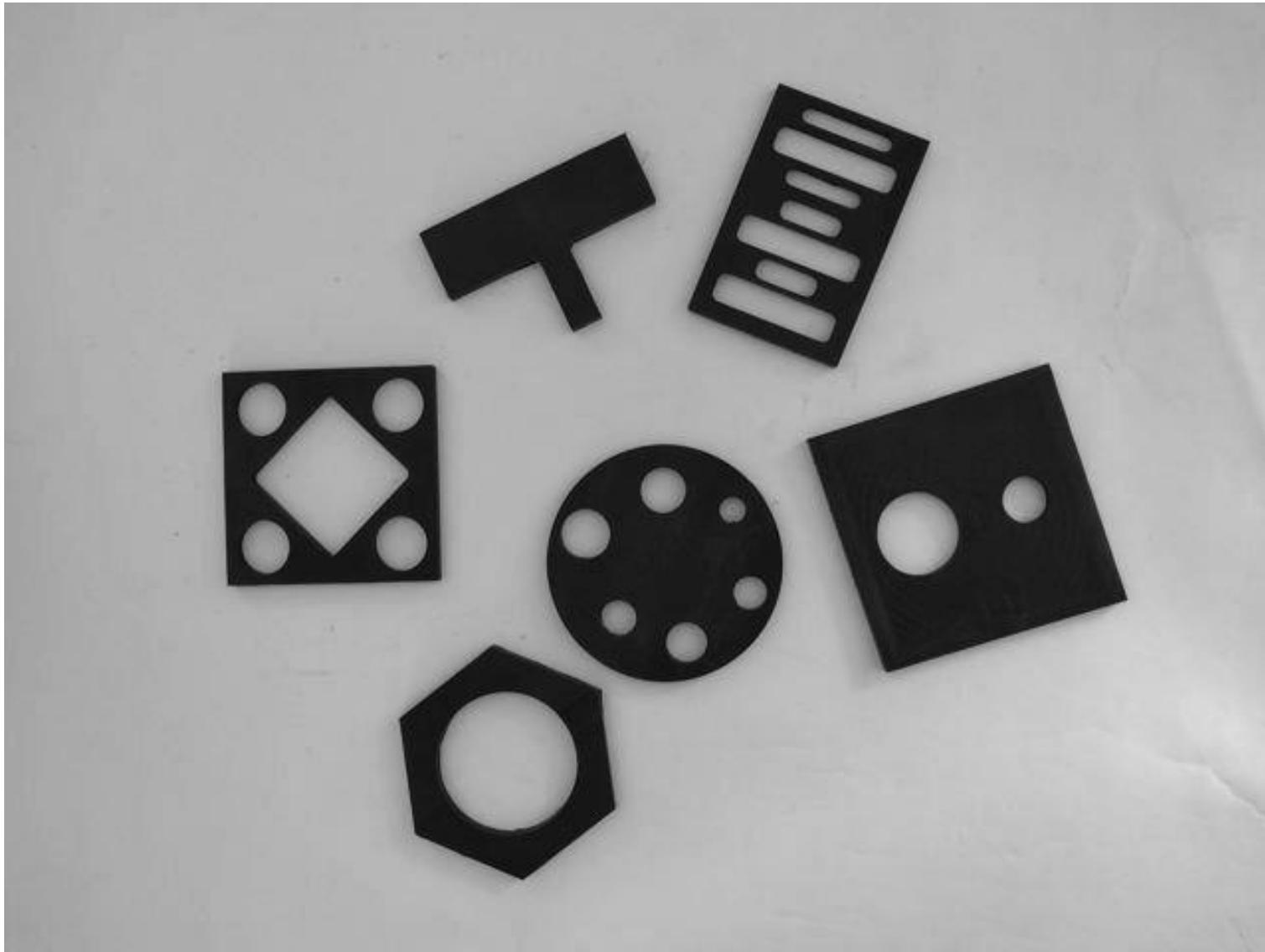


Imagen original

```
fsp = imnoise(f,'salt & pepper',0.6);  
figure(2)  
imshow(fsp) % imagen con ruido.  
title('Imagen con ruido sal y pimienta')
```

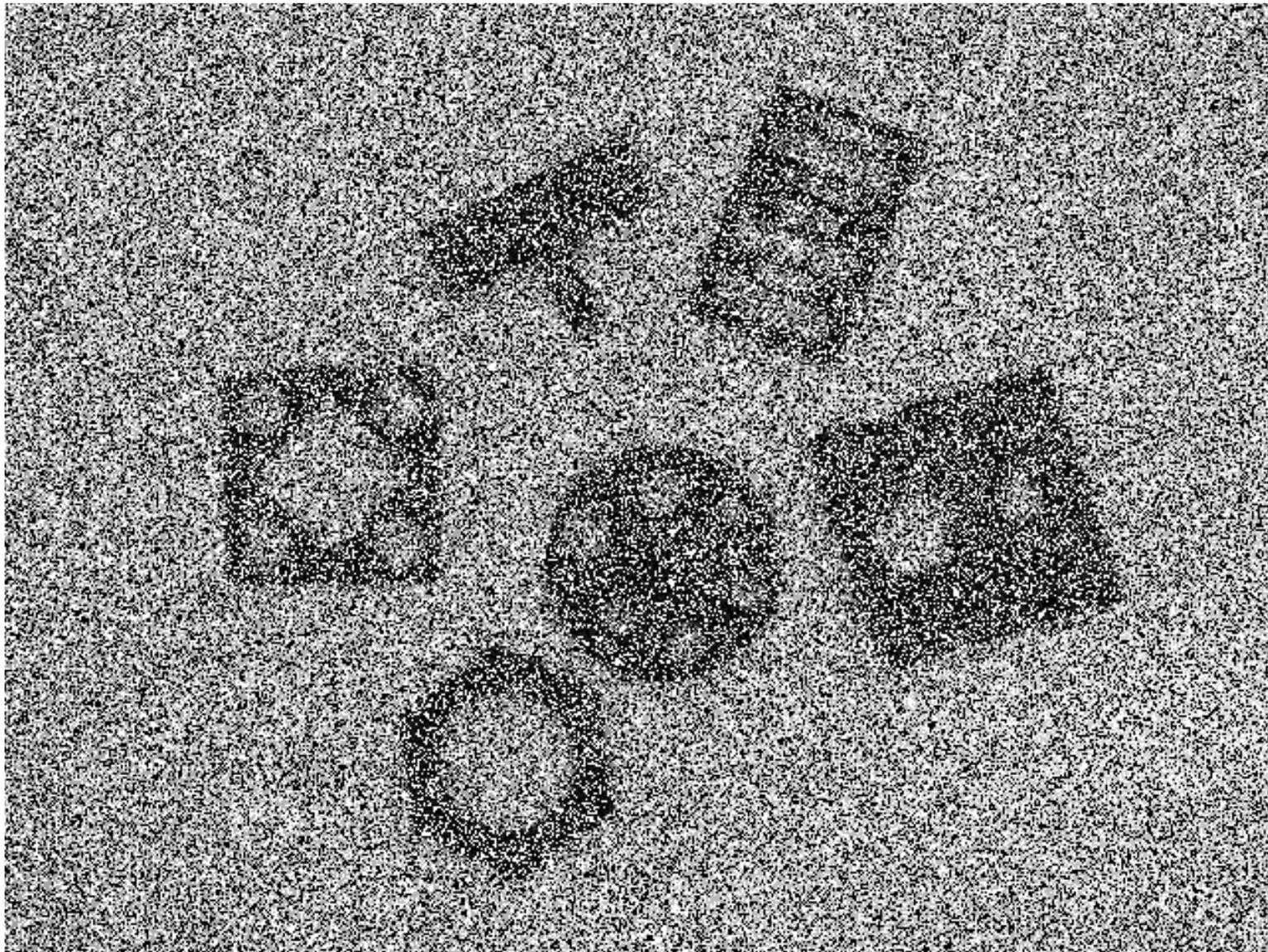


Imagen con ruido sal y pimienta

```
fmed = medfilt2(fsp, [3 3]);  
figure(3)  
imshow(fmed) % imagen filtrada con filtro mediana de mxn  
title('Imagen filtrada con mediana de m x n')
```

```
h=3; %tamaño del filtro promedio aritmético  
H = fspecial('average',h);  
blurred = imfilter(fsp,H,'replicate');  
figure(4);  
imshow(blurred);  
title('Imagen filtrada con promedio aritmético')
```

Otros ejemplos:

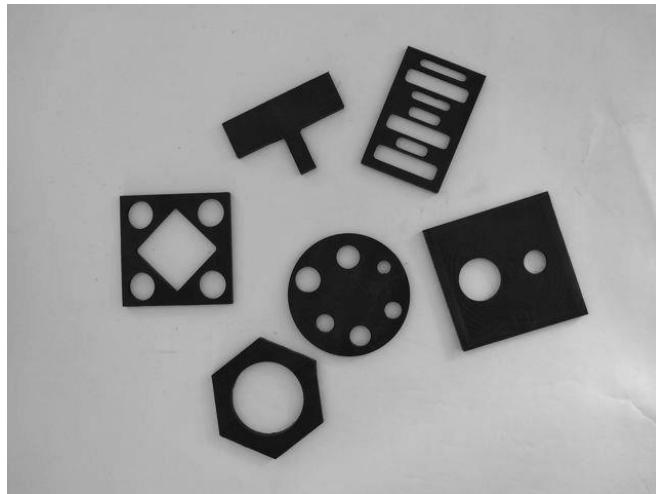
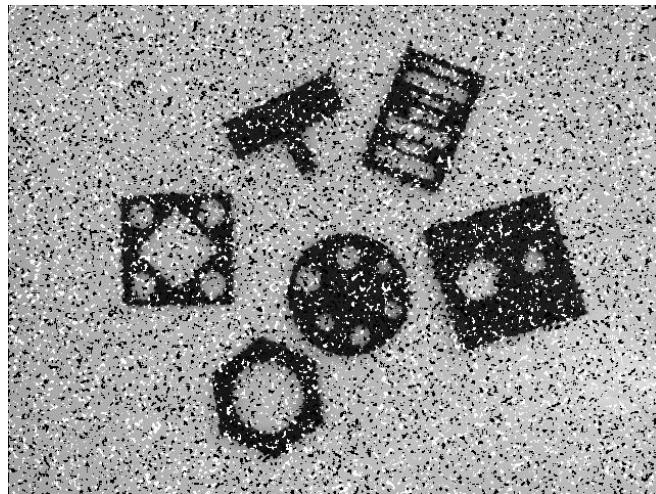
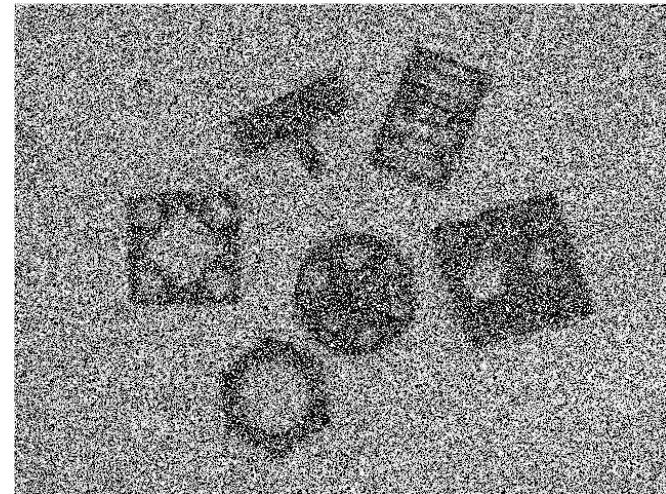


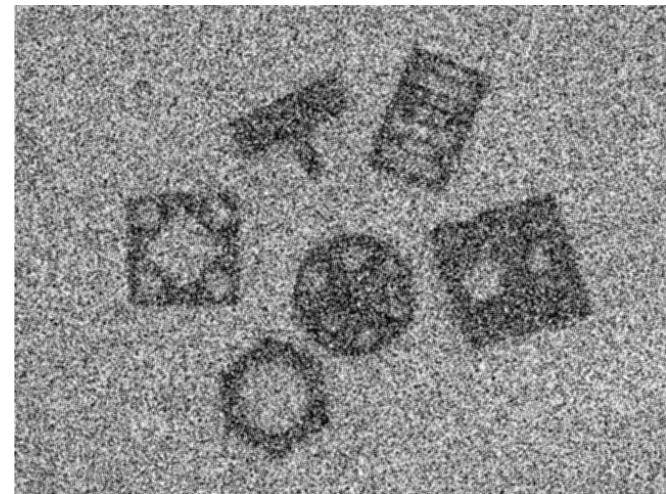
Imagen original



Filtro mediano de 3×3



Sal y pimienta $l = 0.6$



Filtro p. aritmético de 3×3

Otros ejemplos:

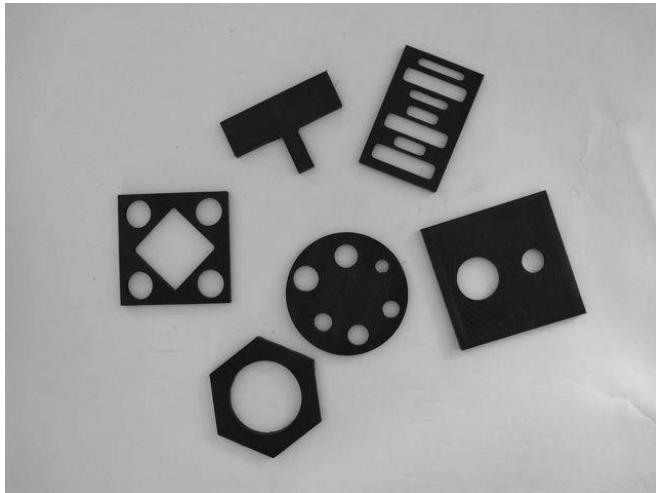
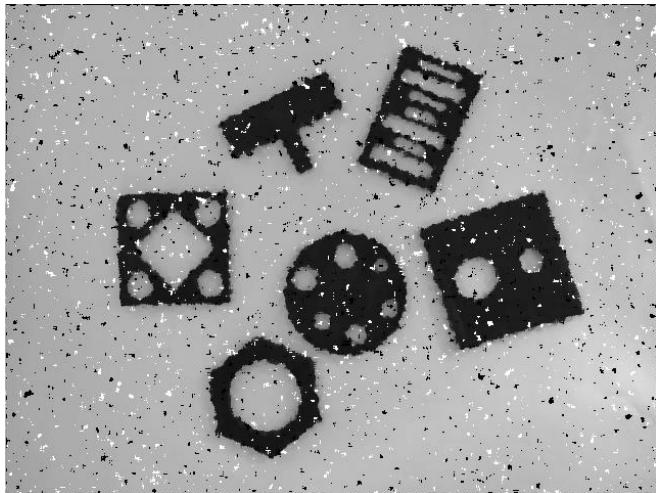
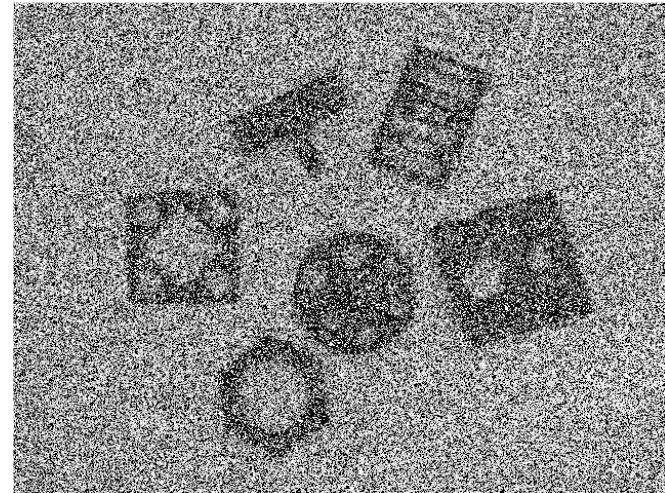


Imagen original



Filtro mediano de 5×5



Sal y pimienta $l = 0.6$



Filtro p. aritmético de 5×5

Otros ejemplos:

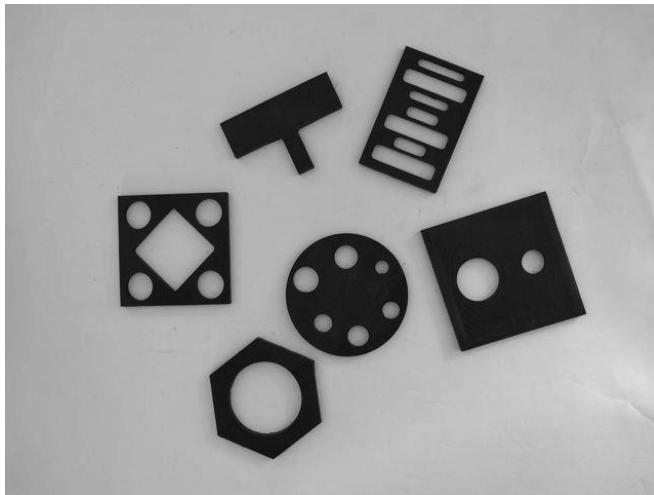
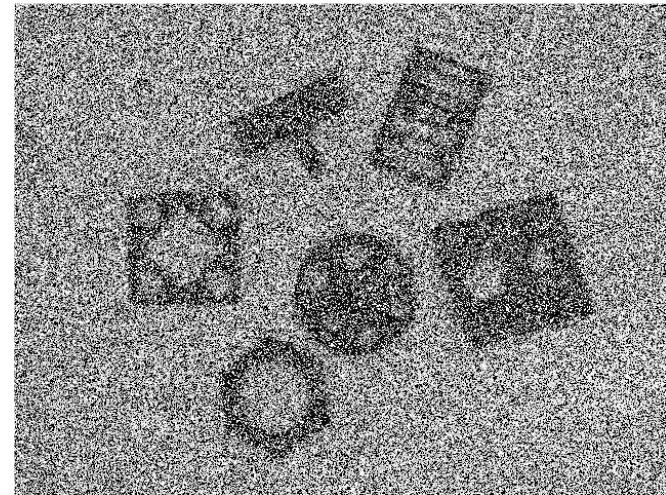
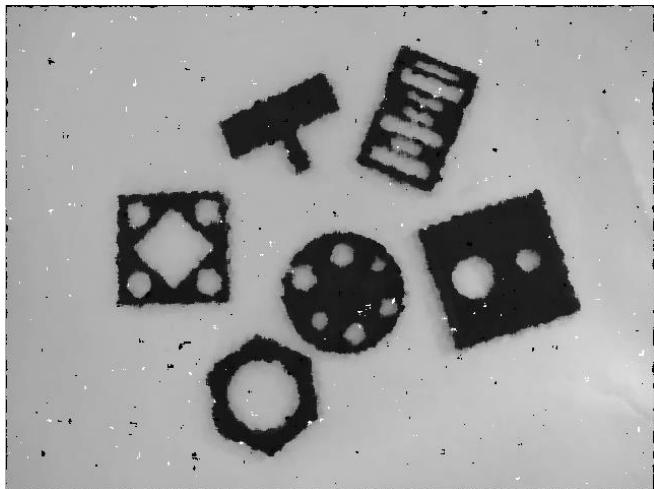


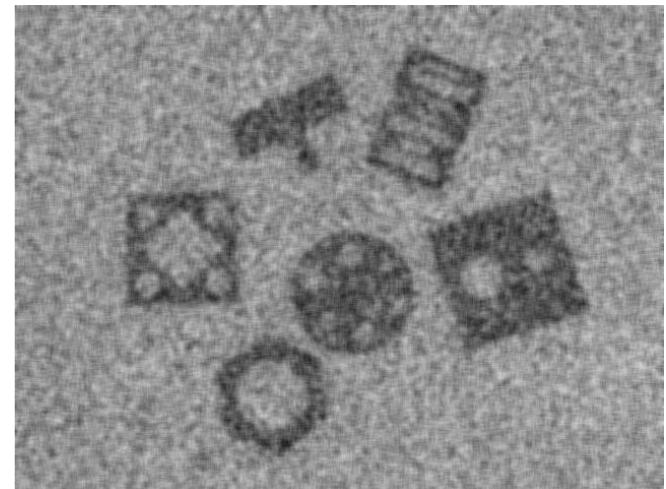
Imagen original



Sal y pimienta $l = 0.6$



Filtro mediano de 7×7



Filtro p. aritmético de 7×7

$$f_1 \leq f_2 \leq f_3 \leq \cdots \leq f_{N^2}$$

Filtros max y min:

$$f_{\max} = \max \{f_i \mid i = 1, 2, \dots, N^2\}$$

$$f_{\min} = \min \{f_i \mid i = 1, 2, \dots, N^2\}$$



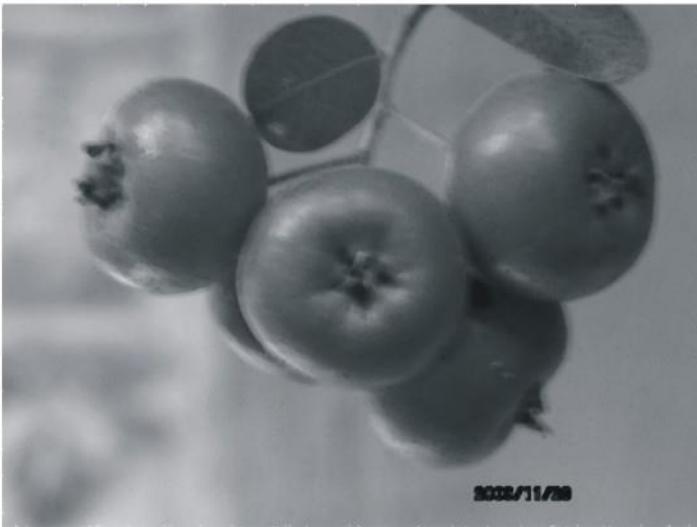
(a)



(b)



(c)



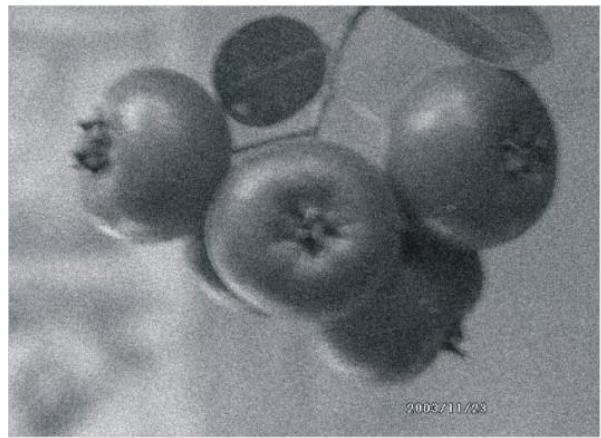
(d)

Efecto de aplicar los filtros **max** y **min** sobre una imágenes contaminadas, respectivamente con ruido pimienta y sal

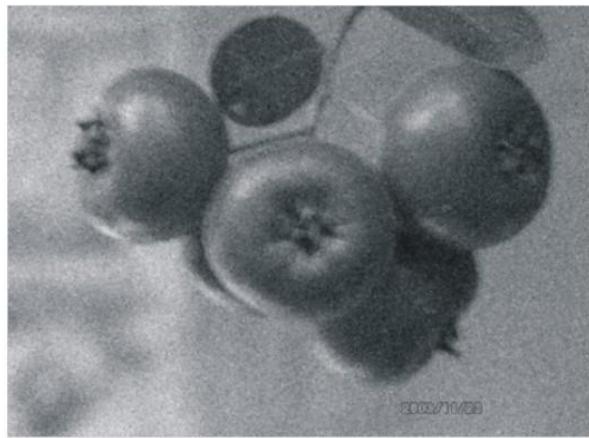
$$f_1 \leq f_2 \leq f_3 \leq \dots \leq f_{N^2}$$

Filtro de punto medio:

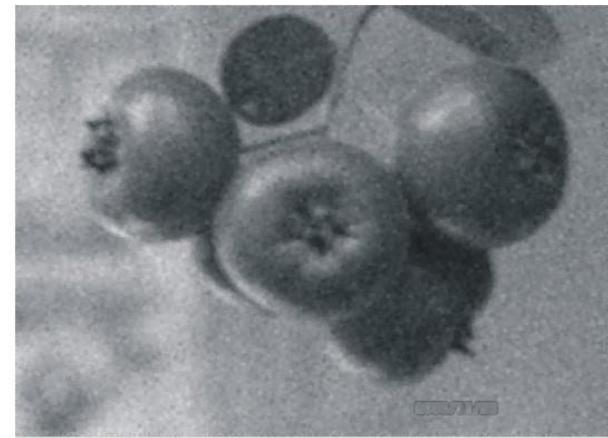
$$f_{pm} = \frac{f_1 + f_{N^2}}{2}$$



(a)



(b)



(c)

Efecto de aplicar el filtro de **punto medio** sobre una imagen.

OTROS FILTROS DE ORDEN

FILTRO α -T

$$f_{\alpha-T} = \frac{1}{N^2 - 2T} \sum_{i=T}^{N^2-T} f_i$$

5	10	8
11	15	13
42	10	12

5 10 8 11 15 13 42 10 12

ventana de 3×3



$$T, T = 0, T = 2, T = 4.$$

(5, 8, 10, 10, 11, 12, 13, 15, 42)



$$T = 0$$

$$T = (N^2 - 1)/2$$

$$T = 2 \quad (5, 8, \boxed{10}, 10, 11, 12, 13, \boxed{15, 42})$$

$$(10+10+11+12+13)/(9-4) = 11$$

OTROS FILTROS DE ORDEN

FILTRO α -T

$$f_{\alpha-T} = \frac{1}{N^2 - 2T} \sum_{i=T}^{N^2-T} f_i$$

ventana de 3×3

Si $T = 0$ mi filtro es un filtro **promedio aritmético.**

$T, T = 0, T = 2, T = 4.$

Si $T = (N^2 - 1)/2$, entonces mi filtro es un filtro **mediano.**

$T = 0$

$T = (N^2 - 1)/2$

OTROS FILTROS DE ORDEN

$$0 \leq T \leq \frac{(N^2 - 1)}{2} \quad N \text{ es impar} \quad 0 \leq T \leq 4$$

$$T = 0$$

FILTRO α -T

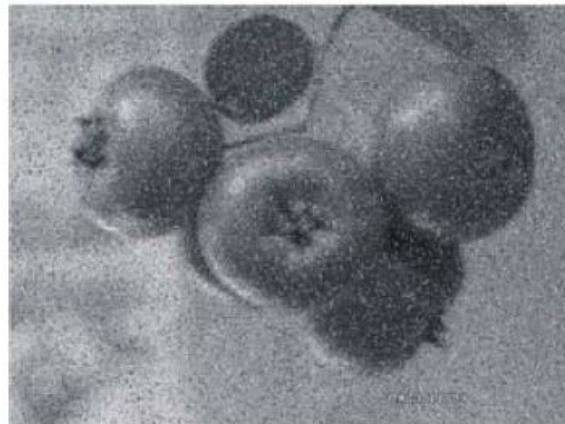
$$f_{\alpha-T} = \frac{1}{N^2 - 2T} \sum_{i=T}^{N^2 - T} f_i$$

ventana de 3×3

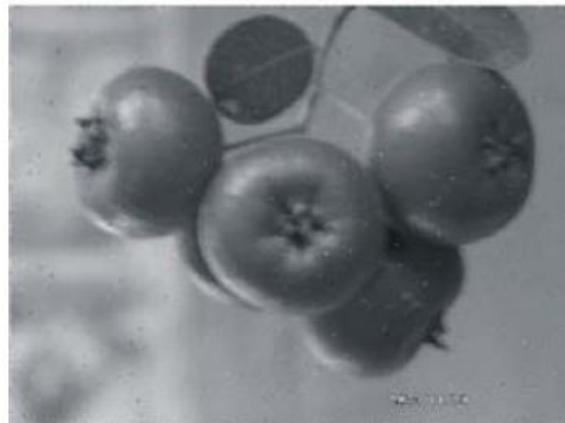
$T, T = 0, T = 2, T = 4.$

$T = 0$

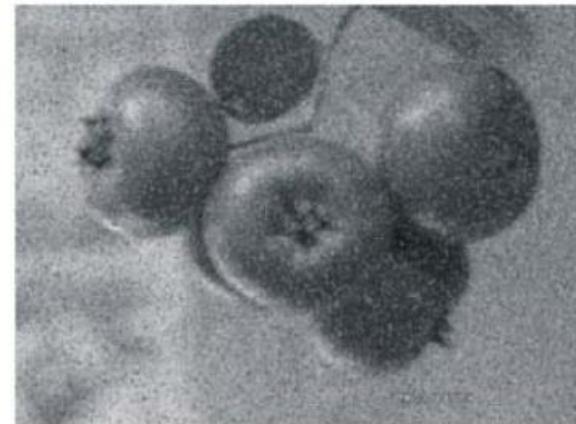
$T = (N^2 - 1)/2$



(a) $T = 2$



(c)



(b) $T = 4$



(d)

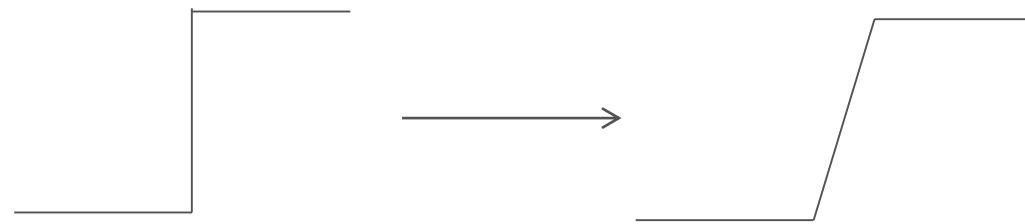
Identificación de siluetas mediante detección de bordes:

Un borde en una imagen es un cambio local significativo, normalmente asociado con una **discontinuidad** ya sea en la imagen de intensidad o la primera derivada de dicha imagen. Discontinuidades de este tipo en una imagen pueden ser:

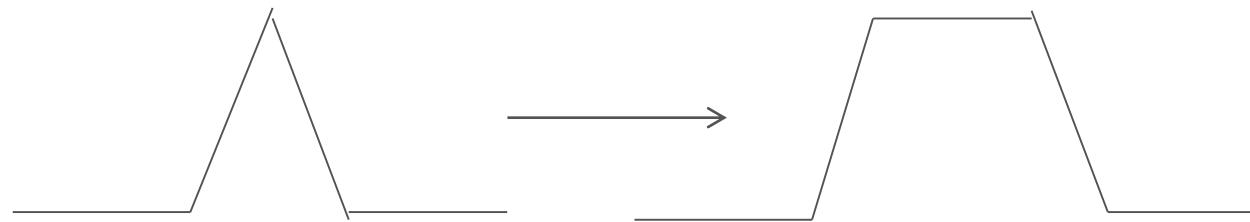
De tipo escalón. En este caso la intensidad en la imagen cambia de manera abrupta de un valor en un lado de la discontinuidad a otro valor muy diferente en el lado opuesto de la discontinuidad.

Lineales. En este caso la intensidad en la imagen cambia de nuevo abruptamente de valor pero regresa al valor inicial en una distancia corta.

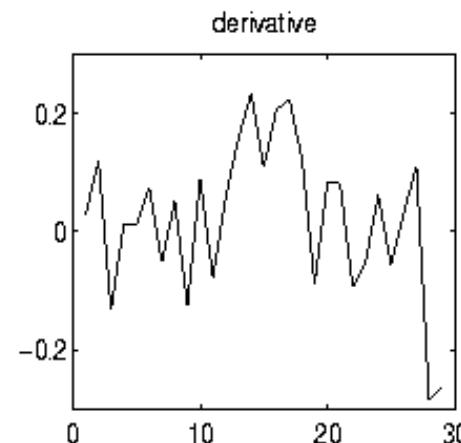
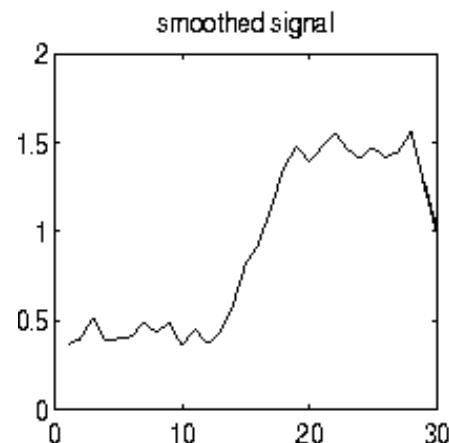
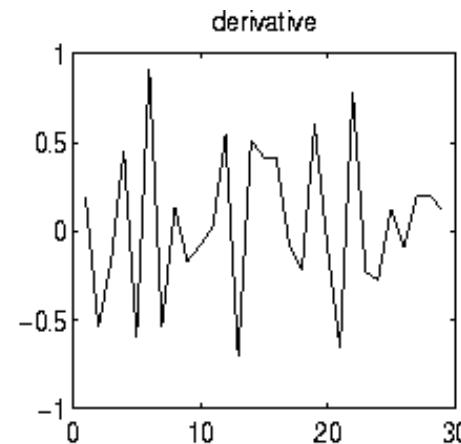
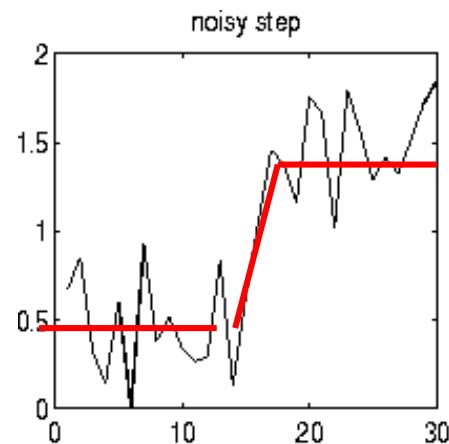
Debido a las componentes de baja frecuencia en la señal o el alisamiento introducido por la mayoría de los dispositivos de captación, rara vez se presentan discontinuidades agudas en una imagen real. Los bordes de tipo paso aparecen como bordes tipo rampa,



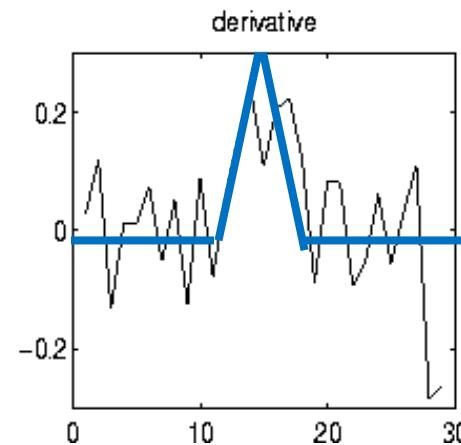
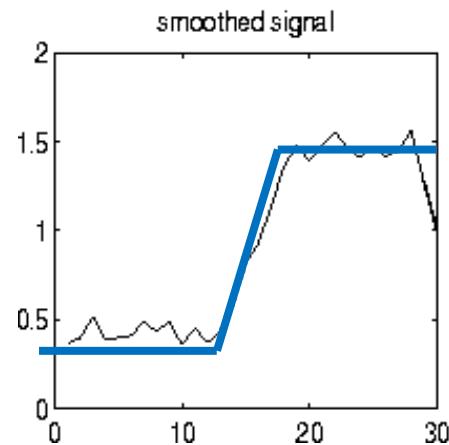
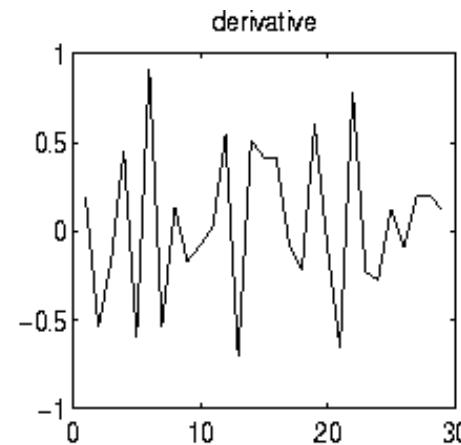
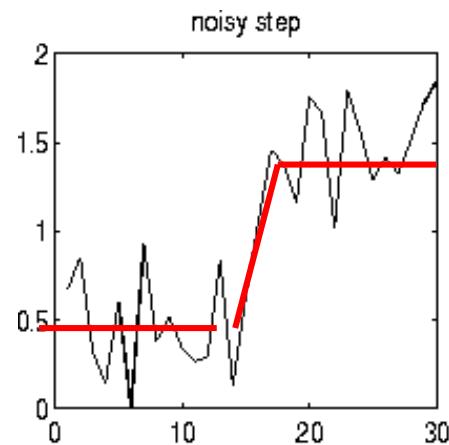
mientras que los bordes tipo línea aparecen como bordes tipo techo.



El ruido en una imagen provoca también variaciones de intensidad, lo cual resulta en bordes espurios. Un buen detector de bordes debería ser capaz de suprimir la mayoría de estos bordes espurios

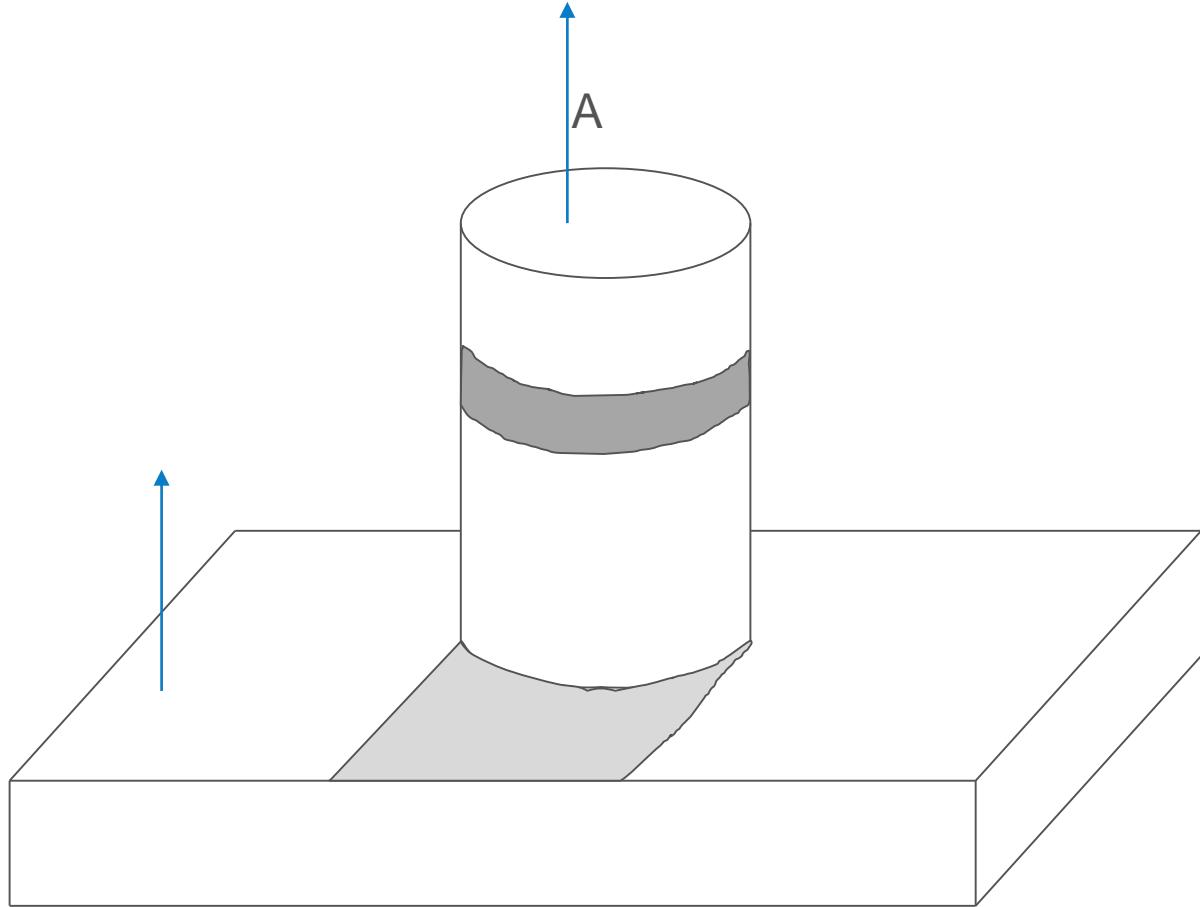


El ruido en una imagen provoca también variaciones de intensidad, lo cual resulta en bordes espurios. Un buen detector de bordes debería ser capaz de suprimir la mayoría de estos bordes espurios



Fuentes provocadoras de bordes:

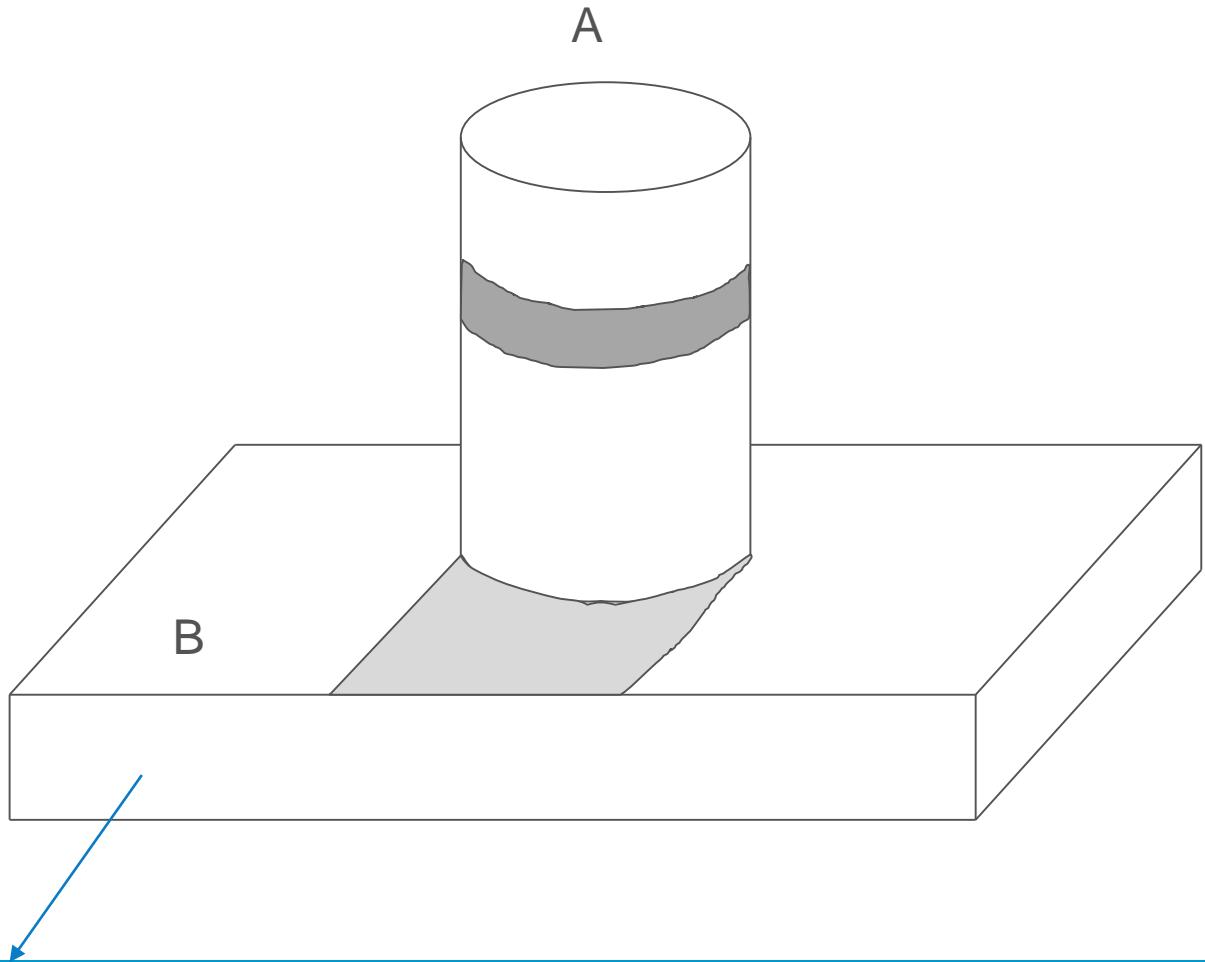
Bordes A



Fuentes provocadoras de bordes:

Bordes A

Bordes B

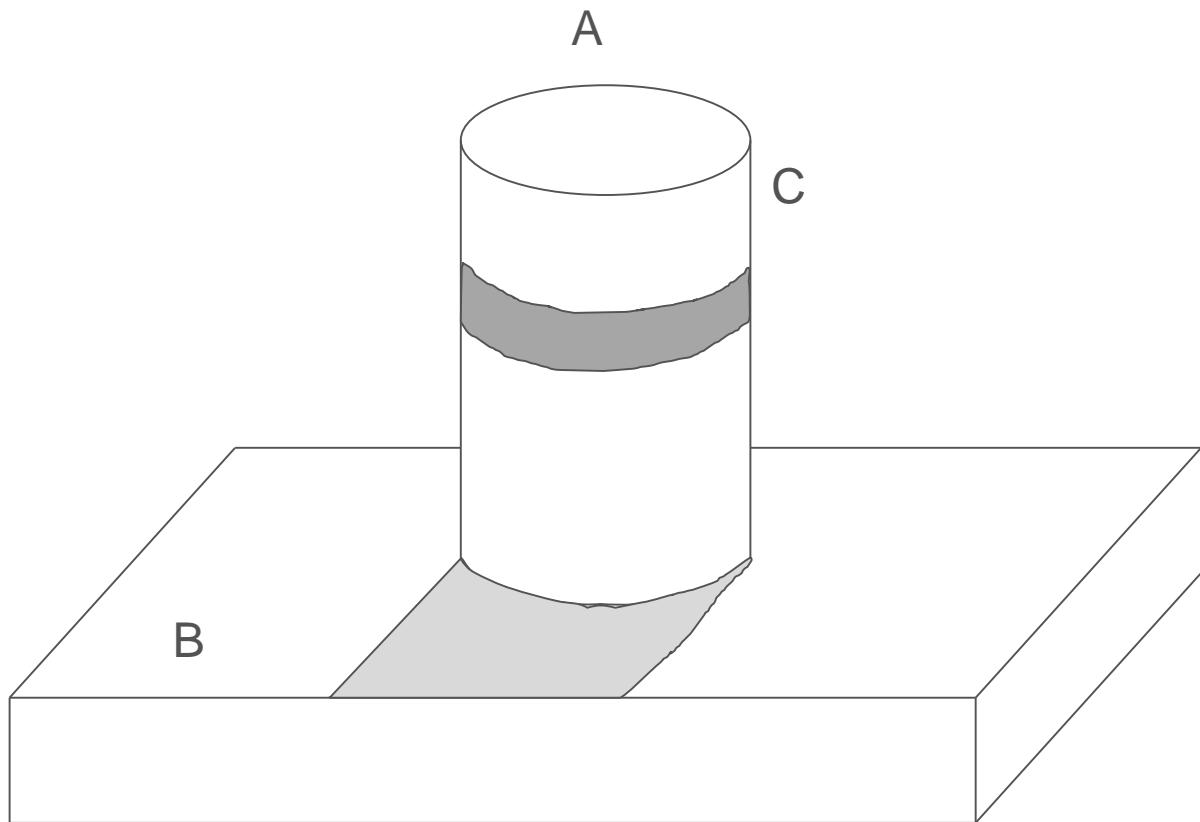


Fuentes provocadoras de bordes:

Bordes A

Bordes B

Bordes C



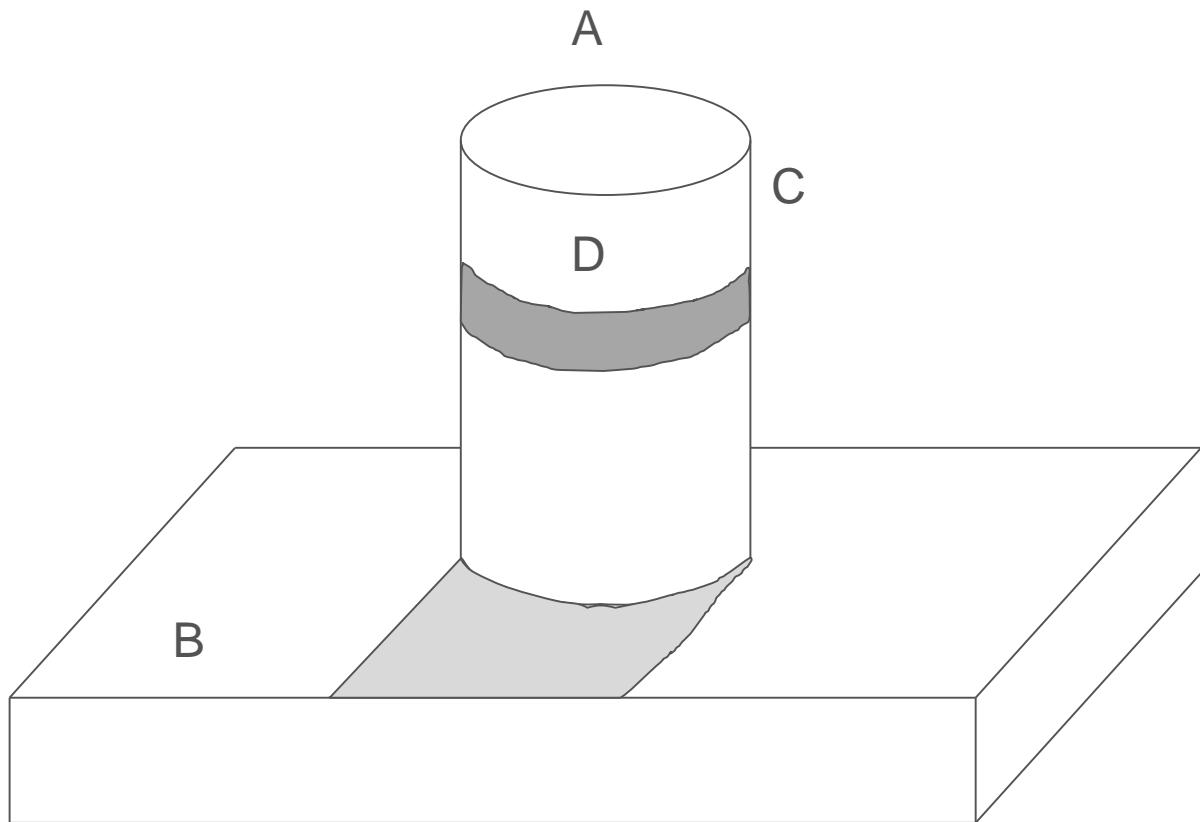
Fuentes provocadoras de bordes:

Bordes A

Bordes B

Bordes C

Bordes D



Fuentes provocadoras de bordes:

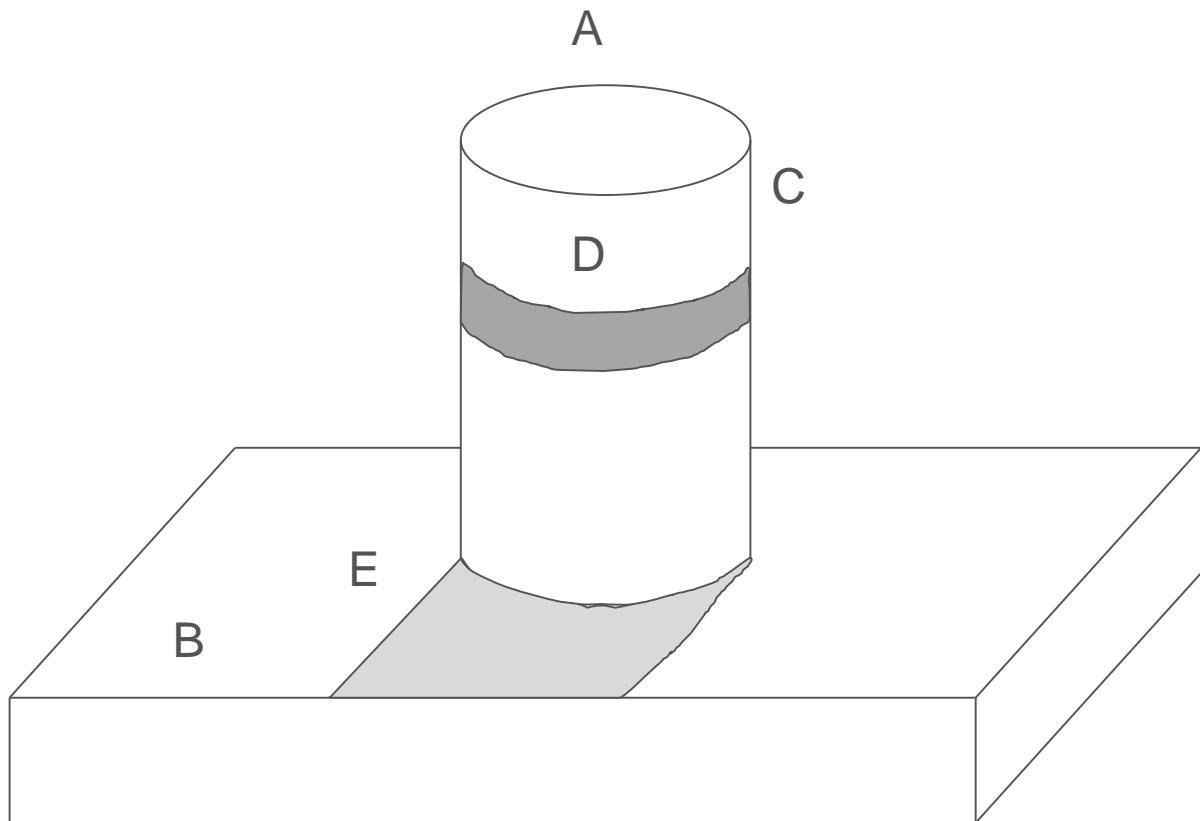
Bordes A

Bordes B

Bordes C

Bordes D

Bordes E



Fuentes provocadoras de bordes:

Bordes A

Bordes B

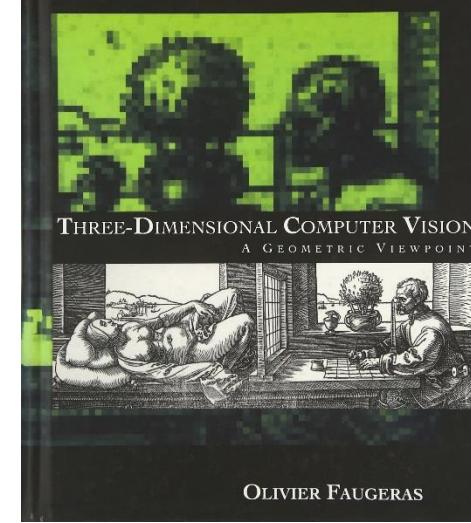
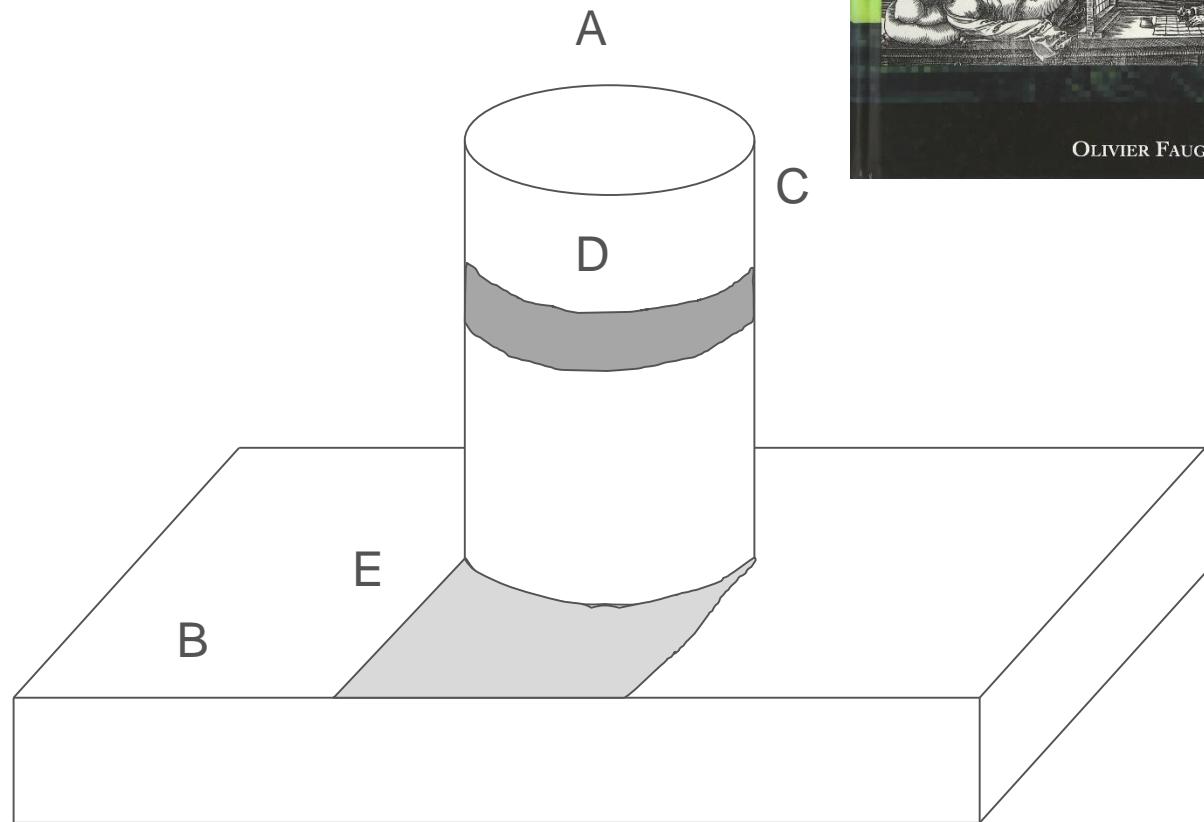
Bordes C

Bordes D

Bordes E

Bordes F (textura)

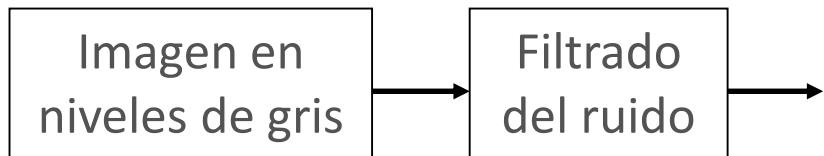
Etcétera.



Proceso de detección de bordes:

Proceso de detección de bordes en una imagen:

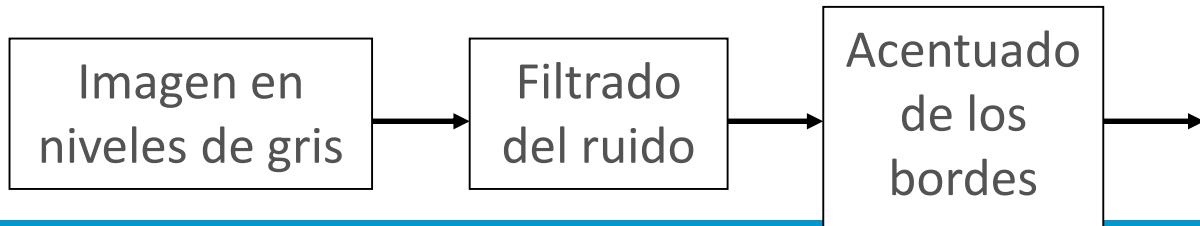
Alisado de ruido. El objetivo de esta etapa es suprimir tanto ruido como sea posible de la imagen, sin destruir bordes verdaderos.



Proceso de detección de bordes en una imagen:

Alisado de ruido. El objetivo de esta etapa es suprimir tanto ruido como sea posible de la imagen, sin destruir bordes verdaderos.

Mejoramiento de bordes. Por filtrado.



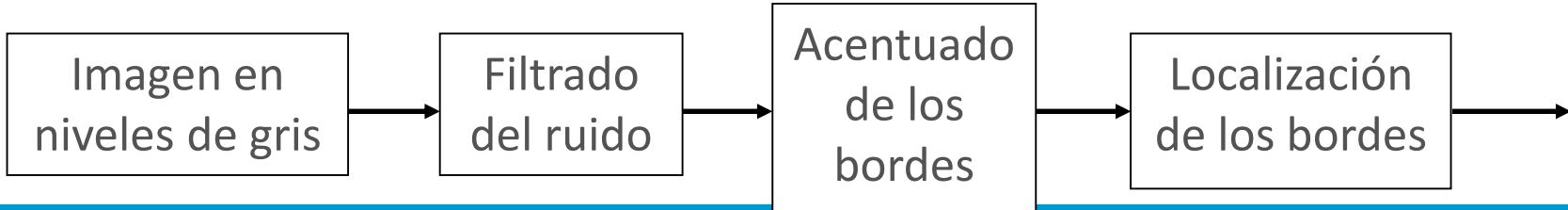
Proceso de detección de bordes en una imagen:

Alisado de ruido. El objetivo de esta etapa es suprimir tanto ruido como sea posible de la imagen, sin destruir bordes verdaderos.

Mejoramiento de bordes. Por filtrado.

Localización del borde. Decidir cuáles de los máximos producidos por un filtro deben ser considerados como verdaderos bordes y cuáles son causados por el ruido:

- Supresión de no máximos.
- Umbralado.



Alisado de ruido



Mejoramiento del bordes:

Detectores de bordes basados en EL CÁLCULO DE la primera derivada: El gradiente:

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$G[f(x, y)] = \sqrt{G_x^2 + G_y^2}$, iguala la tasa máxima de crecimiento de:

$$|\mathbf{G}[f(x, y)]|$$

Aproximaciones:

$$|\mathbf{G}[f(x, y)]| = |G_x| + |G_y| \quad |\mathbf{G}[f(x, y)]| = \max(|G_x|, |G_y|)$$

Otra cantidad importante: $\theta(x, y) = \arctan \frac{G_y}{G_x}$
donde θ se mide con respecto al eje x.

PRIMERA APROXIMACIÓN DEL GRADIENTE:

$$G_x = \frac{\partial f(x, y)}{\partial x} = \Delta_x f(x, y) \cong f(x+1, y) - f(x, y)$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = \Delta_y f(x, y) \cong f(x, y+1) - f(x, y)$$

máscaras de convolución:

$$G_x = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ y } G_y = [-1 \quad 1]$$

producen valores en la posición interpolada

$$(x + 1/2, y + 1/2)$$

(x, y)

$(x, y + 1)$

$(x + 1, y)$

SEGUNDA APROXIMACIÓN: GRADIENTE CRUZADO DE ROBERTS:

$$G[f(x, y)] = |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$

$$G[f(i, j)] = |G_x| + |G_y|$$

Máscaras de convolución:

5	10
5	10

1	0
0	-1

0	1
-1	0

L. G. Roberts,
Machine perception
of three-dimensional
solids. PhD Thesis.
MIT.

también en este caso las diferencias son computadas en la posición interpolada:

$$(x+1/2, y+1/2)$$



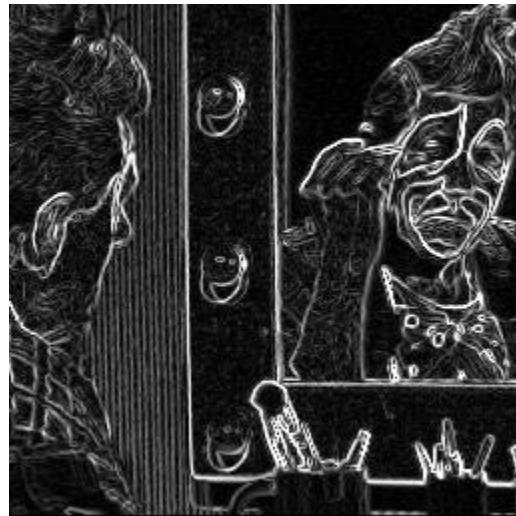
Imagen



Magnitud gradiente Roberts



Imagen



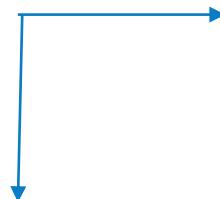
Salida



Resultado de umbralado con 80

EL MEJORADOR O ENFATIZADOR DE SOBEL:

a_0	a_1	a_2
a_7	(x,y)	a_3
a_6	a_5	a_4



Magnitud: $M = \sqrt{s_x^2 + s_y^2}$

derivadas parciales s_x y s_y son calculadas como sigue:

$$s_x = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

$$s_y = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

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

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

La constante c toma el valor de 2.

2	7	7
2	7	7
2	7	7



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

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

$$\text{Dir. } x: -2 - 14 - 7 + 2 + 14 + 7 = 0$$

$$\text{Dir. } y: -2 - 4 - 2 + 7 + 14 + 7 = 20$$

La segunda máscara dio un valor más alto porque su valor indica la posición por dónde pasa un borde en la vertical.



Imagen de entrada

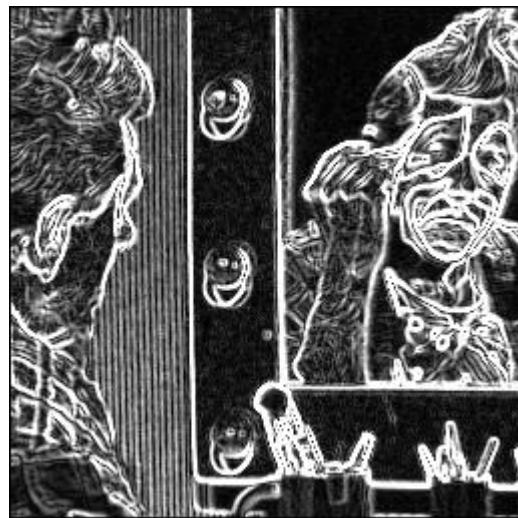


Imagen de magnitud



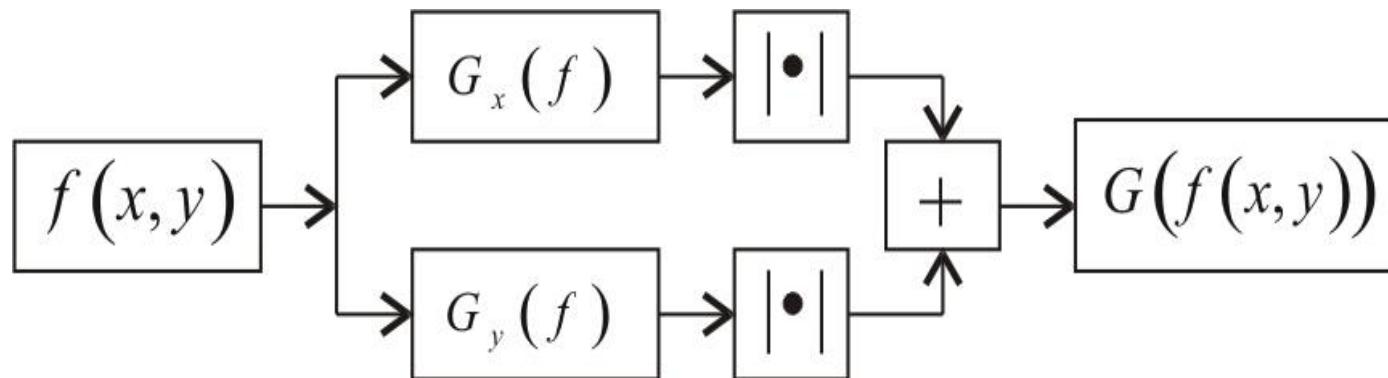
Imagen binaria

EL MEJORADOR DE PREWITT:

usa las mismas ecuaciones que el operador de Sobel, excepto que la constante $c = 1$:

$$\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$



Proceso gráfico para la detección de bordes en una imagen.

Una **propiedad interesante** tanto del operador de Sobel como el de Prewitt es que ambos incorporan una etapa de **alisado de ruido**, cosa que no pasa con el gradiente cruzado de Roberts.

Ambos operadores son el resultado de la convolución de:

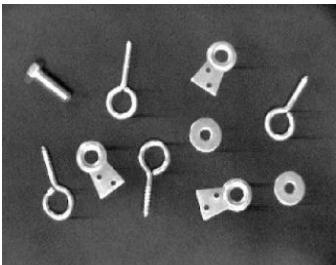
1	2	1
1	2	1

Con: $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

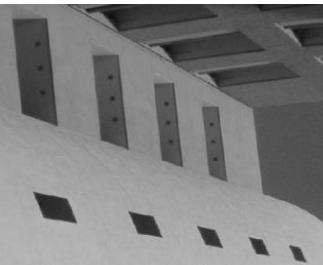
Para obtener:

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

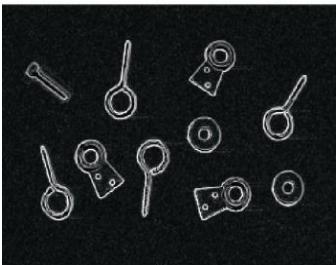
La primera máscara, como se puede ver, es un filtro paso-bajo mientras que la segunda es la ya vista diferencia en la dirección x .



(a)



(b)

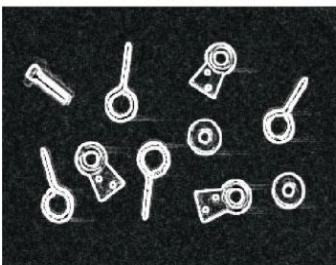


(c)



Roberts

(f)

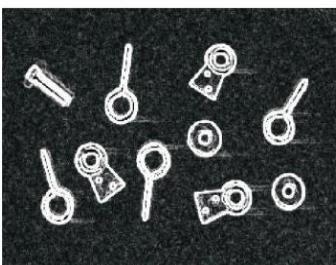


(d)

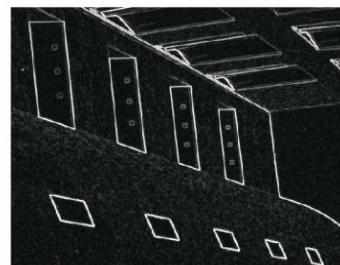


Prewitt

(g)



(e)



Sobel

(h)

(a) y (b) dos imágenes, una sencilla y una complicada;

(c-h) efecto de aplicar los filtros de Roberts, Sobel y Prewitt sobre dichas imágenes

% Programa para detectar siluetas de objetos:

```
clc % limpia pantalla.  
clearvars; % remueve todas las variables.  
close all;% cierra gráficas, archivos y recursos abiertos.  
  
f=imread('6 objetos.JPG'); % lee imagen de prueba f(x,y).  
figure(1)  
imshow(f) % muestra imagen de prueba.  
title('Imagen original')  
[n,m]=size(f); % dimensiones de la imagen f(x,y).  
%g=zeros(n,m);  
%g=f;  
%disp(n); % filas  
%disp(m); % columnas
```

Sobel

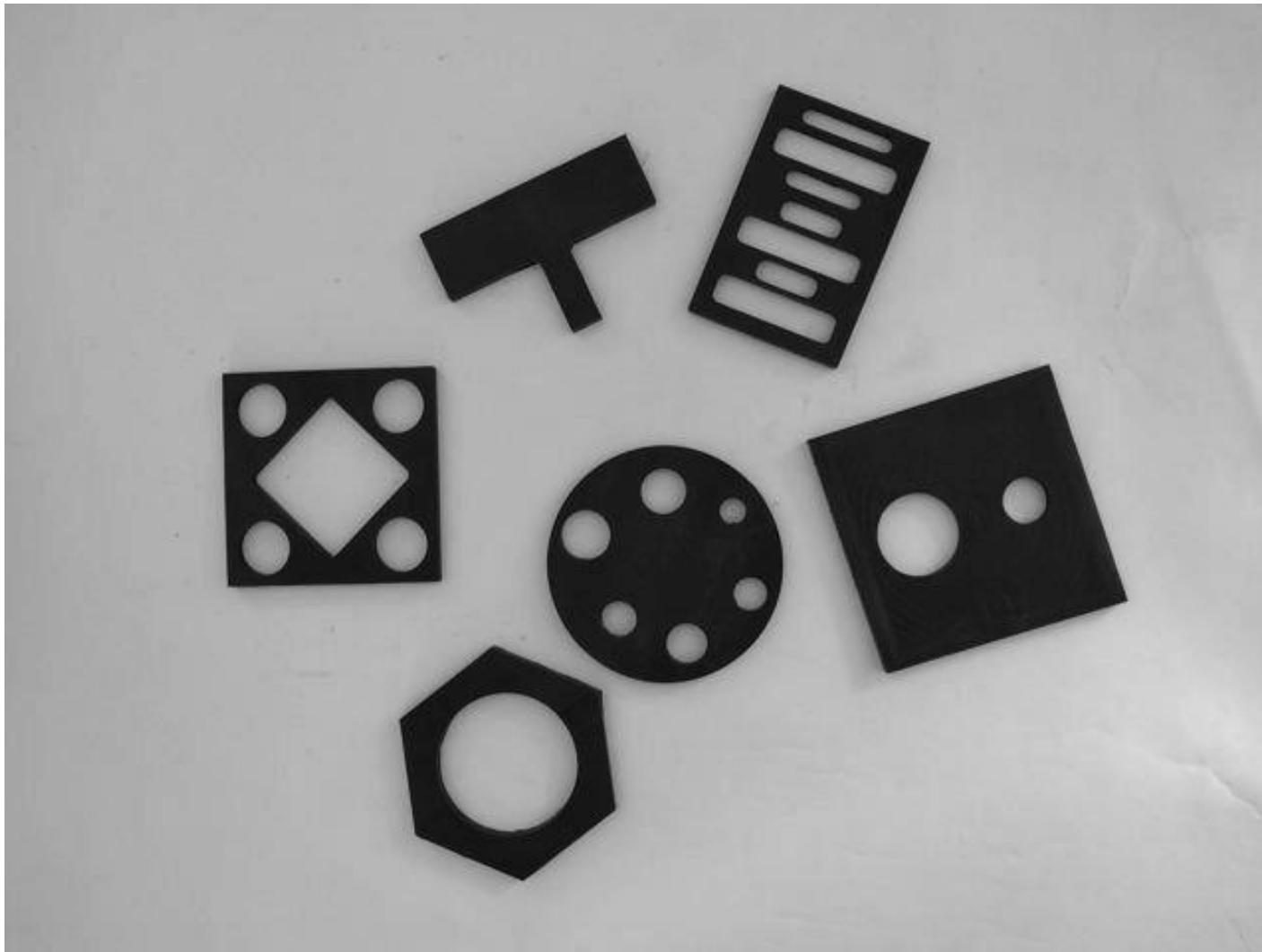


Imagen original

```
BW1 = edge(f, 'sobel', 'nothinning');  
% 'nothinning' omite adelgazamiento  
  
%BW1 = edge(f, 'sobel', threshold); devuelve bordes más  
fuertes que th.
```

%BW = edge(___, 'nothinning') omite la etapa de
adelgazamiento de bordes.

```
figure(2)  
imshow(BW1)  
title('Filtro Sobel');
```

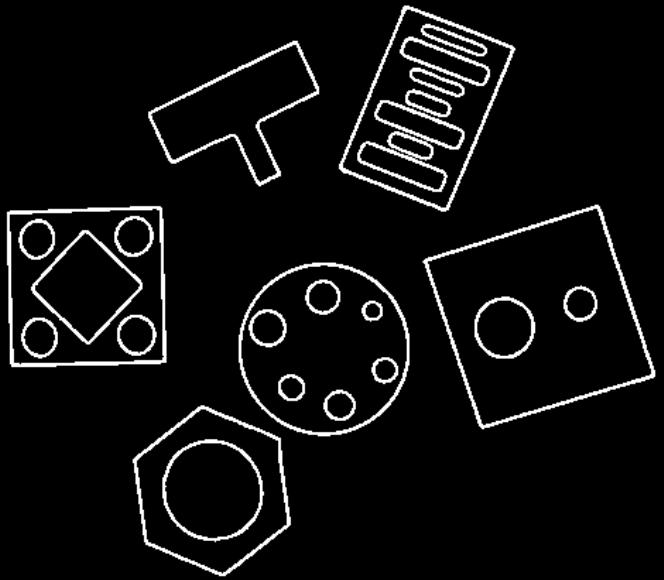


Imagen Sobel

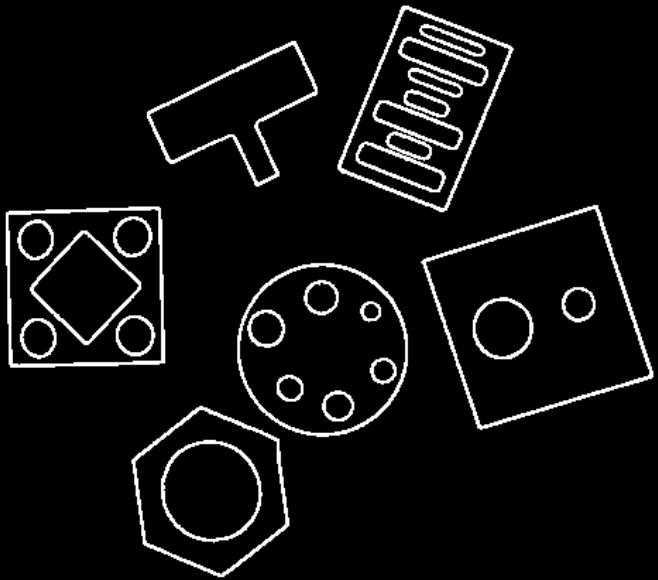
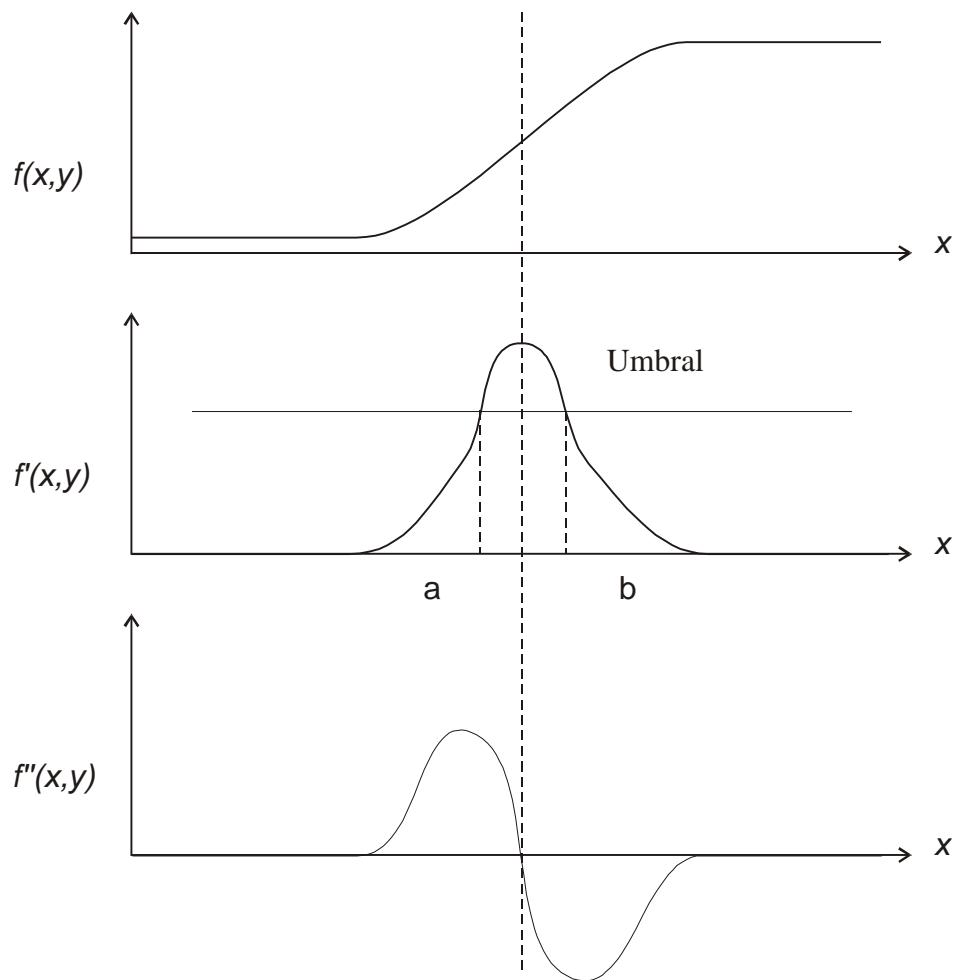


Imagen Prewitt

Detectores de bordes basados en el cálculo de segundas derivadas:

Detectores de bordes basados en el cálculo de la segunda derivada:



Si se usa un umbral para la detección de bordes, todos los puntos entre a y b serán marcados como puntos tipo borde. Sin embargo, al remover puntos que no son máximos locales en la primera derivada, los bordes pueden ser detectados de manera más precisa. El máximo local en la primera derivada corresponde al cruce por cero en la segunda derivada.

EL OPERADOR LAPLACIANO:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Aproximaciones de las derivadas: $\frac{\partial^2 f}{\partial x^2}$ $\frac{\partial^2 f}{\partial y^2}$

centradas alrededor del píxel (x, y) :

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - 2f(x, y) + f(x-1, y)$$

	$(x-1, y)$	
$(x, y-1)$	(x, y)	$(x, y+1)$
	$(x+1, y)$	

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Estas dos aproximaciones pueden ser aplicadas a través de las siguientes máscaras de convolución:

$$\frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad \frac{\partial^2 f}{\partial y^2} = [1 \quad -2 \quad 1]$$

pueden combinarse en una sola:

0	1	0
1	-4	1
0	1	0

El operador laplaciano emite la presencia de un borde cuando su salida causa un cruce por cero en la señal.

$$\frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial y^2} = [1 \quad -2 \quad 1]$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

0	1	0
1	-4	1
0	1	0

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

0	1	0
1	-4	1
0	1	0

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

$$1+1+5+1-4=+4$$

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

0	1	0
1	-4	1
0	1	0

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

$$1+1+5+1-4=+4$$

$$1+5+9+5-20=0$$

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

0	1	0
1	-4	1
0	1	0

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

$$1+1+5+1-4=+4$$

$$1+5+9+5-20=0$$

$$9+5+9+9-36=-4$$

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

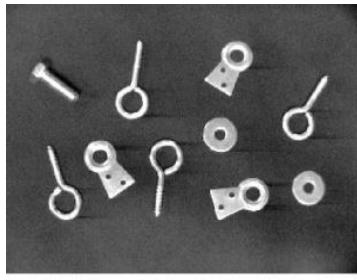
0	1	0
1	-4	1
0	1	0

1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9
1	1	5	9	9

$$1+1+5+1-4=+4$$

$$1+5+9+5-20=0 \text{ (cruce por cero)}$$

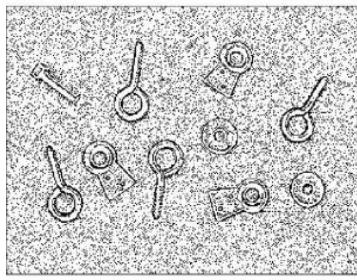
$$9+5+9+9-36=-4$$



(a)

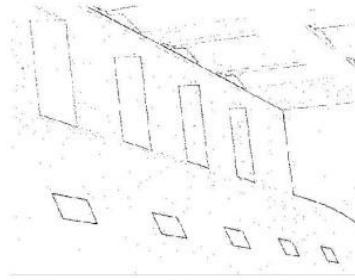


(b)

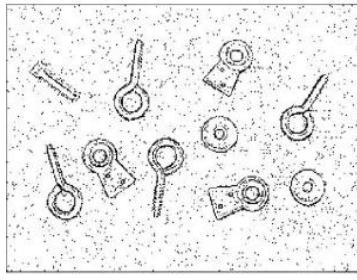


(c)

$$\nabla^2 f \geq 20$$

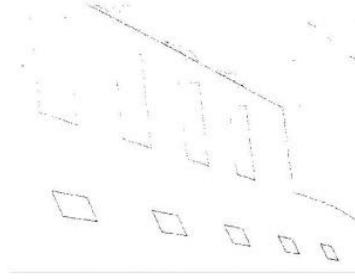


(f)

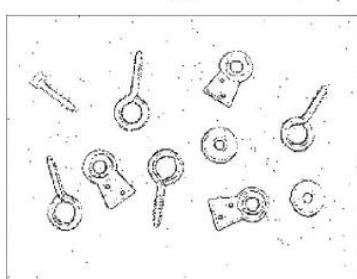


(d)

$$\nabla^2 f \geq 40$$

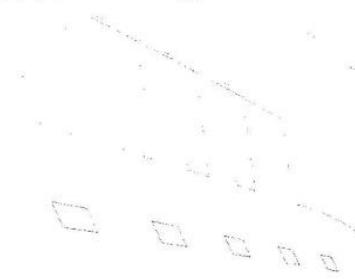


(g)



(e)

$$\nabla^2 f \geq 60$$



(h)

(a) y (b) Dos imágenes.

(c-h) Cruces por cero detectados por el Laplaciano para varios valores de del umbral

Nota: Rara vez, el Laplaciano se utiliza sólo.

¡¡Porque es un doble amplificador del ruido!!

EL LAPLACIANO DEL GAUSSIANO (LDG) (D. Marr):

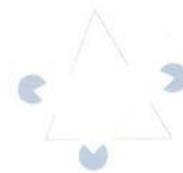
los puntos de borde detectados al encontrar los cruces por cero de la segunda derivada de la imagen **son muy sensibles al ruido.**

Es, por tanto, deseable filtrar el ruido antes de acentuar los bordes. Para lograr esto, el operador **laplaciano del gaussiano (LdG)**

$$h(x, y) = \nabla^2 [g(x, y) * f(x, y)]$$

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

VISION

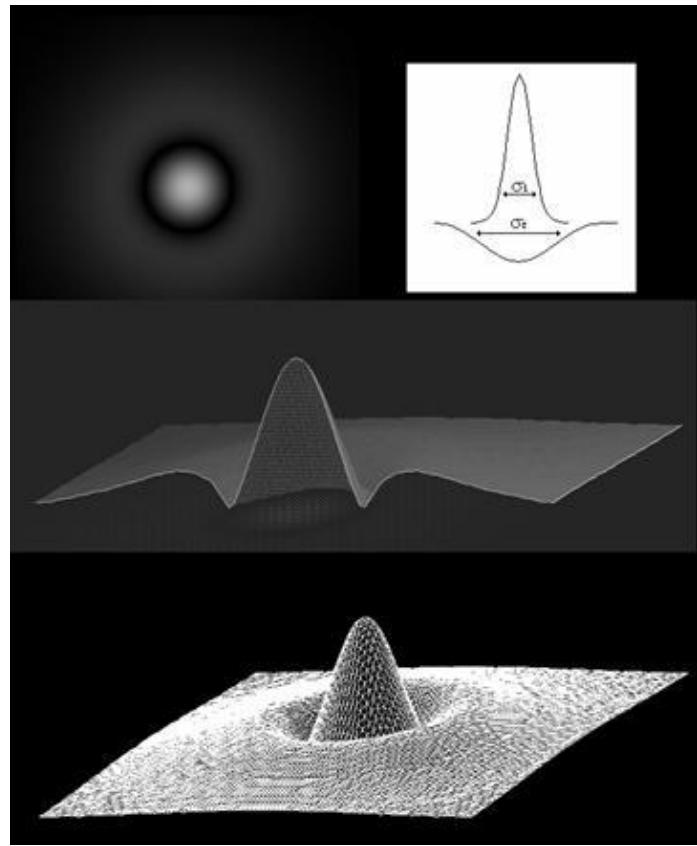
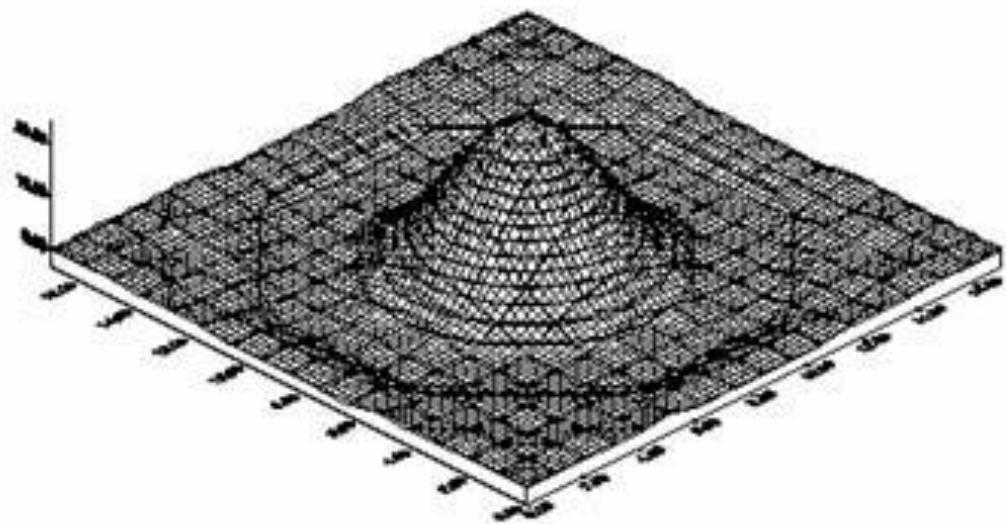


Teoría de la visión de D. Marr

DAVID MARR

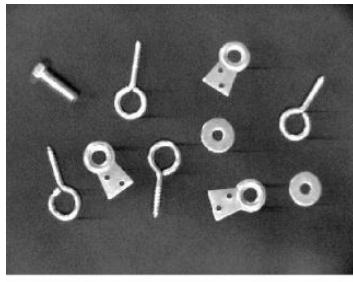
Al usar la regla de la derivada para la convolución:

$$h(x, y) = [\nabla^2 g(x, y)] * f(x, y) \quad \nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Luego, entonces, los siguientes dos métodos son equivalentes:

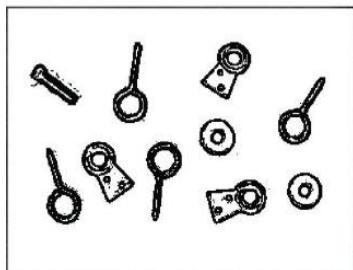
- 1 Convolucionar la imagen con un filtro alisador Gaussiano, y al resultado aplicarle enseguida el Laplaciano.
- 2 Convolucionar la imagen con el filtro lineal Laplaciano del gaussiano.



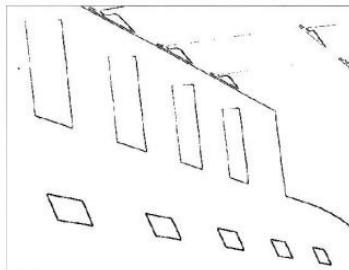
(a)



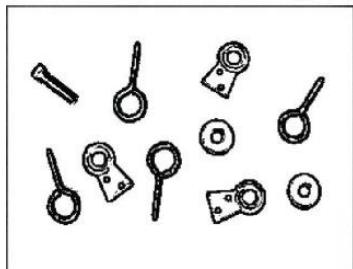
(b)



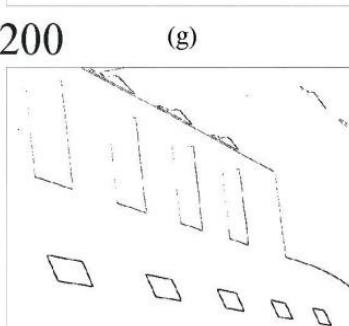
(c)

 $u \geq 150$ 

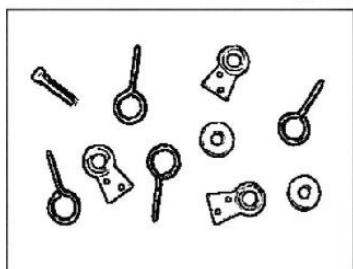
(f)



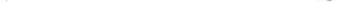
(d)

 $u \geq 200$ 

(g)



(e)

 $u \geq 250$ 

(h)

(a) y (b) Dos imágenes.

(c-f) Cruces por cero detectados sobre estas imágenes para varios valores del umbral u .

Ejemplo de un LdG:

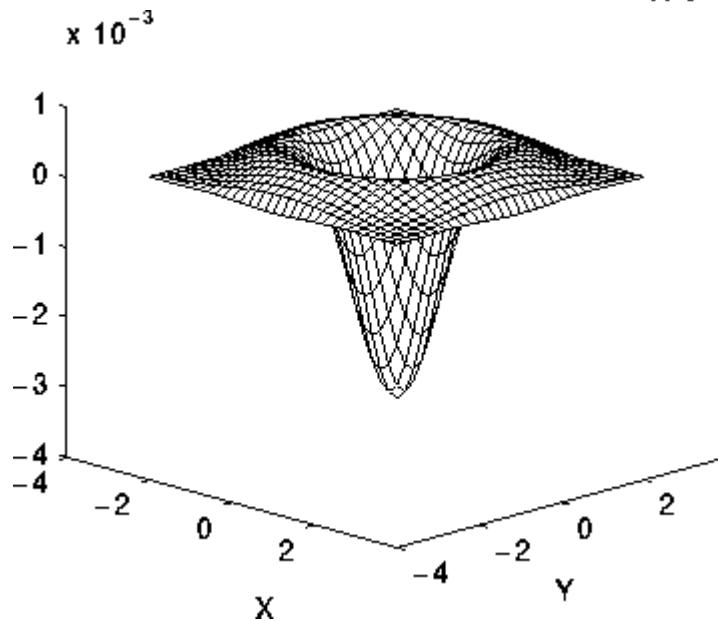
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Otra forma de ver el Laplaciano del Gaussiano:

Máscaras convolutivas:

$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Laplaciano del Gaussiano: $LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$



Una máscara discreta que aproxima el LdG con $\sigma = 1.4$ es la siguiente:

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

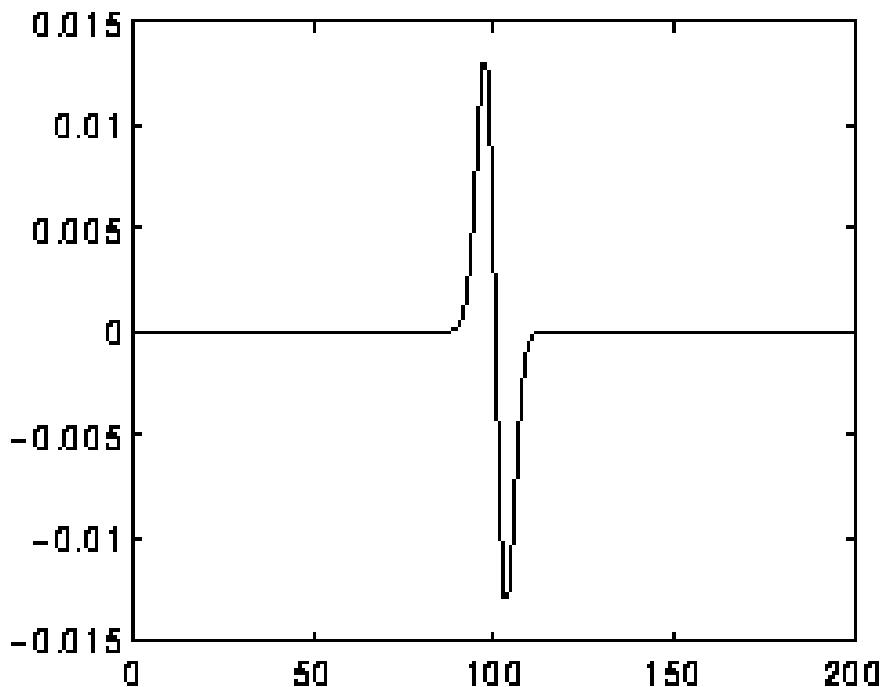
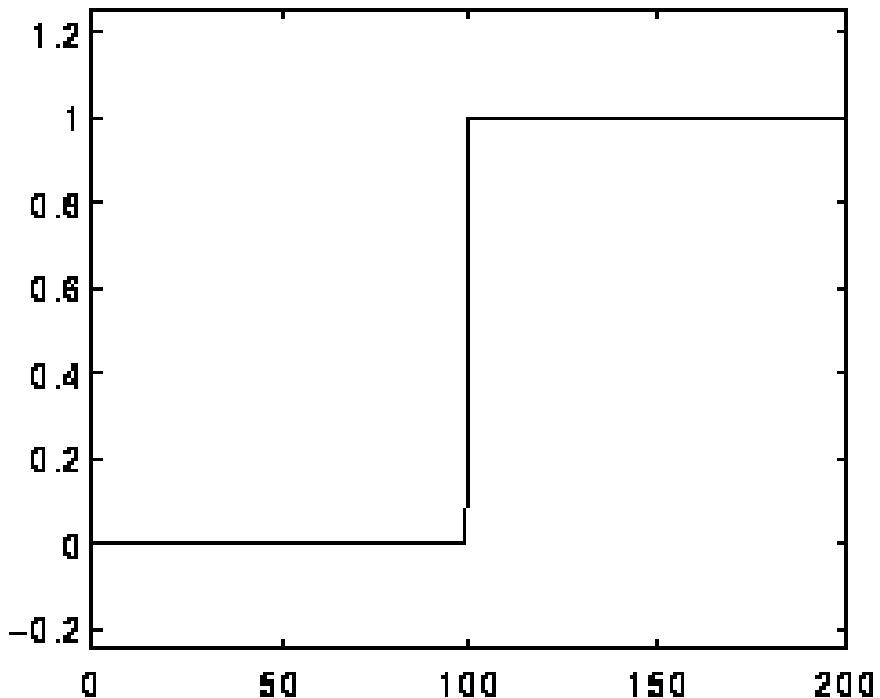
Respuesta del LoG ante un borde tipo paso:

Cero a distancias grandes del borde.

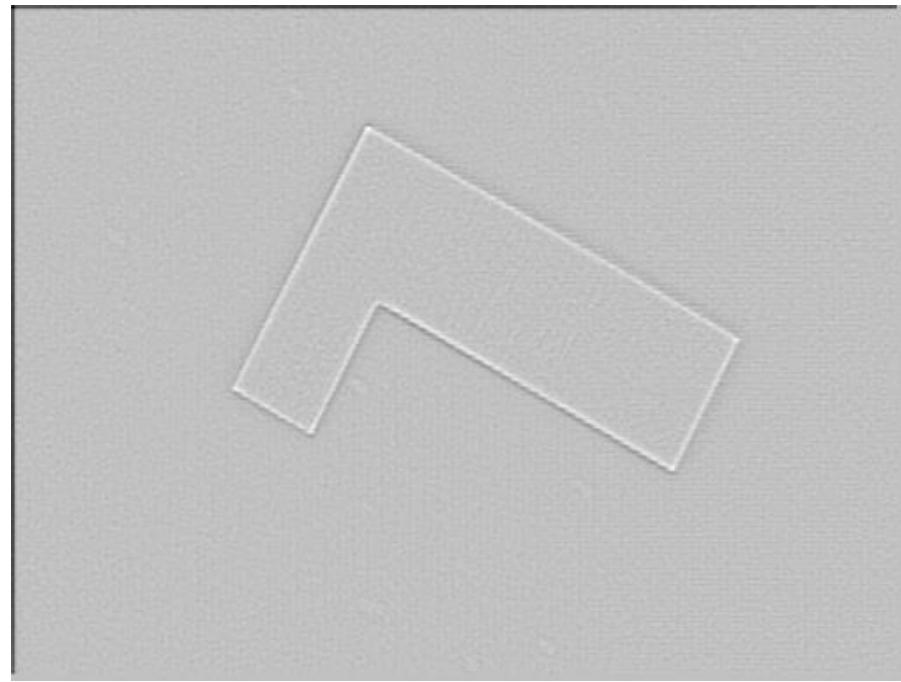
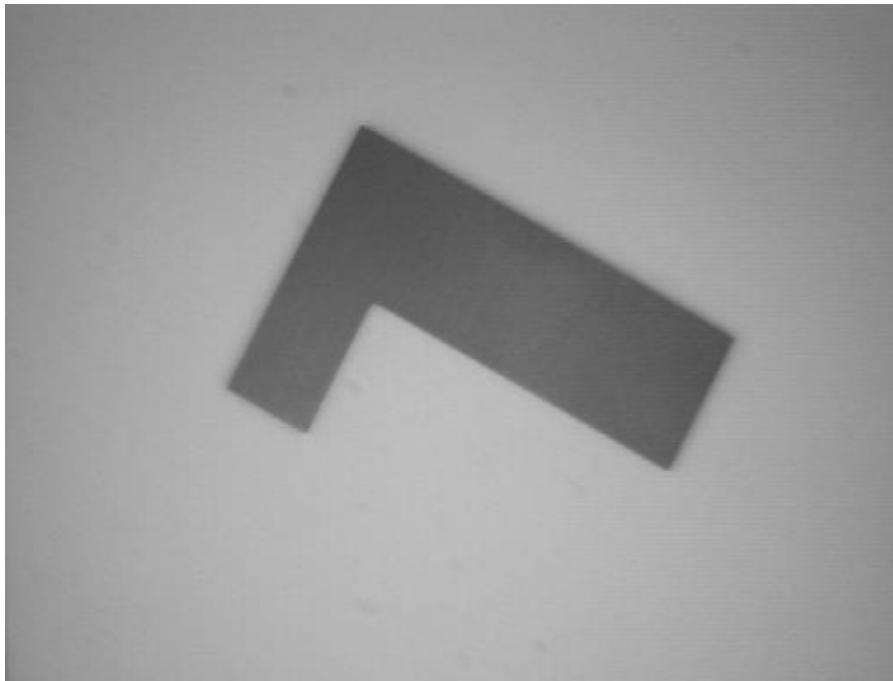
Positivo a un lado del borde.

Negativo al otro lado del borde.

Cero en un punto justo por donde sube el borde.



Resultado de aplicar el LoG de 7×7 y $\sigma = 1$



Otros resultados:



Resultado de aplicar un filtro LoG con una desviación
estándar de 1.0

Otros detectores de bordes (máscaras tipo compás):

Otros mejoradores de bordes:

Ventanas de Kirsch:

-3	-3	5
-3	0	5
-3	-3	5

K1

-3	5	5
-3	0	5
-3	-3	-3

K2

5	5	5
-3	0	-3
-3	-3	-3

K3

5	5	-3
5	0	-3
-3	-3	-3

K4

5	-3	-3
5	0	-3
5	-3	-3

K5

-3	-3	-3
5	0	-3
5	-3	-3

K6

-3	-3	-3
-3	0	-3
5	5	5

K7

-3	-3	-3
-3	0	5
-3	5	5

K8

1	1	2	1	6
2	1	1	6	5
2	1	5	6	7
1	6	6	5	6
5	6	7	5	6

$$-3-3+30-3+0+30-18-18+25=40$$

El valor del módulo del **gradiente** y su dirección se obtienen como sigue:

Se aplican las ocho máscaras sobre ese punto, como resultado se obtienen ocho valores. Se obtienen los valores absolutos de estos ocho valores. El valor más grande determina el valor del gradiente.

La dirección del gradiente en p se obtiene de manera directa por el valor asociado a la máscara: {0,45,90,**135**,180,225,270,315}.

Ejemplo:

1	1	2	1	6
2	1	1	6	5
2	1	5	6	7
1	6	6	5	6
5	6	7	5	6

Al aplicar los dos pasos antes dados, se tiene: 40, 8, 32, **72**, 32, 8, 40, 40.

Se puede ver que el valor máximo recae en la ventana K4.

La dirección a tomar en cuenta es de 135 grados.

Máscaras de Robinson:

R1	<table border="1"><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1
-1	0	1								
-2	0	2								
-1	0	1								

R2	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>-1</td><td>0</td></tr></table>	0	1	2	-1	0	1	-2	-1	0
0	1	2								
-1	0	1								
-2	-1	0								

R3	<table border="1"><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table>	1	2	1	0	0	0	-1	-2	-1
1	2	1								
0	0	0								
-1	-2	-1								

R4	<table border="1"><tr><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>-2</td></tr></table>	2	1	0	1	0	-1	0	-1	-2
2	1	0								
1	0	-1								
0	-1	-2								

R5	<table border="1"><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>2</td><td>0</td><td>-2</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr></table>	1	0	-1	2	0	-2	1	0	-1
1	0	-1								
2	0	-2								
1	0	-1								

R6	<table border="1"><tr><td>0</td><td>-1</td><td>-2</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>2</td><td>1</td><td>0</td></tr></table>	0	-1	-2	1	0	-1	2	1	0
0	-1	-2								
1	0	-1								
2	1	0								

R7	<table border="1"><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1
-1	-2	-1								
0	0	0								
1	2	1								

R8	<table border="1"><tr><td>-2</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>2</td></tr></table>	-2	-1	0	-1	0	1	0	1	2
-2	-1	0								
-1	0	1								
0	1	2								

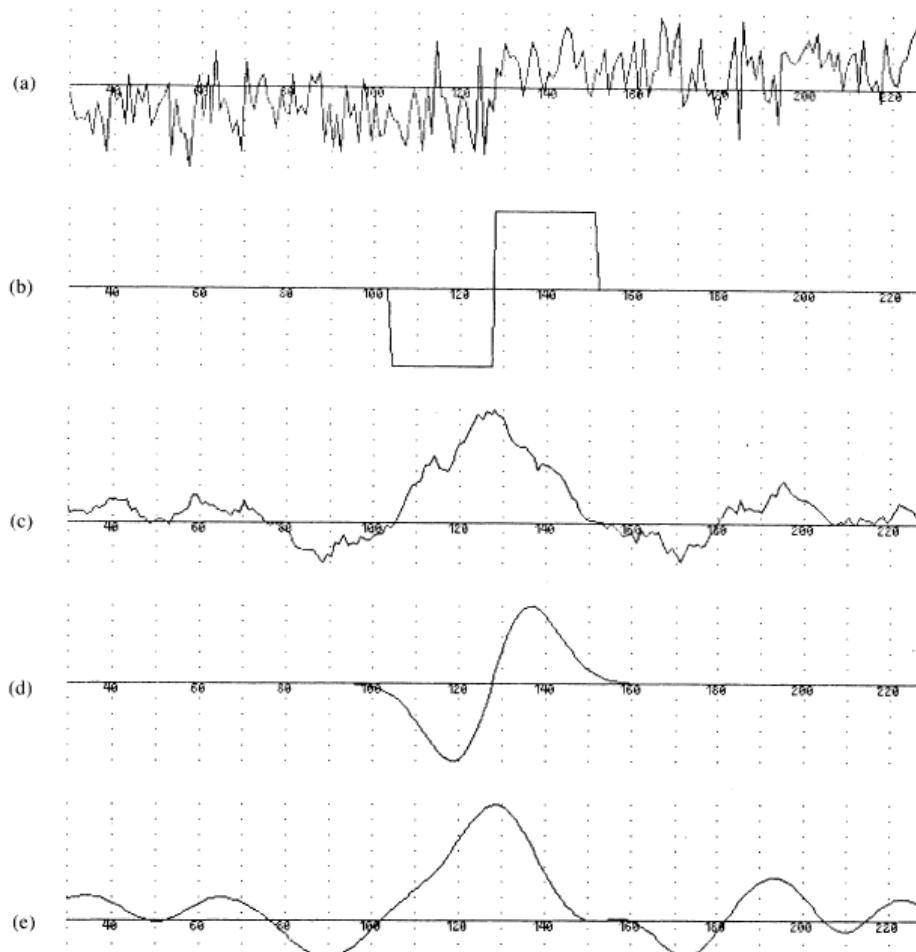
Ejemplo:

1	1	2	1	6
2	1	1	6	5
2	1	5	6	7
1	6	6	5	6
5	6	7	5	6

Al considerar la imagen del ejemplo anterior, tenemos los siguientes valores: 14, 0, 14, **19**, -14, 0, -14 y -19. De nuevo la máscara ganadora es de nuevo la número cuatro. La dirección del borde es también 135° .

El detector de bordes de Canny:

El problema:



Alisado

- a) Borde ruidoso. (b) Operador diferencia de cajas. (c) Resultado. (d) Primera derivada del Gaussiano. (e) Resultado

El detector de bordes de Canny es la primera derivada del Gaussiano:

Busca optimizar los siguientes tres criterios:

Buena detección. Maximiza la probabilidad de detectar bordes verdaderos y minimiza la probabilidad de detectar bordes falsos.

Buena localización. Ubica los bordes lo más cerca de los reales.

Una sola respuesta. Trata de detectar uno y solo un borde.

Canny demuestra que la primera derivada del Gaussiano de la señal optimiza estos tres criterios.

Criterios de detección y localización:

Si $f(x)$ es la respuesta del filtro y $G(x)$ es el borde y suponiendo que el borde aparece centrado en $x = 0$:

Respuesta del filtro viene dada por la convolución:

$$H_G = \int_{-W}^{+W} G(-x) f(x) dx$$

Si se asume que el filtro tiene una respuesta impulsiva acotada en $[-W, W]$, la respuesta del cuadrado de la raíz de la media al ruido: $n(x)$:

$$H_n = n_0 \left[\int_{-W}^{+W} f^2(x) dx \right]^{1/2}$$

El primer criterio, el cociente señal a ruido viene dado como el cociente de las dos respuestas anteriores:

$$\text{SNR} = \frac{\left| \int_{-W}^{+W} G(-x) f(x) dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f^2(x) dx}}$$

El segundo criterio de localización viene dado por el siguiente cociente:

$$\text{Localization} = \frac{\left| \int_{-W}^{+W} G'(-x) f'(x) dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f'^2(x) dx}}.$$

El diseño del detector se reduce a la maximización de estos dos criterios en forma simultanea. Esto se logra al maximizar el producto de ambos:

$$\frac{\left| \int_{-W}^{+W} G(-x) f(x) dx \right| \left| \int_{-W}^{+W} G'(-x) f'(x) dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f^2(x) dx} \sqrt{\int_{-W}^{+W} f'^2(x) dx}}.$$

J. Canny. A Computational Approach to Edge Detection. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986

Consiste en los siguientes pasos:

- Alisado la imagen de entrada:** Esto lo hace mediante un filtro paso-bajo.
- Obtención del gradiente:** Se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión de no máximos:** Se adelgaza el ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Aplicación de un doble umbral:** Se aplican dos umbrales; se busca reducir la aparición de contornos falsos.
- Seguimiento de bordes mediante histéresis:** Se enlazan bordes verdaderos.

1 Alisar imagen $f(x, y)$ por convolución con un filtro

Gaussiano: $s(x, y) = f(x, y) * h(x, y)$. Ejemplo:

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$



2 Para cada píxel :

a) Calcular imágenes de gradiente en las direcciones x y y :

G_x y G_y ,

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

b) Estimar la magnitud (fuerza) del borde:

$$G[s(x, y)] = \sqrt{G_x^2 + G_y^2}$$

c) Estimar la orientación de la normal al borde como:

$$\theta(x, y) = \arctan \frac{G_y}{G_x}$$



Imagen



Imagen alisada



Imagen de magnitudes

LOCALIZACIÓN DE BORDES:

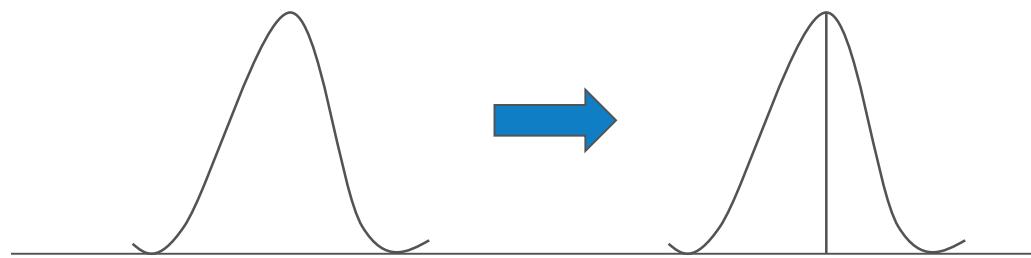
Adelgazado de bordes gruesos a borde de grosor de un píxel
(supresión de no máximos).

Selección de un valor mínimo que permite declarar un máximo local como borde (**umbralado**).

SUPRESIÓN DE NO MÁXIMOS:

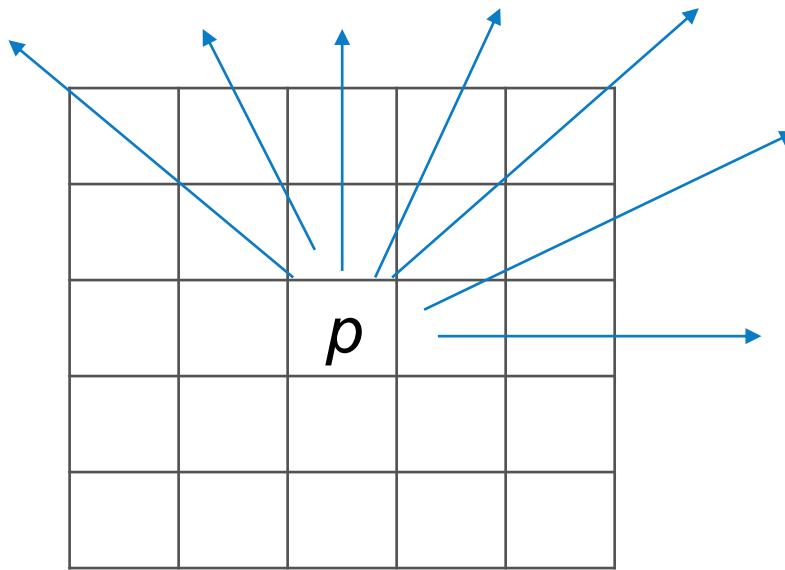
Algoritmo para la supresión de no máximos por adelgazamiento.

Borde grueso



Este algoritmo recibe como entrada las imágenes de magnitud y de dirección $G[s(x, y)]$ y $\theta(x, y)$, produciendo como salida una imagen $N(x, y)$ con los **no máximos suprimidos**.

Consideremos las siguientes ocho direcciones: d_1, d_2, \dots, d_8 representando las orientaciones a $0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ$ y 157.5°



Para cada píxel (x, y) en la imagen de magnitudes:

1 Encontrar la dirección, \hat{d}_k , que mejor aproxime la dirección de la normal al borde $\theta(x, y)$.

Ejemplo: $\theta(x, y) = 43$, entonces $\hat{d}_k = 45$.

2 Sí el valor $G[s(x, y)]$ es **más pequeño** que al menos uno de sus dos puntos de borde vecinos dentro de la vecindad, a lo largo de \hat{d}_k , realizar la siguiente asignación

$N(x, y) = 0$ (supresión);

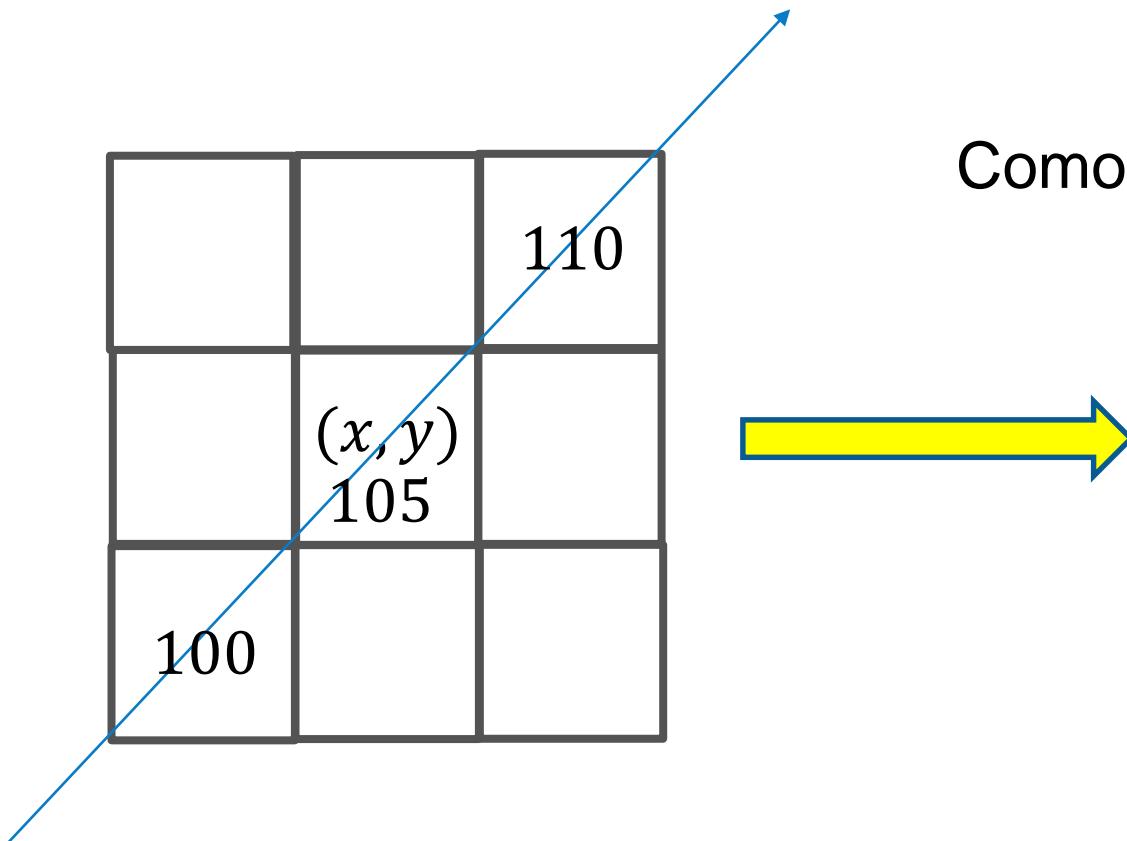
de otra forma $N(x, y) = G[s(x, y)]$.

La salida de este procedimiento, como se dijo arriba, es una imagen de bordes adelgazados $N(x, y)$ después de la supresión de no máximos locales.

Ejemplos:

$$\theta(x, y) = 43$$

Mejor aproximación: 45 grados



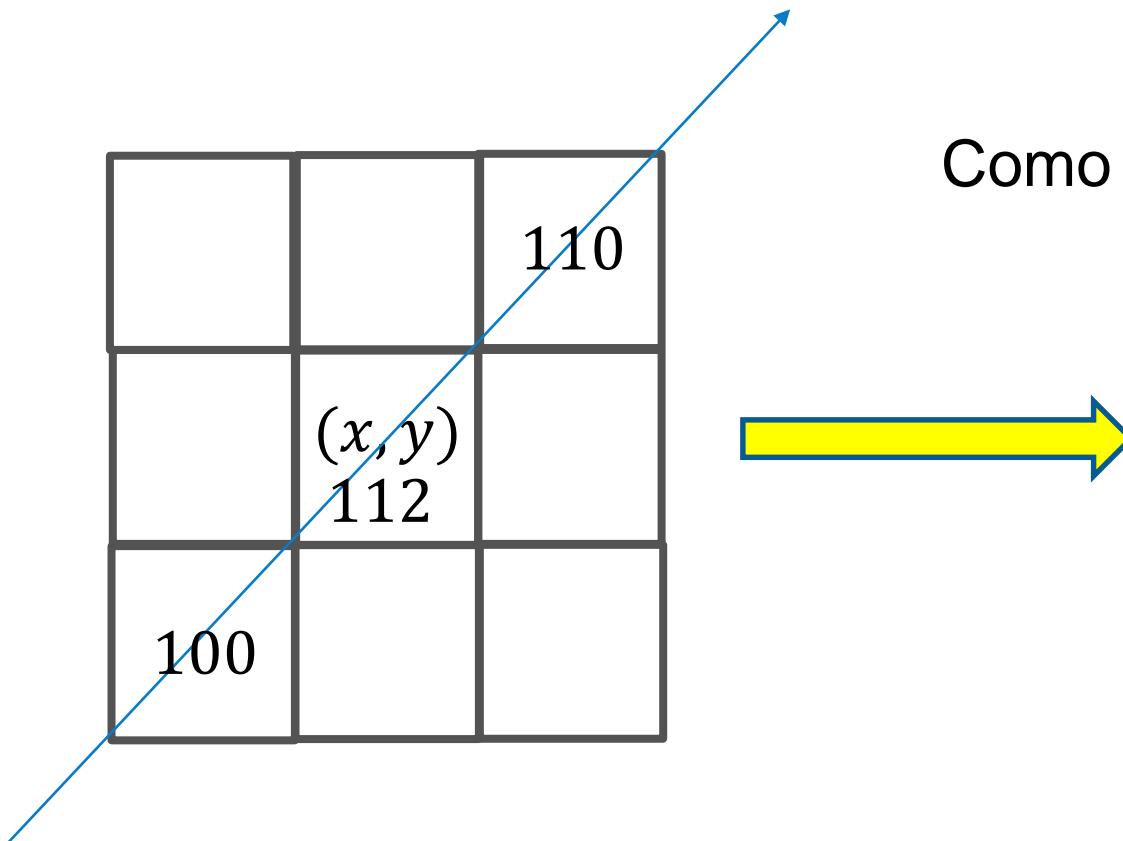
Como es **menor**, entonces:

$$N(x, y) = 0$$

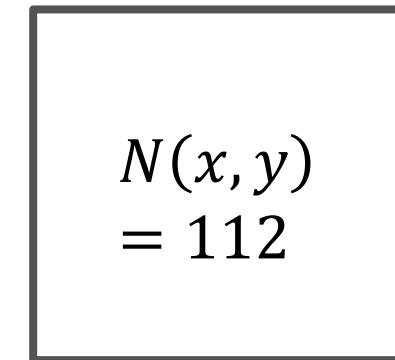
Ejemplos:

$$\theta(x, y) = 43$$

Mejor aproximación: 45 grados



Como es **mayor**, entonces:



Ejemplo: Las magnitudes de gradiente aparecen indicadas en colores y números, mientras que las direcciones aparecen con flechas.

1	3	5	4	3	1
↑	↑	↑	↑	↑	↑
2	4	6	5	5	5
↑	↑	↑	↑	↑	↑
3	6	7	7	6	4
↑	↑	↑	↑	↑	↑
4	5	6	6	4	3
↑	↑	↑	↑	↑	↗
2	4	4	5	3	2
↑	↑	↑	↑	↗	↗

Resultado del proceso de eliminación de no máximos:

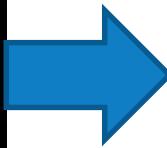


Imagen de magnitudes

$$N(x, y)$$

Bordes después de la
supresión de magnitudes

Doble umbralado:

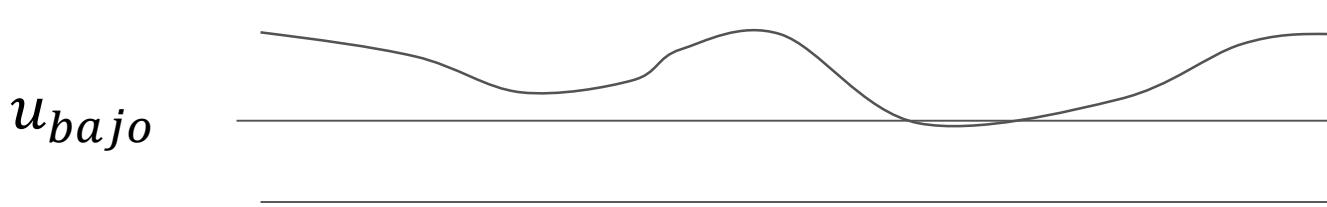
Los bordes restantes todavía aparecen marcados con sus magnitudes.

Muchos puede que sean **verdaderos (bordes con magnitud grande)** y otros **falsos (bordes con magnitud pequeña)**.

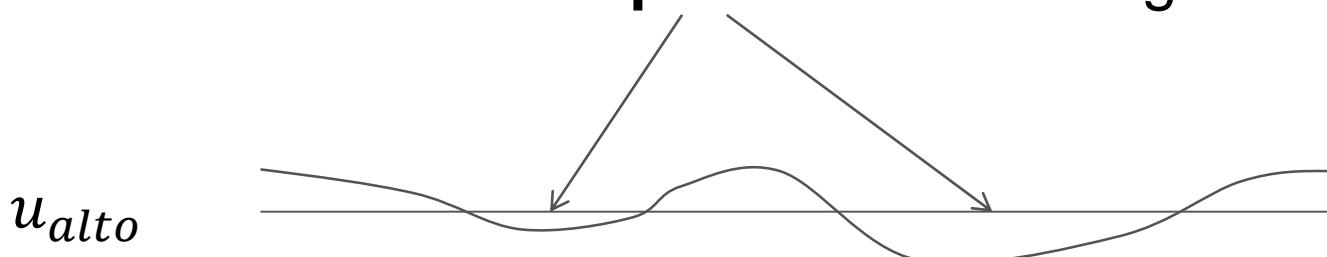
Una manera de eliminar a los falsos y mantener a los verdaderos es mediante un **umbralado**.

Doble umbralado:

1 Si el valor del umbral u es escogido como muy **bajo** con miras a dejar pasar verdaderos bordes débiles, puede causar también el que bordes debidos al ruido (contornos falsos) pasen junto con los verdaderos.



2 El valor de un verdadero máximo local puede presentar variaciones arriba y debajo del umbral, resultando en consecuencia en **rompimientos** a lo largo del borde.



Doble umbralado:

Una solución bastante conocida a ambos problemas consiste en usar **dos** umbrales (**histéresis**). Cualquier método de histéresis recibe como entrada la salida del supresor de no máximos, $N(x, y)$ y le aplica dos umbrales u_1 y u_2 de forma que $u_1 < u_2$, y produce como salida dos imágenes umbraladas $u_1(x, y)$ y $u_2(x, y)$.



u_1



$u_1(x, y)$

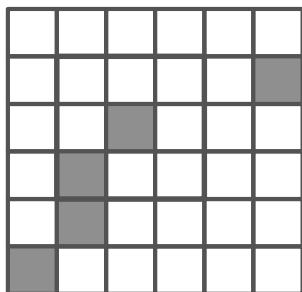
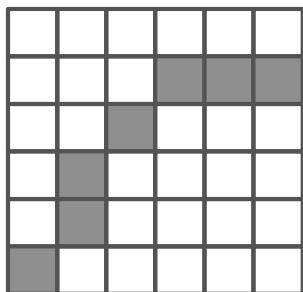
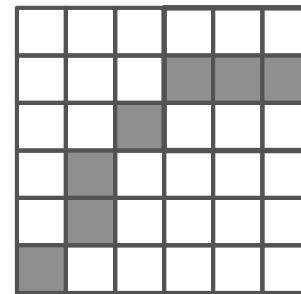
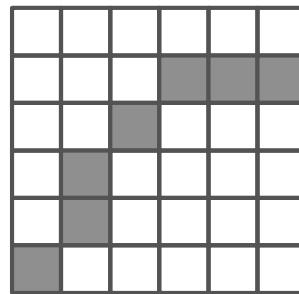
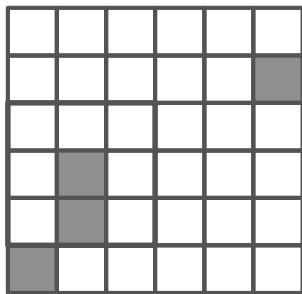
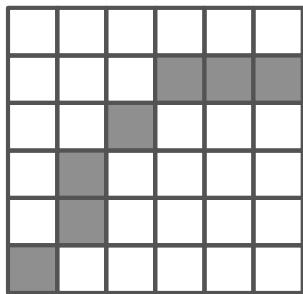
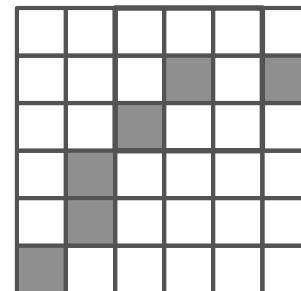
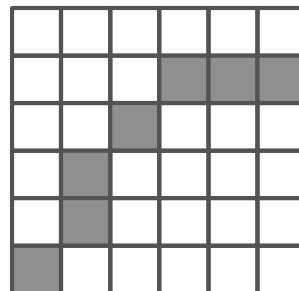
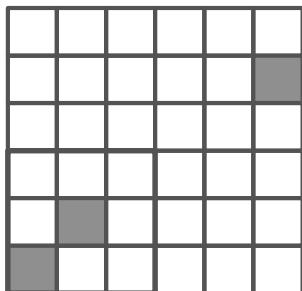
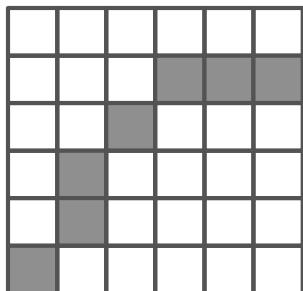
u_2



$u_2(x, y)$

El uso apropiado de ambas imágenes $u_1(x, y)$ y $u_2(x, y)$ permite producir una imagen con **menos** bordes ruidosos y **menos** rompimientos. Esto se logra a través de la **visita alternada** de ambas imágenes.

Seguimiento de bordes por histéresis:



$u1(x, y)$

$u2(x, y)$



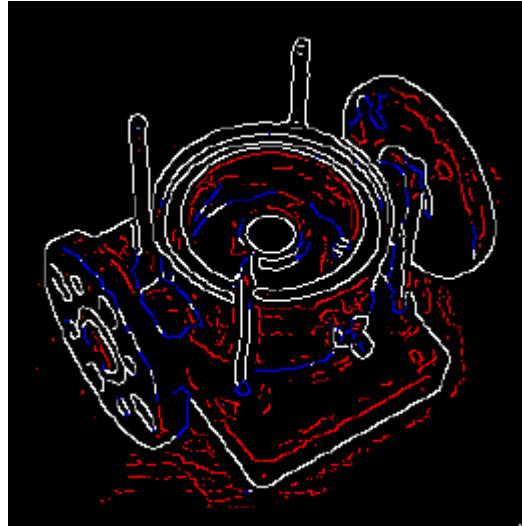
Bordes después de la
supresión de magnitudes
 $N(x, y)$



Doble umbralado
Bordes fuertes en
blanco, bordes débiles
en gris.



Doble umbralado



Seguimiento por
histéresis

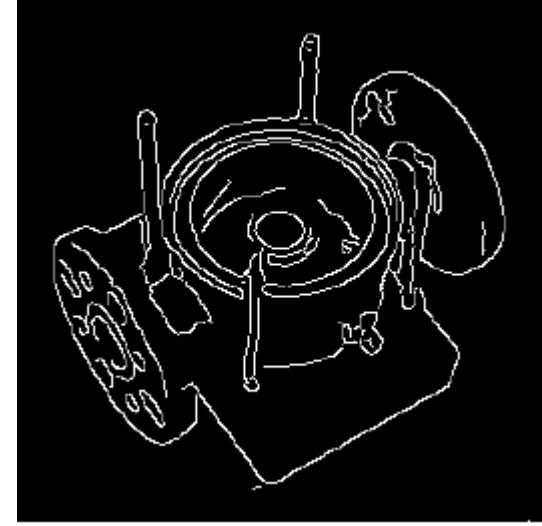
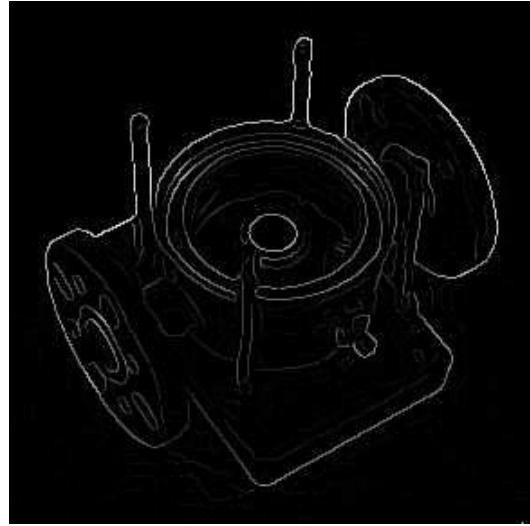
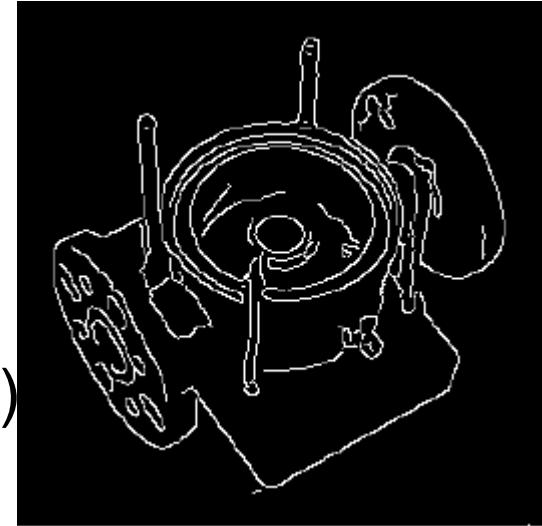


Imagen final



Canny (N, σ, ND, u_1, u_2)



Más resultados:



(a)

 $\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

(b)

 $\frac{1}{115}$

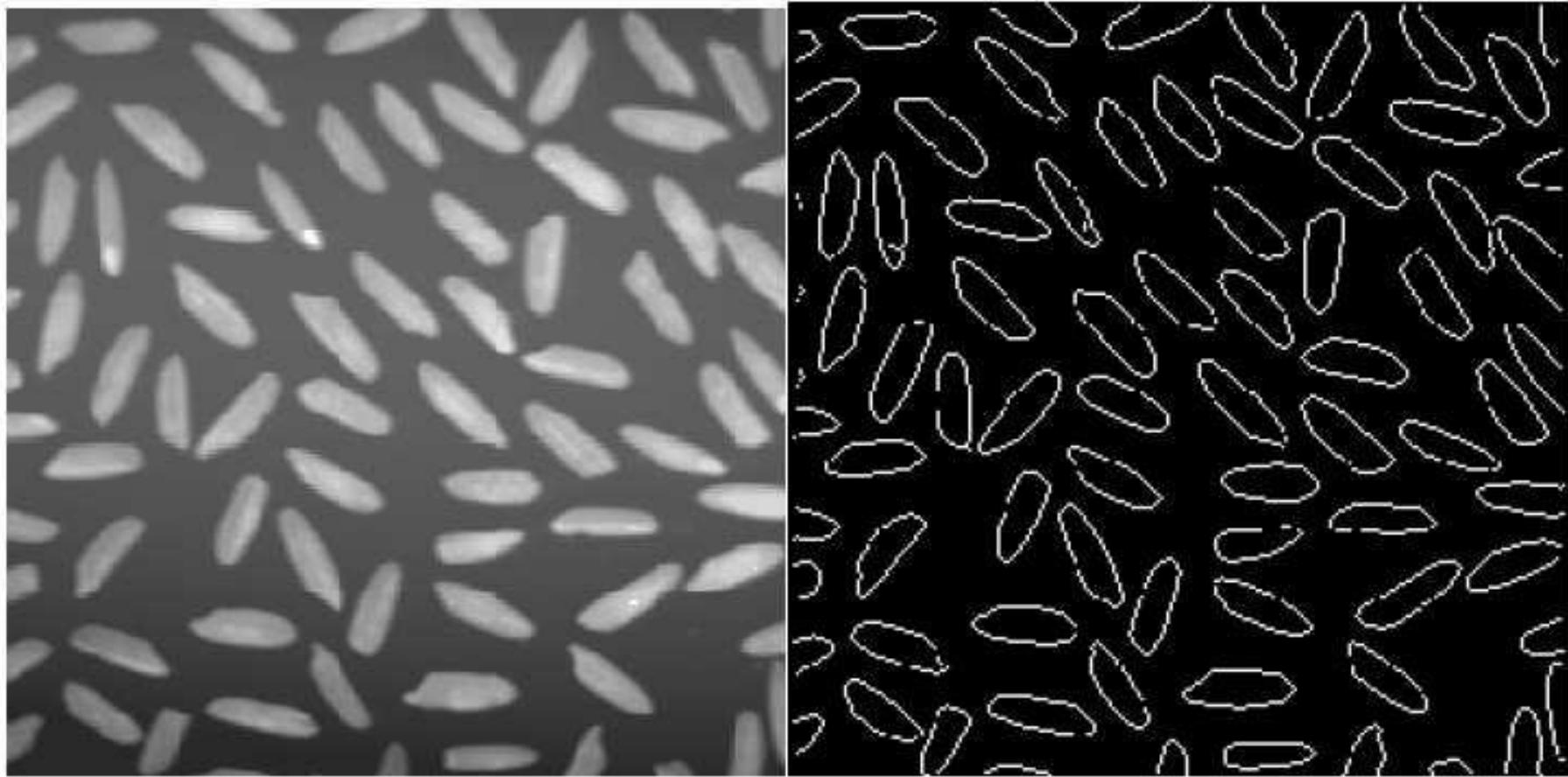
2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2



$u_1=100$, $u_2=130$,

$u_1=110$, $u_2=150$,

$u_1=120$, $u_2=180$.



```
% Programa para detectar mediante filtro Canny:  
  
clc % limpia pantalla.  
clearvars; % remueve todas las variables del espacio  
actual de trabajo.  
close all;% cierra grácas, archivos y recursos abiertos.  
  
f=imread('6 objetos.JPG'); % lee imagen de prueba f(x,y).  
figure(1)  
imshow(f) % muestra imagen de prueba.  
title('Imagen original')  
[n,m]=size(f); % dimensiones de la imagen f(x,y).  
%g=zeros(n,m);  
%g=f;  
%disp(n); % filas  
%disp(m); % columnas
```

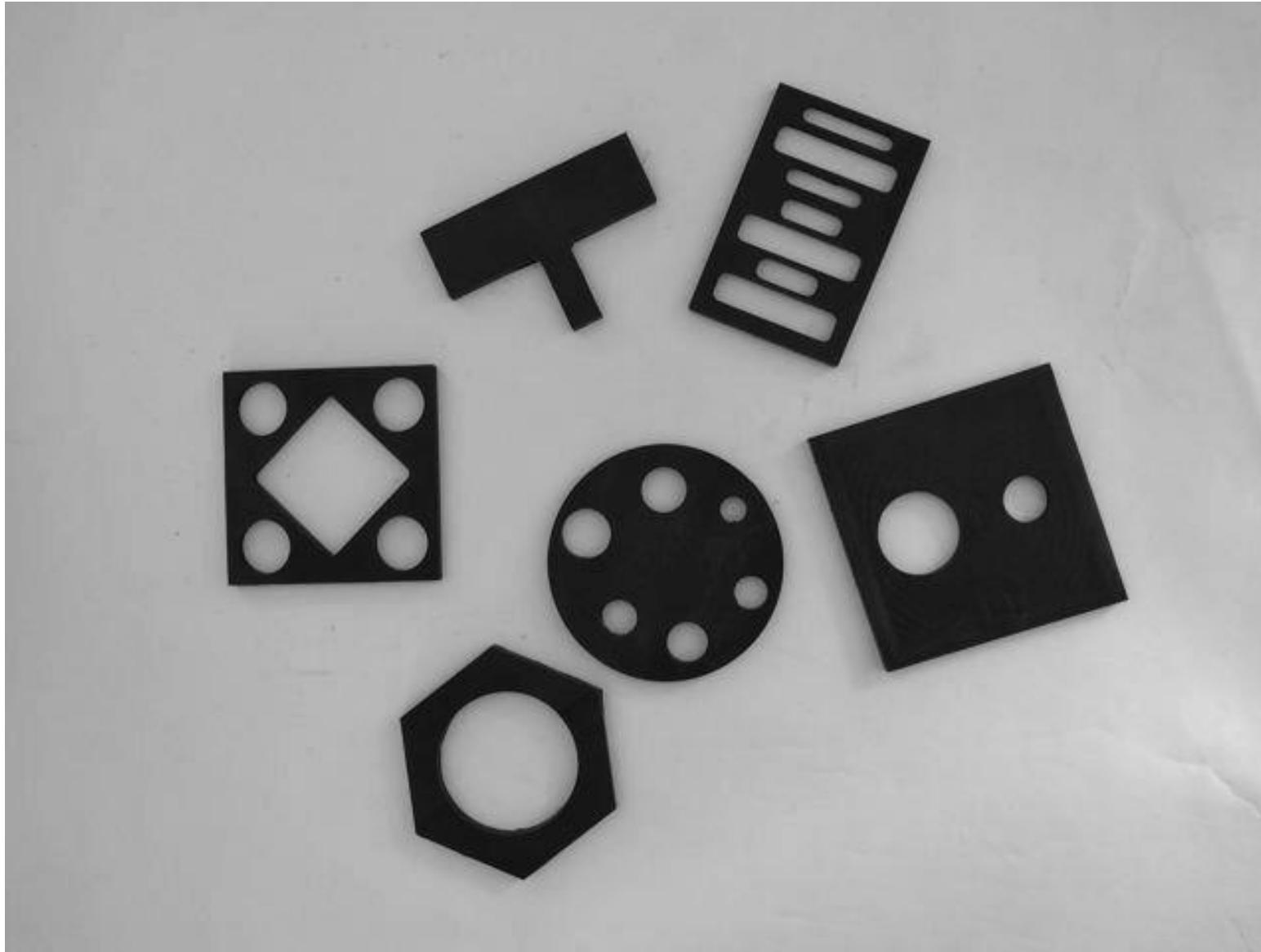
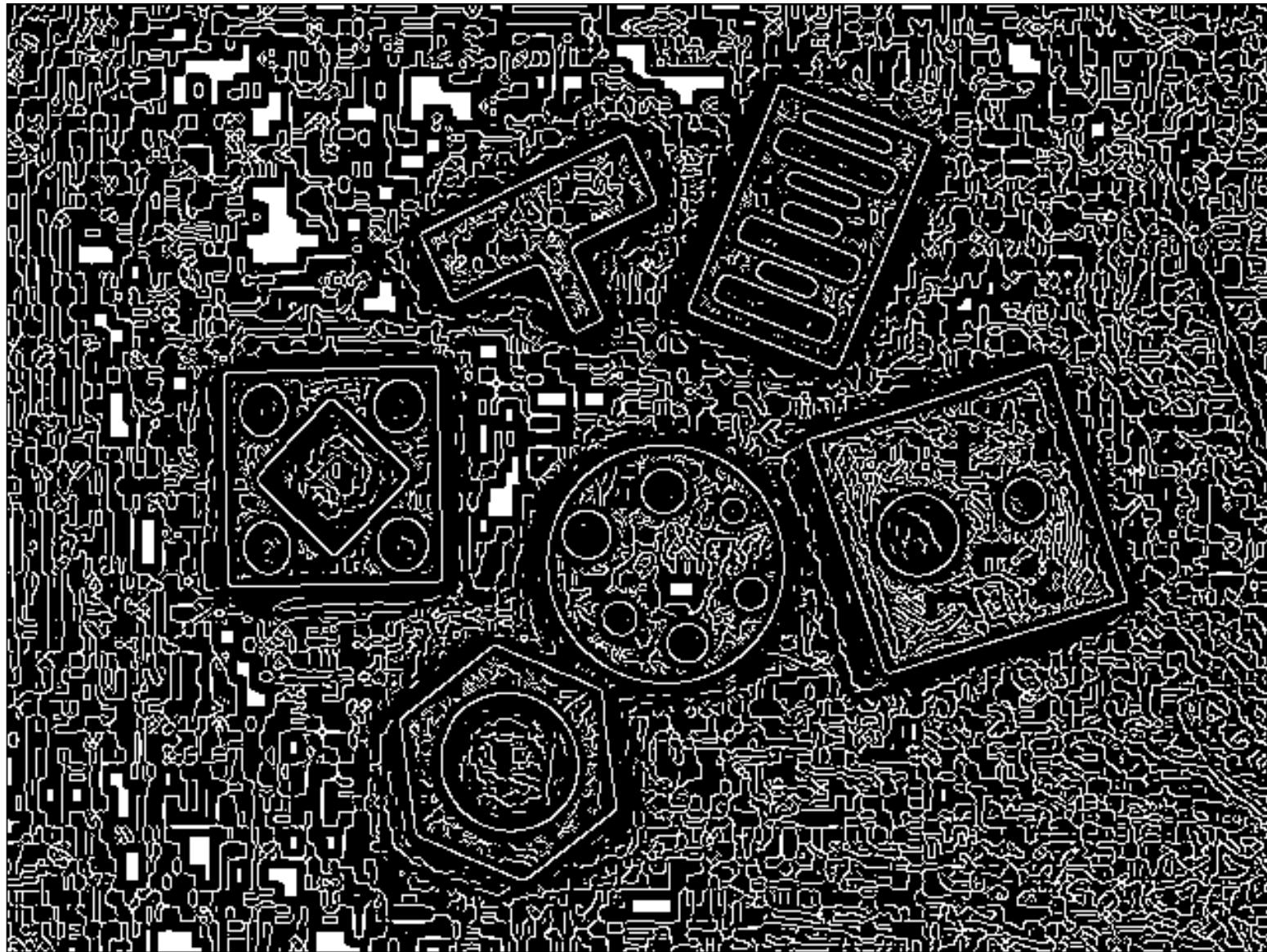
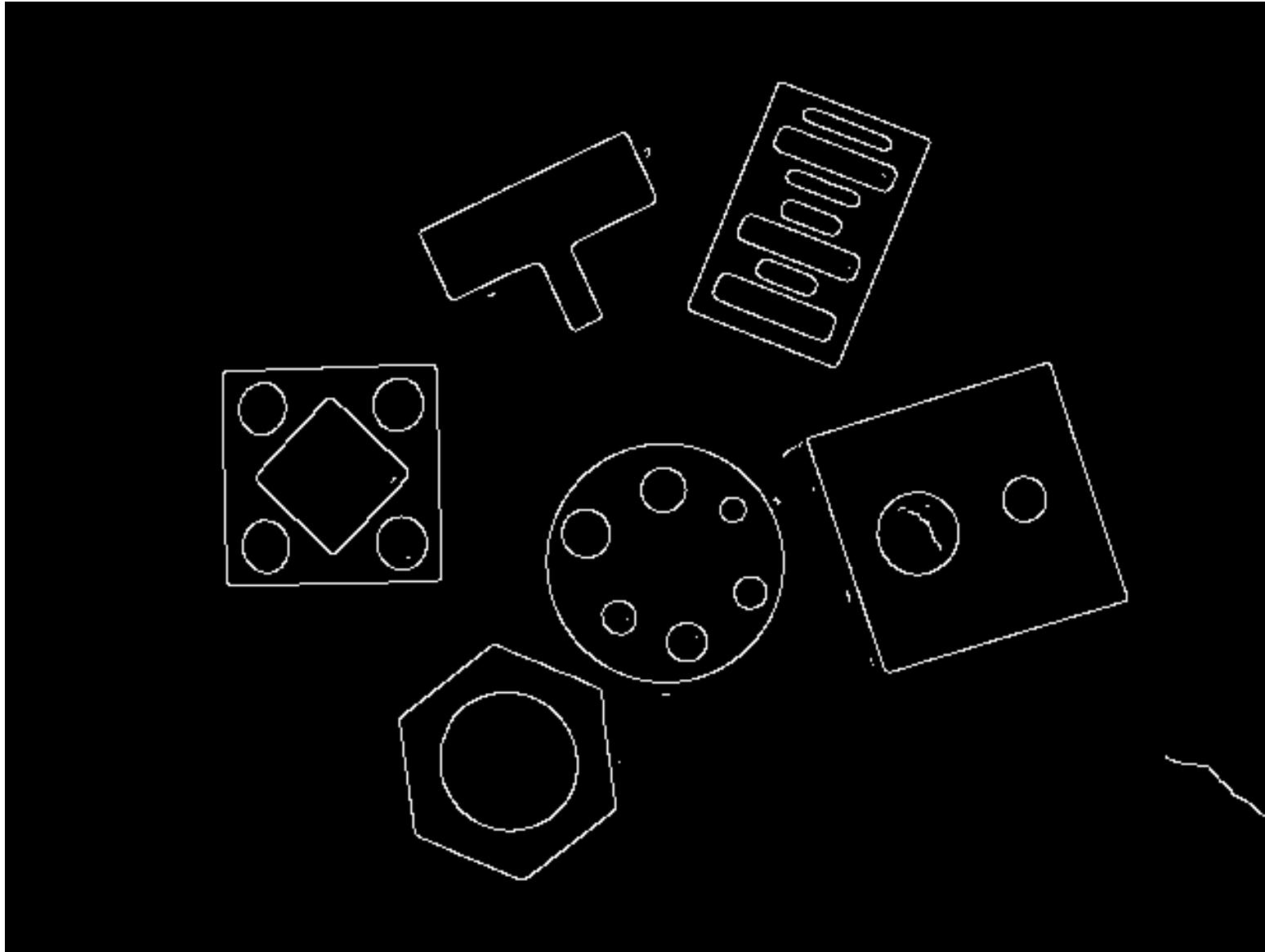


Imagen original

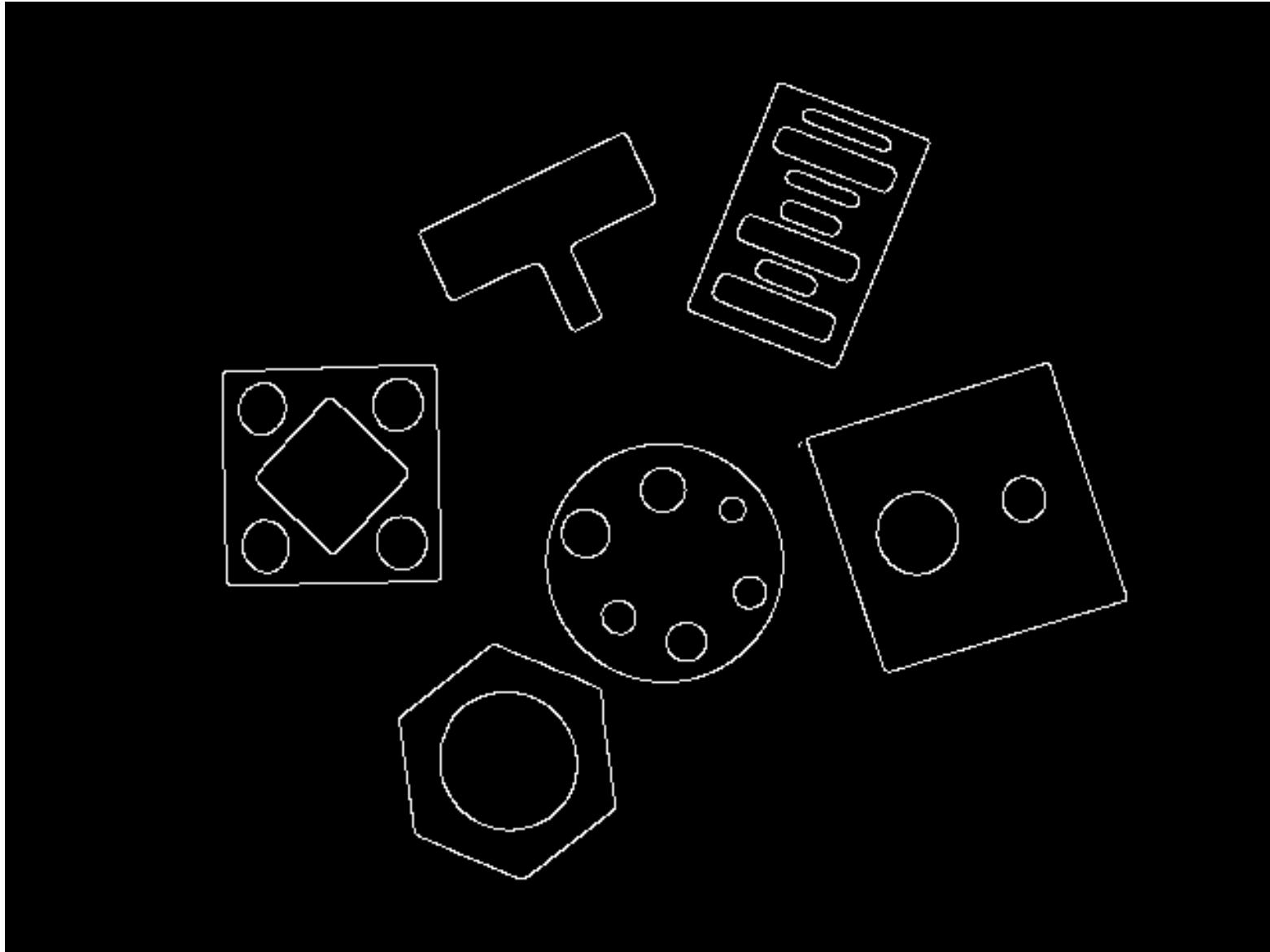
```
BW2 = edge(f, 'canny', 0.0, 1);  
figure(2)  
imshow(BW2)  
title('Filtro Canny');
```



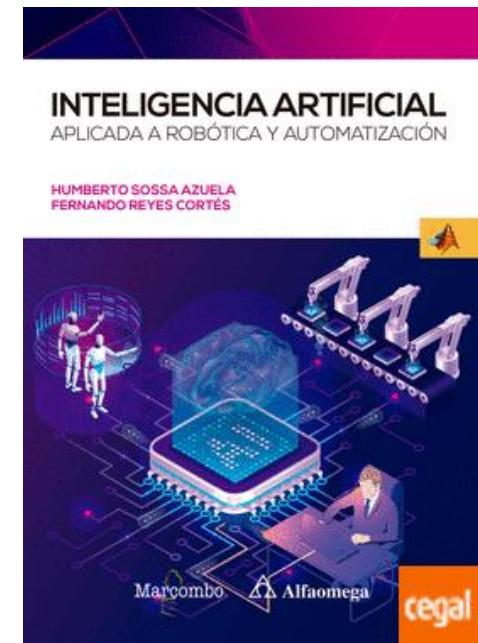
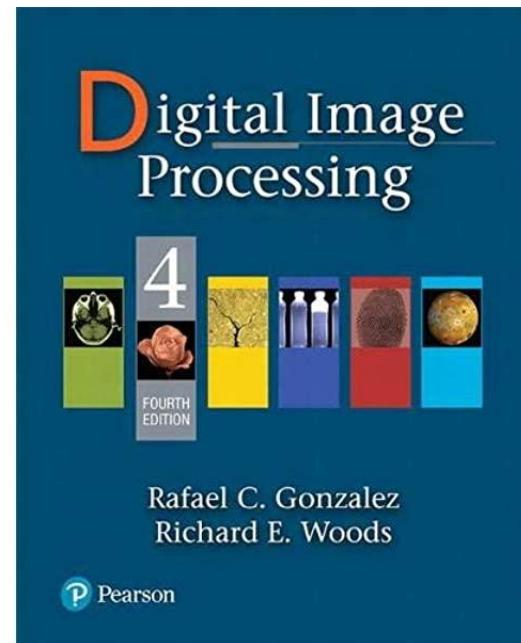
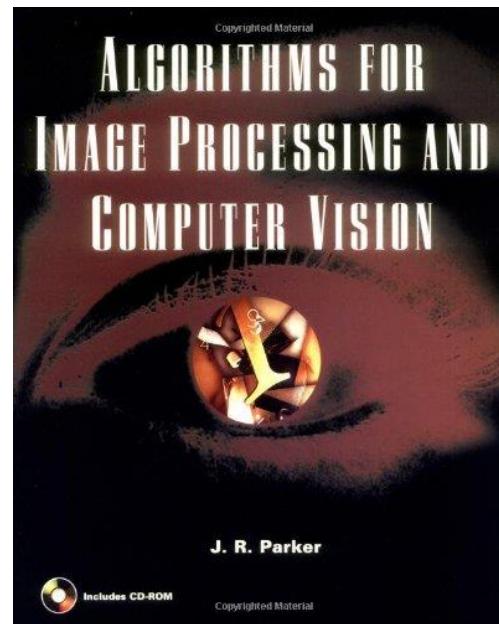
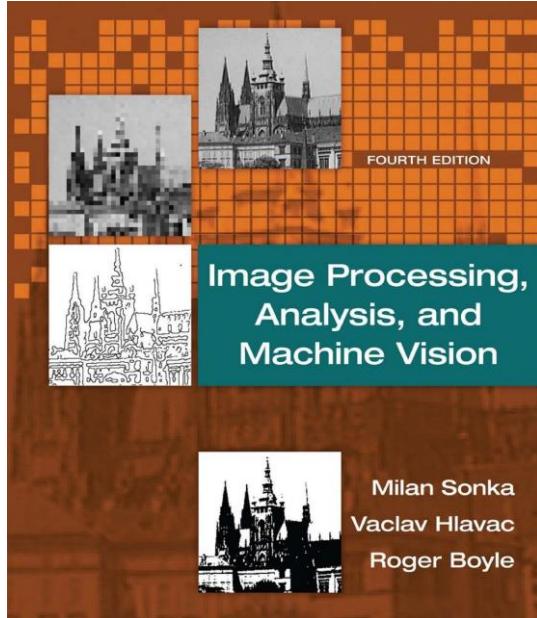
$$th = 0.0, \sigma = 1$$



$$th = 0.1, \sigma = 1$$



$$th = 0.2, \sigma = 1$$



UNIVERSIDAD
INTERNACIONAL
DE LA RIOJA

unir

www.unir.net