

# Filtros Espaciales y Morfológicos

**Juan Humberto Sossa Azuela**

**“GRABAR CLASE”**

**Favor de contestar a la siguiente encuesta:**

<https://forms.office.com/Pages/ResponsePage.aspx?id=pLTIIlibZskO8x4e5mFkLR0SdRk5MjR9FsliZPD56xZNUNIRVVzFEUUNVMDRJWjl2RzE1VDVHMjFTSi4u>

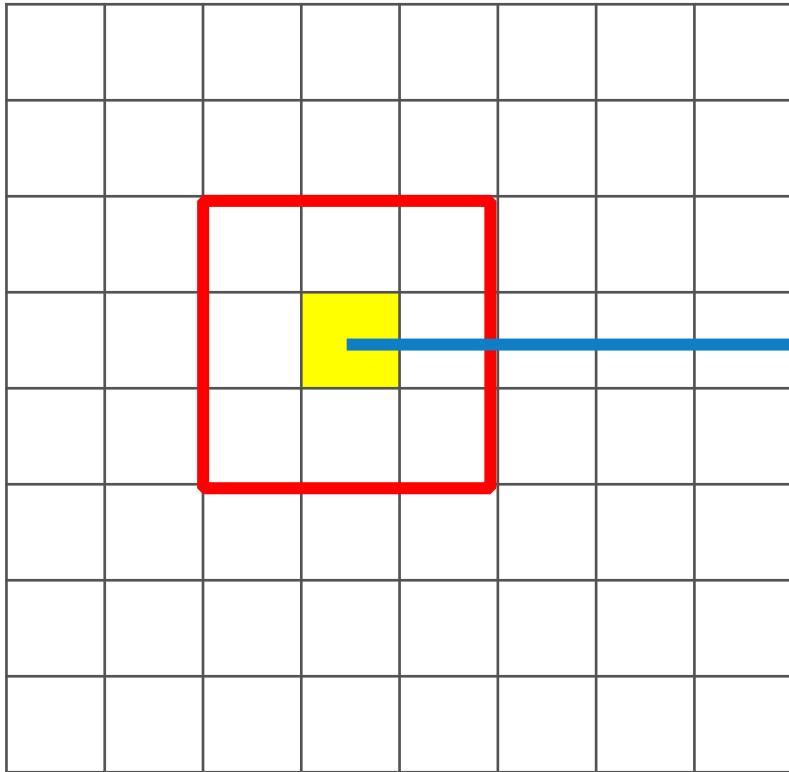
# **Objetivo:**

**Buscar y analizar los filtros espaciales y morfológicos más habitualmente usados en la literatura.**

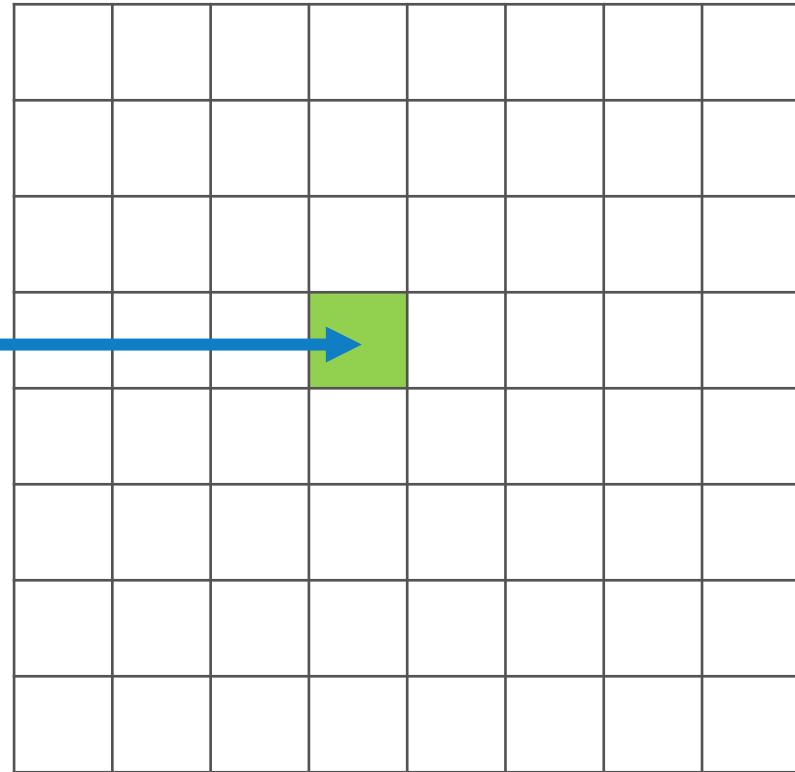
**Esto permitirá consolidar los conceptos y operaciones sobre imágenes aprendidas.**

# Descripción:

Las operaciones espaciales tienen en cuenta las relaciones de vecindad entre los píxeles para aplicar diferentes tipos de mejoras a la imagen.

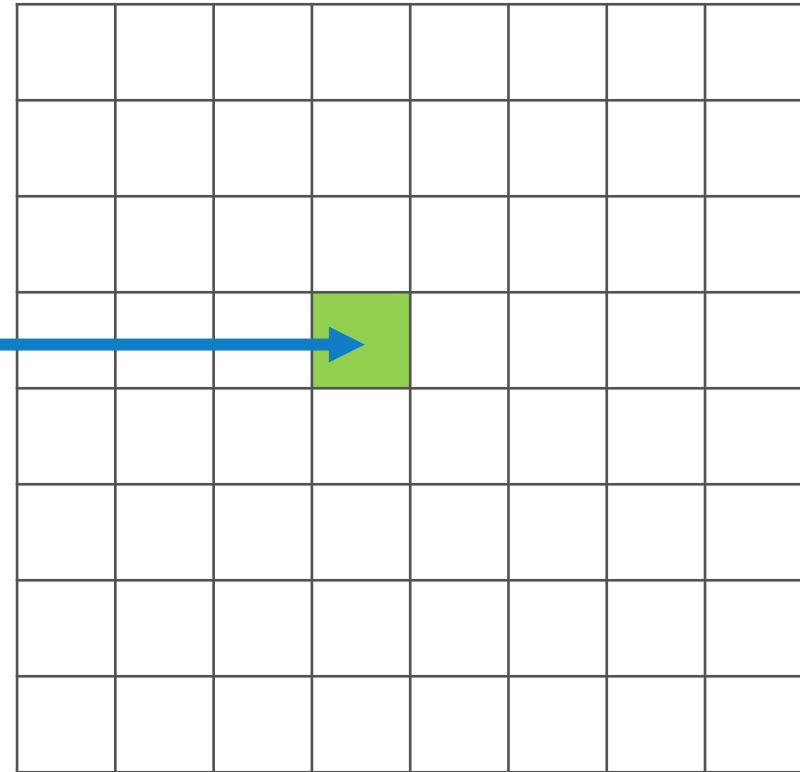
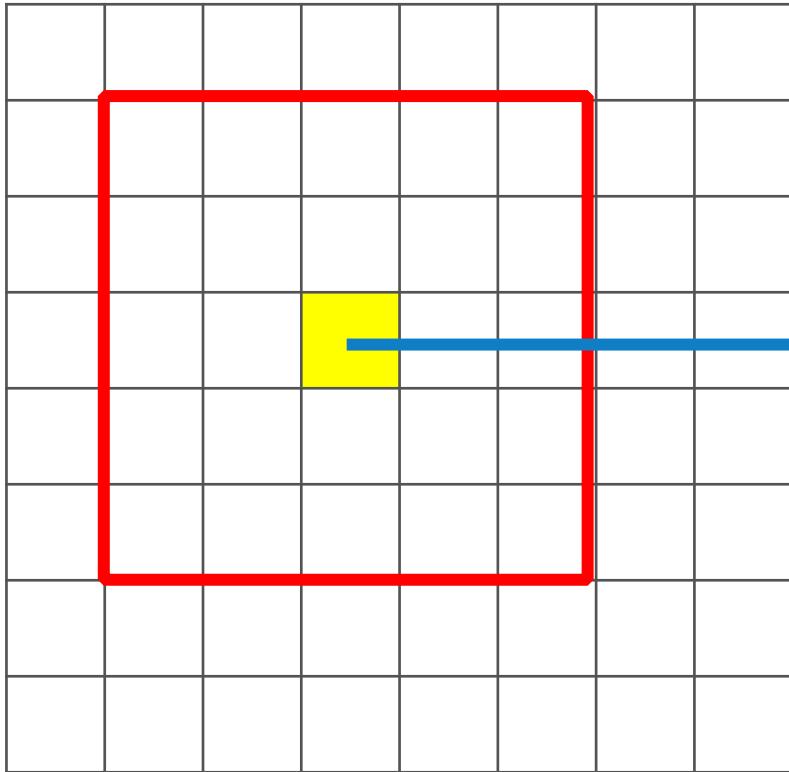


$T$



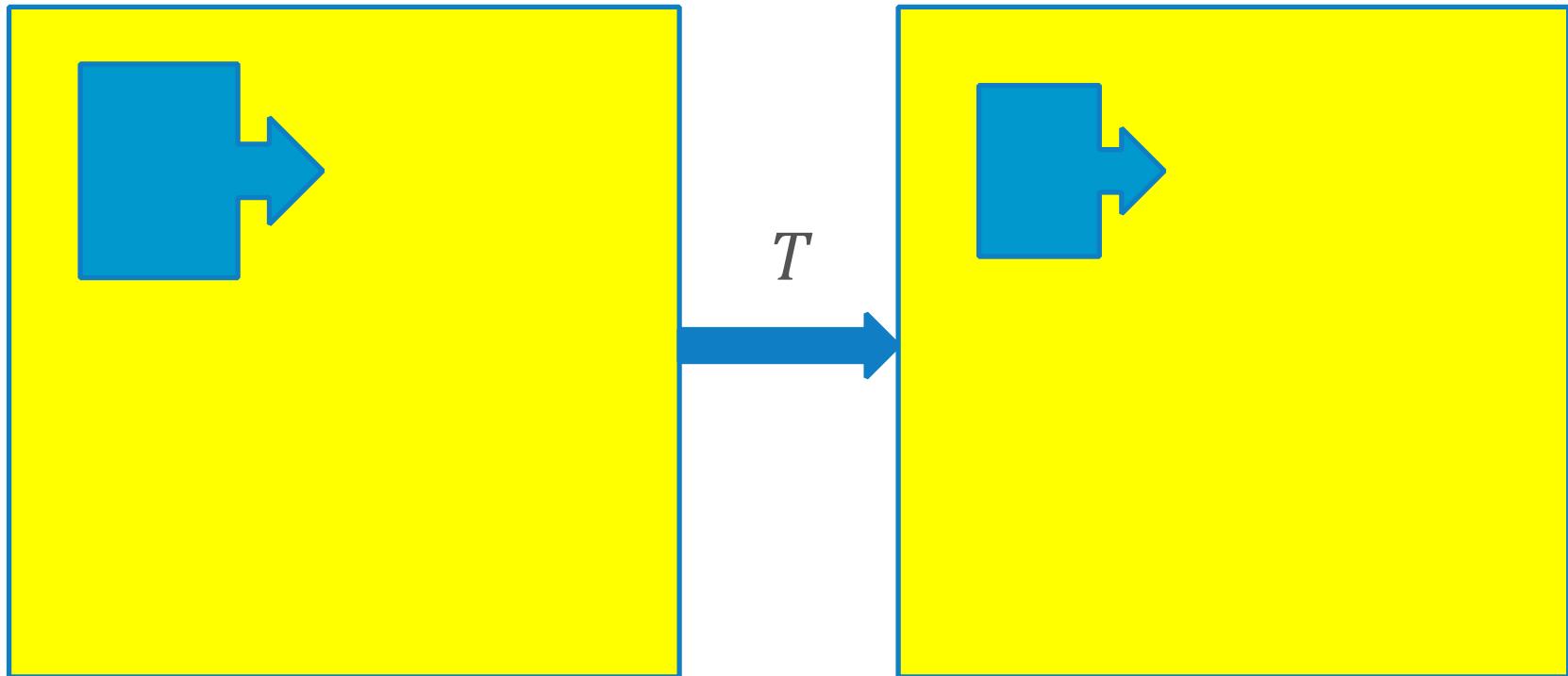
# Descripción:

Las operaciones espaciales tienen en cuenta las relaciones de vecindad entre los píxeles para aplicar diferentes tipos de mejoras a la imagen.



# Descripción:

Las operaciones morfológicas simplifican imágenes y conservan las principales características de los objetos.



# **Descripción:**

## **Los estudiantes:**

- 1) Deben analizar los filtros estudiados y compararlos con otros que se estén usando actualmente para resolución de problemas reales.**
  
- 2) Pueden buscar e investigar en paquetes de imagen de Python como Skimage, OpenCV, Mahotas, PIL, etc.**

## **Descripción:**

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

**En caso de que se utilice partes de un software existente, deberá referenciarse la fuente.**

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

## **Descripción:**

**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 entregarán dos documentos.**

- 1) Un notebook de Jupyter con la solución propuesta y los ficheros adicionales que se necesiten para ejecutarlo.**
- 2) Un informe donde se explique el problema donde se han implementado los diferentes filtros, cuáles de estos filtros o sus combinaciones y por qué, los resultados obtenidos y unas conclusiones. Se puede incluir bibliografía.**

# **Forma de entrega:**

**Extensión máxima del informe: 6/8 páginas.**

**NOTA: El alumno puede decidir que imágenes usar.**

**Se recomienda que buscar imágenes de su ámbito laboral (médicas, industriales, etcétera) o de su interés personal.**

**También puede se pueden usar imágenes tomadas con cámara o celular.**

# Rúbrica:

Criterio	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	<b>El problema descripto y la solución aportada tienen una utilidad práctica real.</b>	2	20%
Criterio 2	<b>Los objetivos, ventajas e inconvenientes están claramente descritos.</b>	1	10%
Criterio 3	<b>El código aportado muestra los principales pasos, y ejecuta sin errores.</b>	3	30%
Criterio 4	<b>Existe una implementación propia que dirige y evalúa los resultados.</b>	2	20%
Criterio 5	<b>Se describe el proceso de forma sencilla. La redacción es clara, comprensible y sin rodeos o redundancias.</b>	2	20%
Criterio 6	<b>Existe plagio no debidamente referenciado o el trabajo es similar y con los mismos punto de fallo que el de otro grupo.</b>	-10	-100%
		10	100 %

# **Operaciones aritméticas entre imágenes:**

$$p_1 + p_2$$

$$p_1 - p_2$$

$$p_1 \cdot p_2$$

$$p_1 \div p_2$$

8				

4				

12				

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm>

$p_1 + p_2$

$p_1 - p_2$

$p_1 \cdot p_2$

$p_1 \div p_2$

8				

4				

4				

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm>

$p_1 + p_2$

$p_1 - p_2$

$p_1 \cdot p_2$

$p_1 \div p_2$

8				

4				

32				

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm>

$p_1 + p_2$

$p_1 - p_2$

$p_1 \cdot p_2$

$p_1 \div p_2$

8				

4				

2				

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm>



(a)



(b)



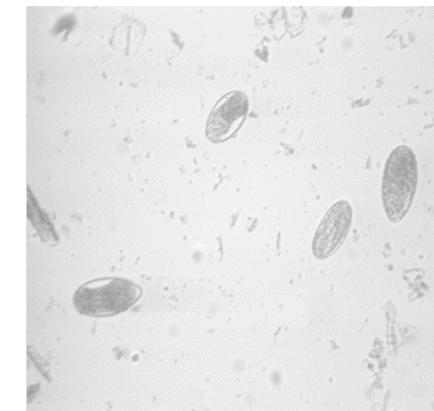
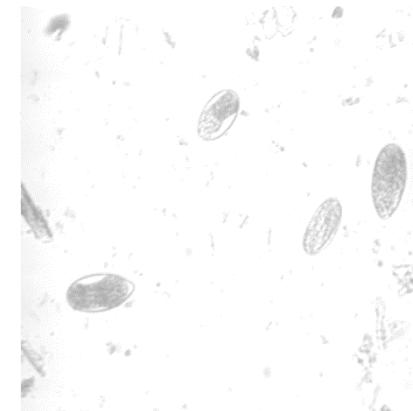
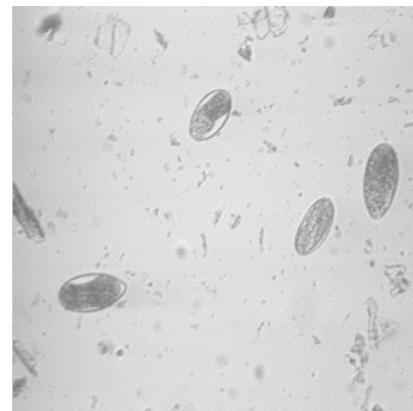
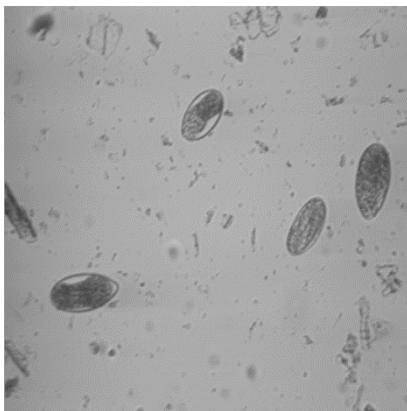
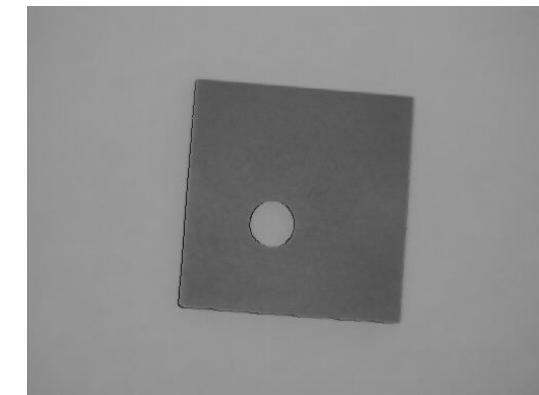
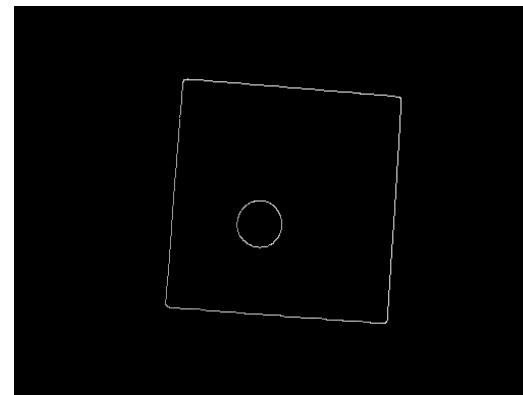
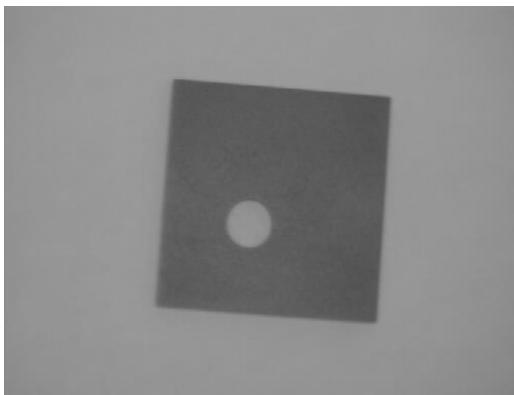
(c)



(d)

Efecto de aplicar las operaciones aritméticas de suma y resta entre imágenes. (a) Imagen número 1. (b) Imagen número 2. (c) Suma de (a) y (b). (d) Resta de (b) y (a)

# Aplicaciones de la suma aritmética de imágenes:



+50

saturación

reescalado

% Programa para la suma y resta de imágenes.

```
clc; % limpia pantalla  
clearvars; % remueve todas las variables.  
close all; % cierra gráficas, archivos y recursos abiertos  
  
f1=imread('3 objetos.JPG'); % se lee primera imagen  
f2=imread('6 objetos.JPG'); % se lee segunda imagen  
  
figure(1) % se abre una ventana auxiliar.  
imshow(f1) % se muestra imagen f1.  
title('Imagen 1') % se asigna título de esta imagen.  
  
[n,m]=size(f1); % dimensiones de la imagen f1  
  
figure(2) % se abre una ventana auxiliar.  
imshow(f2) % se muestra imagen f2.  
title('Imagen 2') % se asigna título de esta imagen.
```

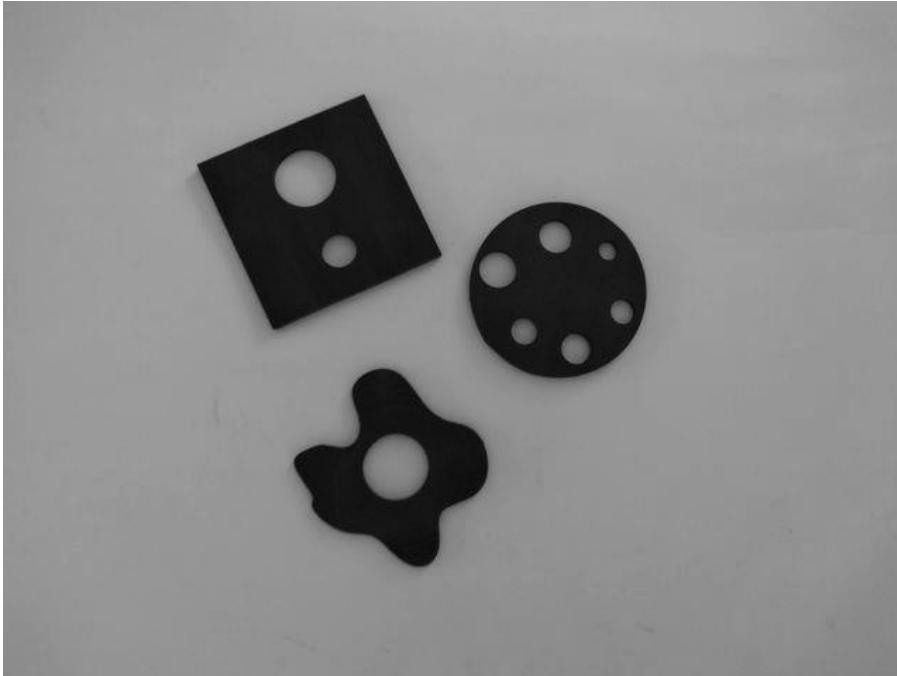


Imagen 1

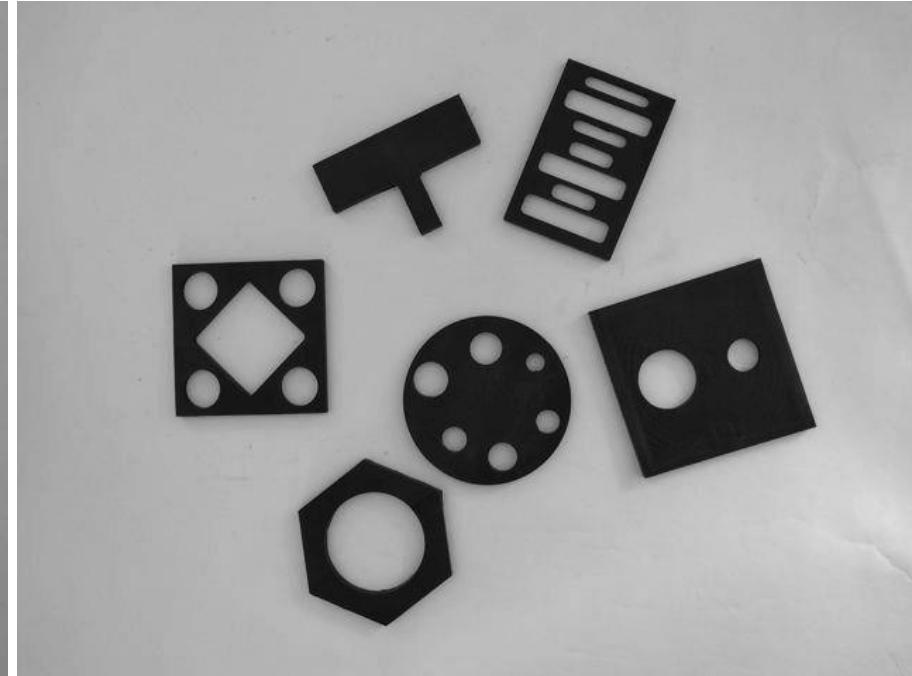


Imagen 2

```
for x=1:n % se barre imagen en renglones.  
    for y=1:m % se barre imagen en columnas.  
        i=f1(x,y); % se obtiene nivel de gris de imagen f1.  
        j=f2(x,y); % se obtiene nivel de gris de imagen f2.  
  
        k=i+j; % suma de píxeles  
        if k > 255 % si el valor sobrepasa se satura en 255.  
            k=255  
        end  
        g1(x,y)=k;  
  
        h=i-j; % resta de píxeles  
        if h < 0  
            h=0 % si el valor sobrepasa se satura en 0.  
        end  
        g2(x,y)=h;  
    end  
end
```

```
figure (3) % se abre una ventana auxiliar.  
imshow(g1) % se muestra imagen suma f1+f2.  
title('Suma de imágenes'); % se asigna título de esta imagen.
```

```
figure (4) % se abre una ventana auxiliar.  
imshow(g2) % se muestra imagen suma f1-f2.  
title('Resta de imágenes'); % se asigna título de esta imagen.
```

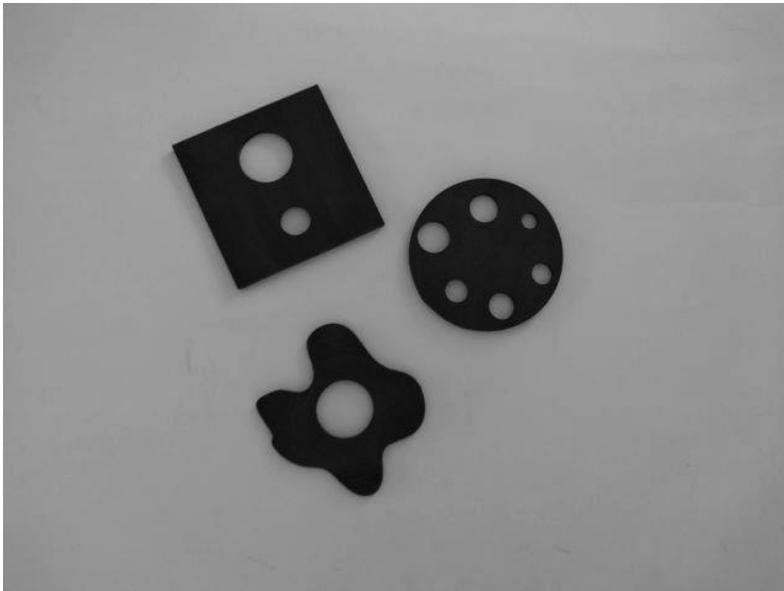


Imagen 1

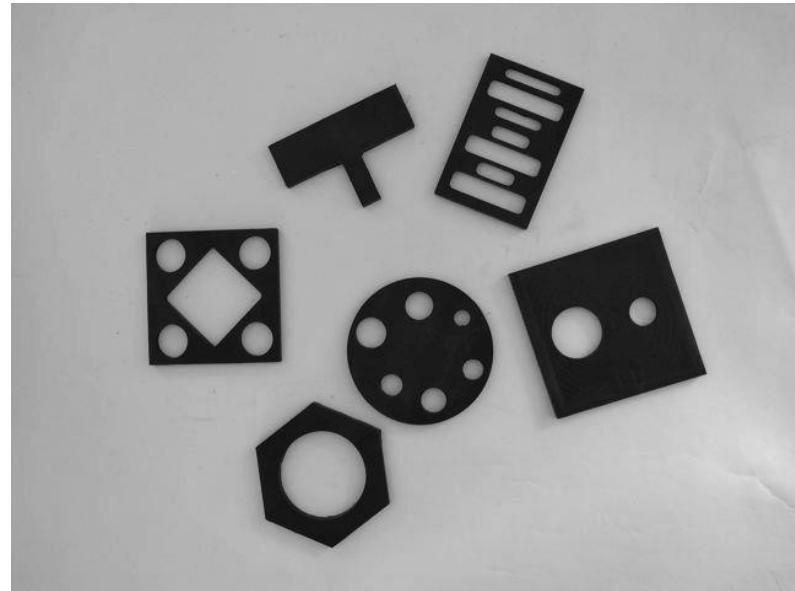
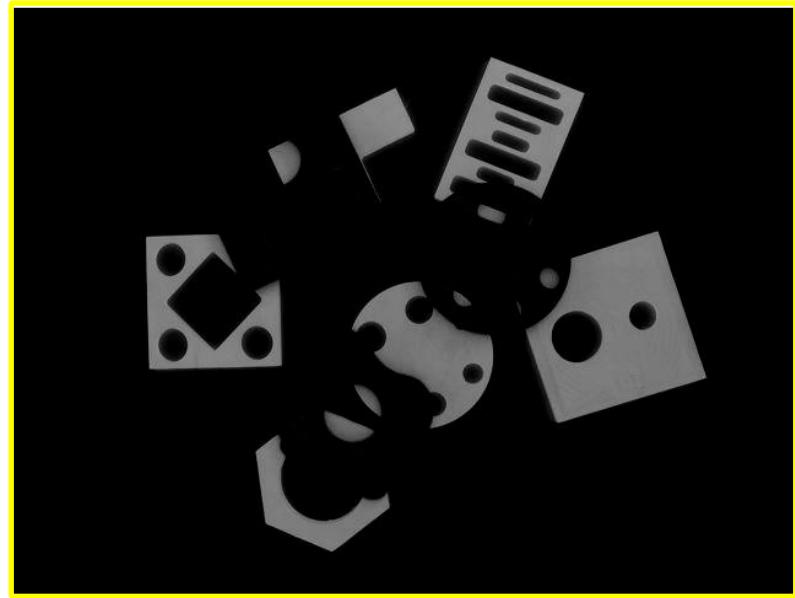


Imagen 2

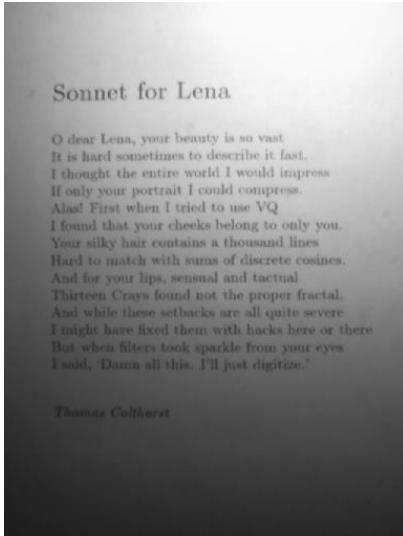


suma

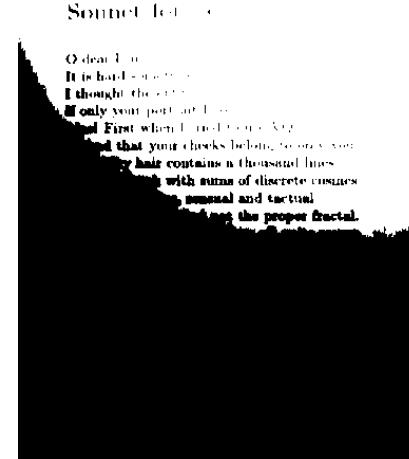


resta

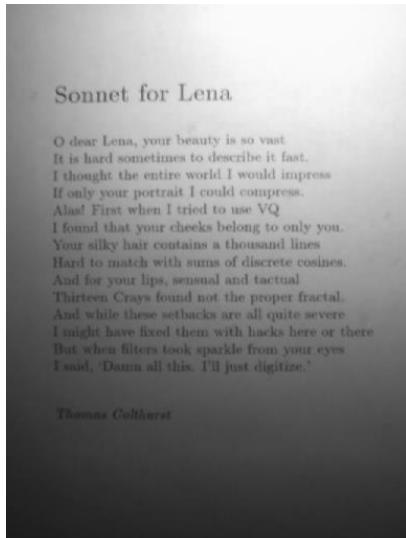
# Aplicaciones de la resta aritmética de imágenes:



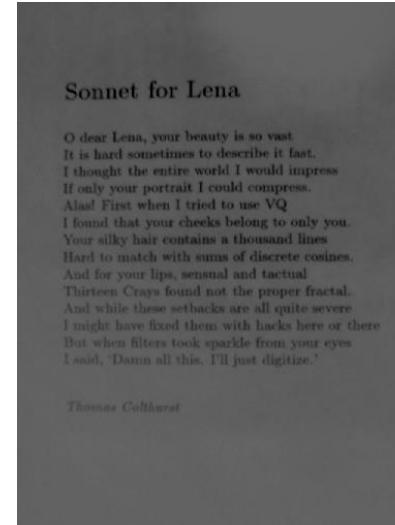
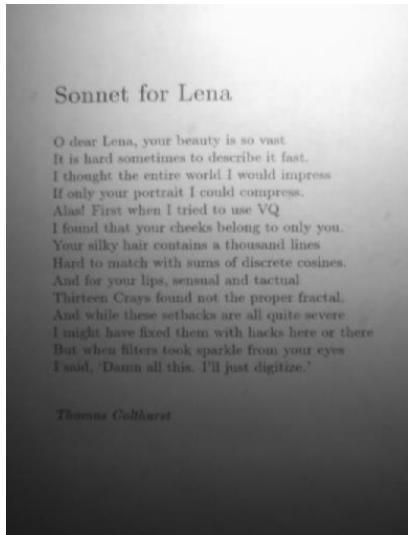
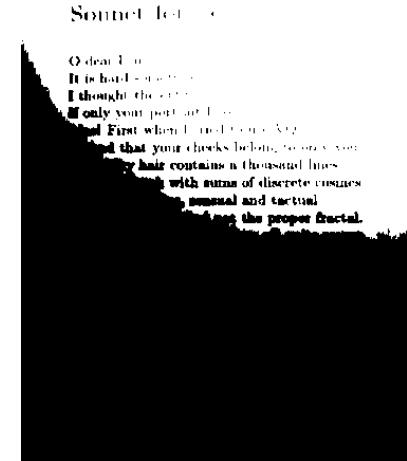
umbralado



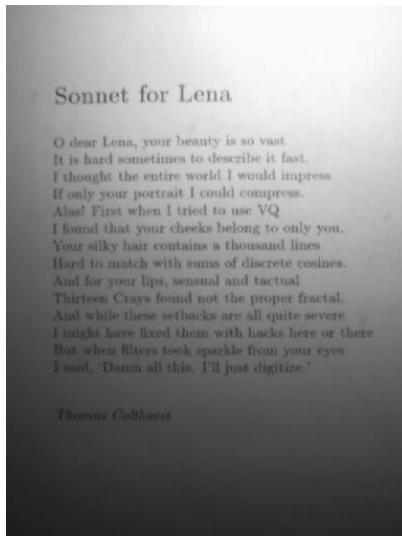
# Aplicaciones de la resta aritmética de imágenes:



umbralado



# Aplicaciones de la resta aritmética de imágenes:



Sonnet for Lena

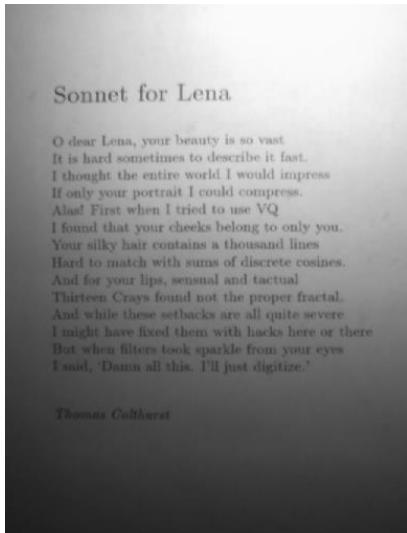
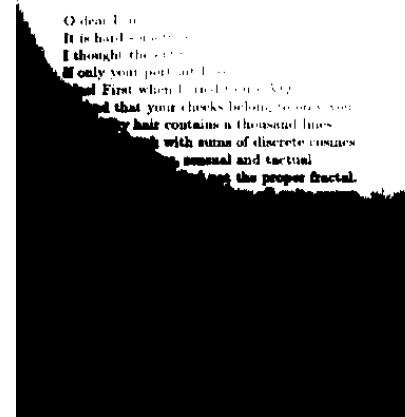
O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactful  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

## umbralado

Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactful  
Thirteen Crays found not the proper fractal.



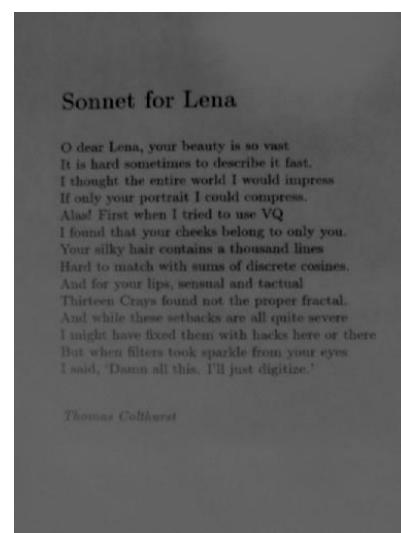
Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactful  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



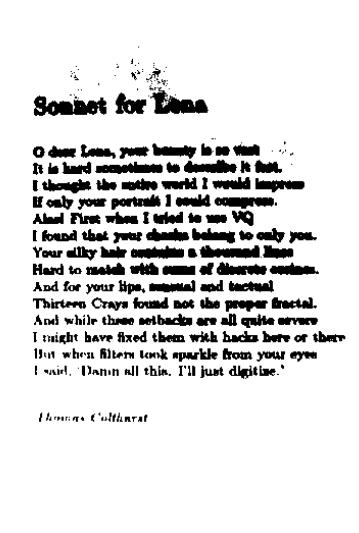
=



Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactful  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



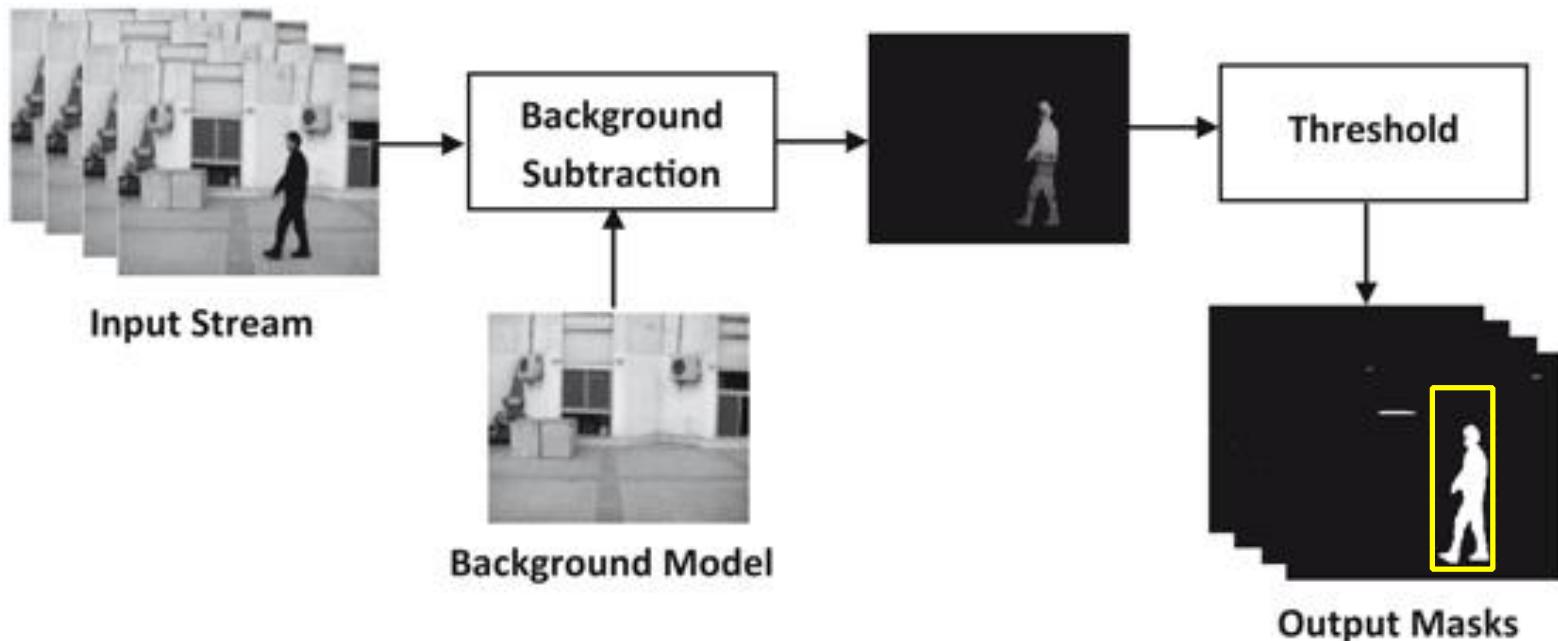
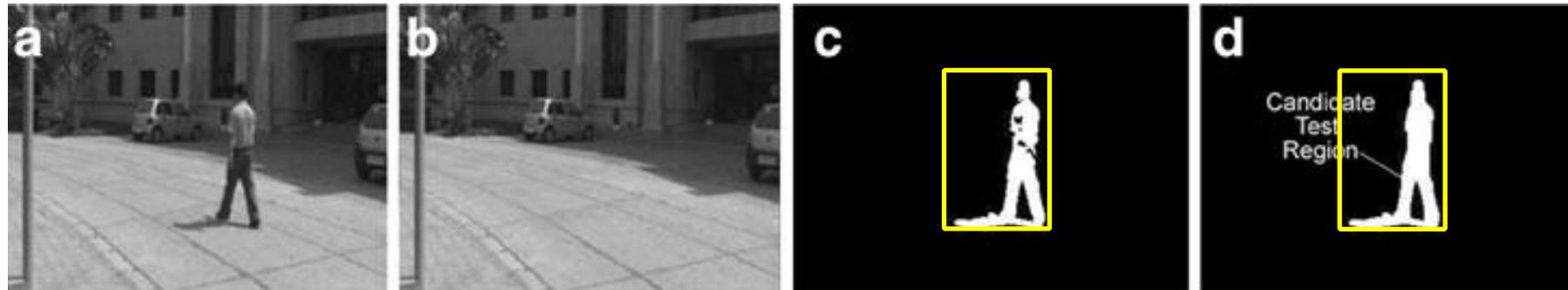
Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactful  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

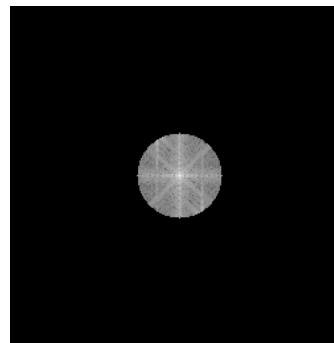
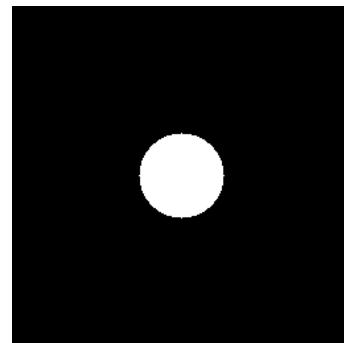
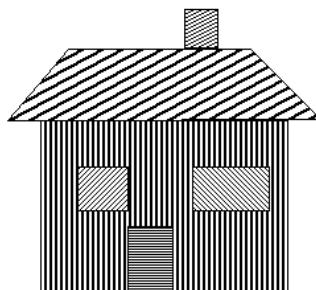
Thomas Colthurst

# Otras aplicaciones de la resta de imágenes:

Detección de movimiento.



# Aplicaciones del producto aritmético de imágenes:



# Aplicaciones de la división aritmética de imágenes:



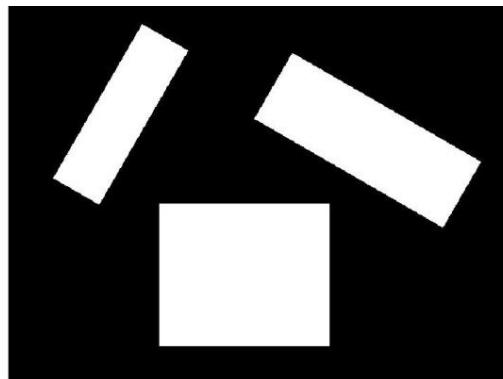
Al dividir las imágenes y al aplicar un ensanchado de contraste:



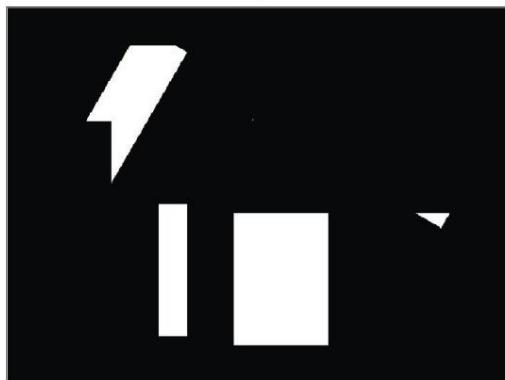
# Operaciones binarias entre imágenes:



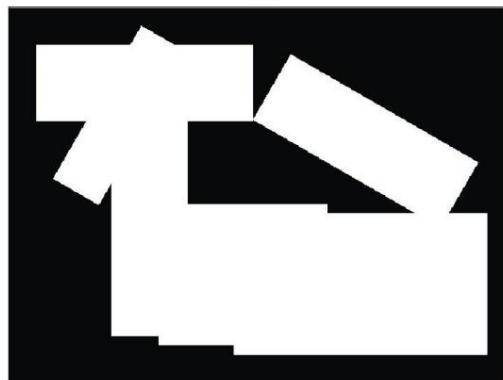
(a)



(b)



(c)



(d)



(e)



(f)

Operaciones lógicas entre imágenes binarias.

- (a) Imagen A.
- (b) Imagen B.
- (c) A Y B.
- (d) A O B.
- (e) NO (A).
- (f) A EXO B.

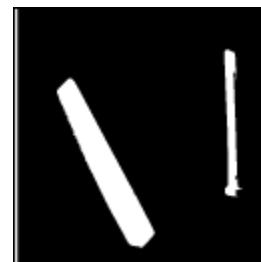
Y lógico:



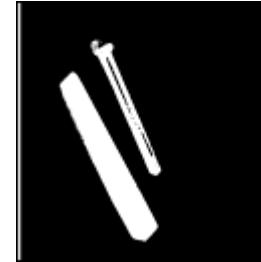
I1



I2



b1



b2



$\sim b1 \wedge \sim b2$

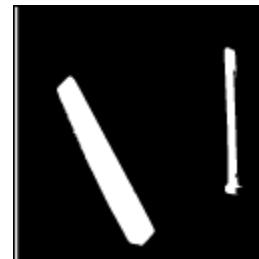
Y lógico:



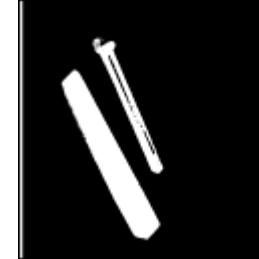
I1



I2



b1



b2



$\sim b1 \text{ Y } \sim b2$

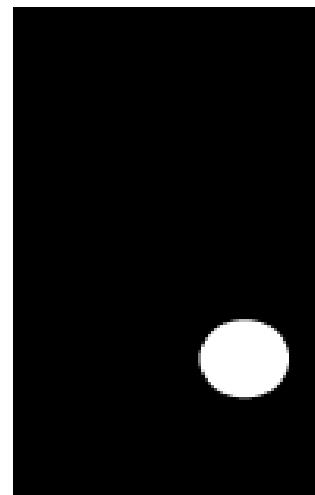
Enmascaramiento (resaltar en brillo una región de la imagen:)



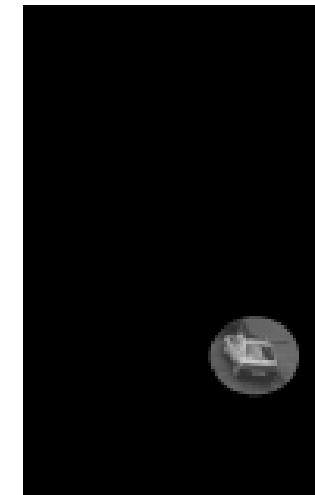
Imagen I



Región sel.



Máscara M

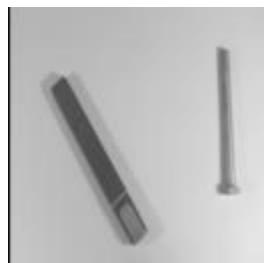


$I \text{ Y } M=N$



$I+N$  con brillo

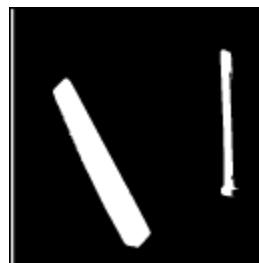
O lógico:



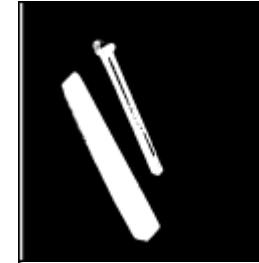
I1



I2



b1

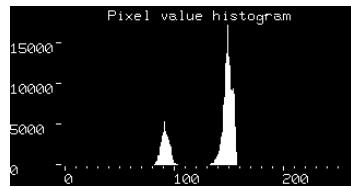
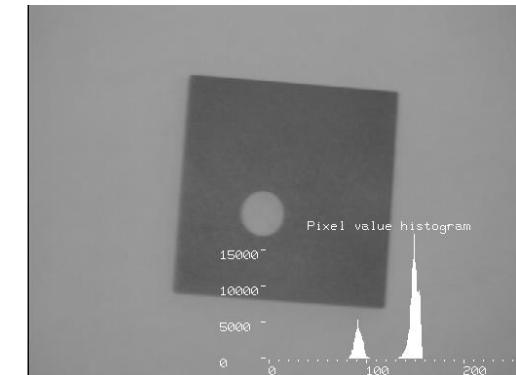
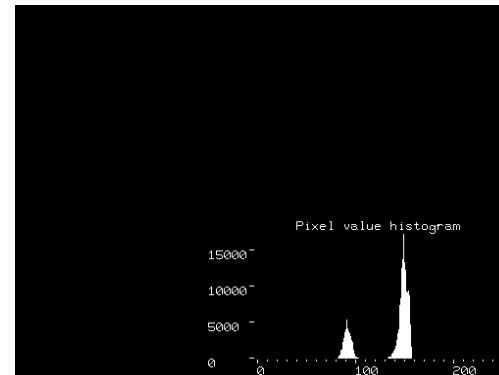


b2

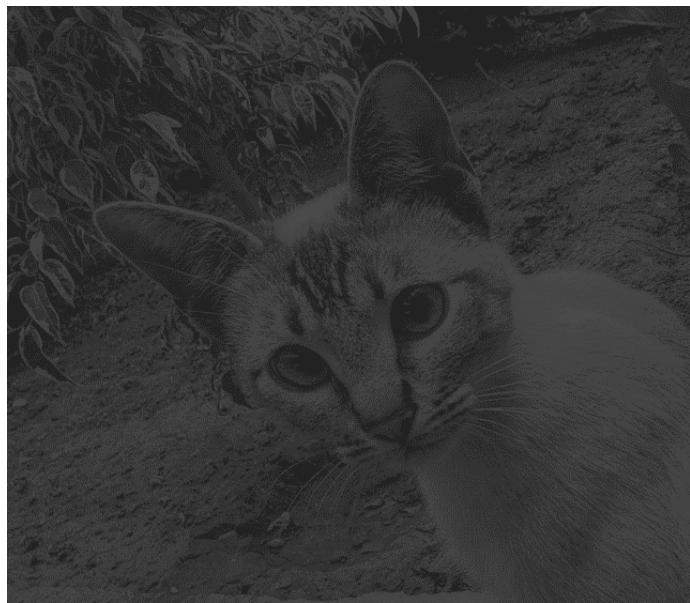
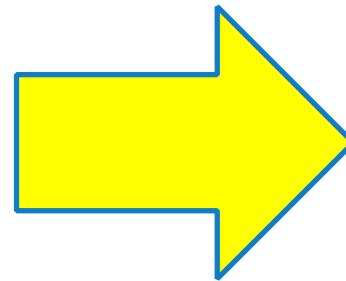


b1 OY b2

Sobre posición de imágenes:



# Operaciones de contrastado:



**Imagen con poco contraste**



**Imagen contrastada**

**Ensanchamiento de contraste:**  $P_{out} = (P_{in} - c) \left( \frac{b-a}{d-c} \right) + a$

c = nivel de gris más bajo actual.

d = nivel de gris más alto actual.

a = nivel de gris más bajo = 0.

b = nivel de gris más alto = 255.

$$P_{out} = (P_{in} - 79) \left( \frac{255}{57} \right)$$

Ensanchamiento de contraste:  $P_{out} = (P_{in} - c) \left( \frac{b-a}{d-c} \right) + a$

c = nivel de gris más bajo actual.

d = nivel de gris más alto actual.

a = nivel de gris más bajo = 0.

b = nivel de gris más alto = 255.

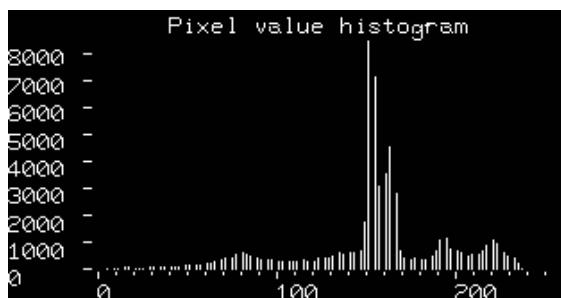
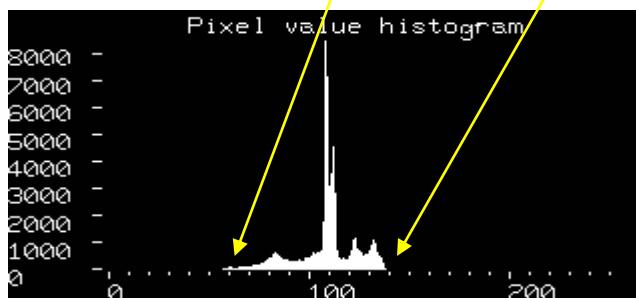
Imagen con  
c=79, d=136.

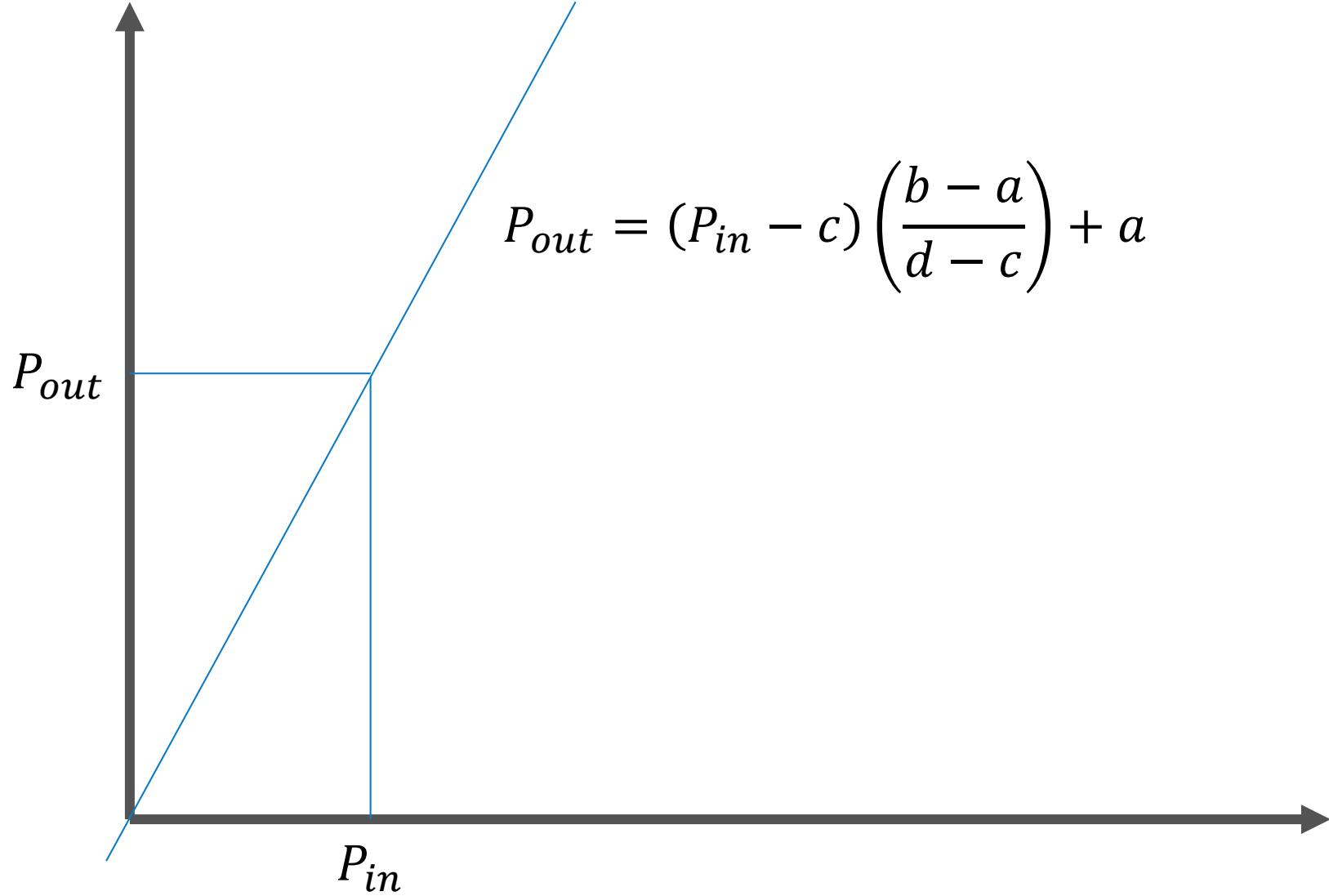
Imagen con  
contraste  
ensanchado.

Imagen con  
c=79, d=136.



Imagen con  
contraste  
ensanchado.





3	3	3	3	3	3	3	3
3	4	4	4	4	4	4	3
3	4	5	5	5	5	4	3
3	4	5	5	5	5	4	3
3	4	5	5	5	5	4	3
3	4	5	5	5	5	4	3
3	4	4	4	4	4	4	3
3	3	3	3	3	3	3	3

## Ensanchamiento de contraste:

c = nivel de gris más bajo actual=3

d = nivel de gris más alto actual=5

a = nivel de gris más bajo = 0.

b = nivel de gris más alto = 7.

$$P_{out} = (P_{in} - c) \frac{b - a}{d - c} + a, P_{out} = (3 - 3) \frac{7 - 0}{5 - 3} + 0 = 0$$

$$P_{out} = (P_{in} - c) \frac{b - a}{d - c} + a, P_{out} = (4 - 3) \frac{7 - 0}{5 - 3} + 0 = 3.5$$

$$P_{out} = (P_{in} - c) \frac{b - a}{d - c} + a, P_{out} = (5 - 3) \frac{7 - 0}{5 - 3} + 0 = 7.0$$

## Imagen original

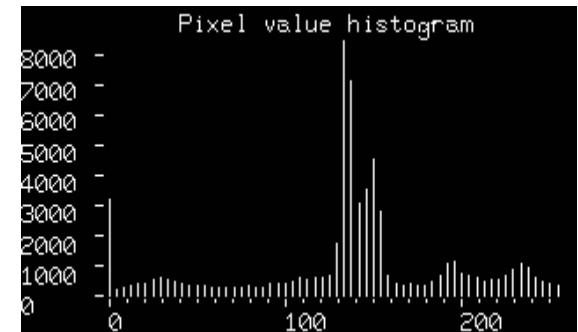
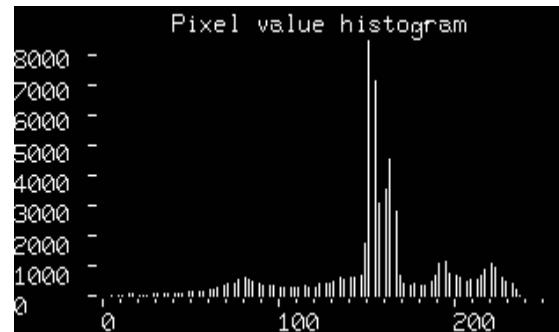
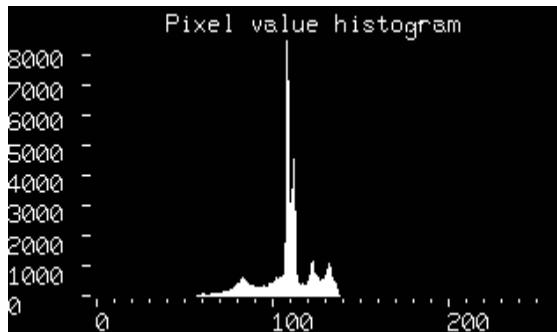
3	3	3	3	3	3	3	3	3
3	4	4	4	4	4	4	4	3
3	4	5	5	5	5	4	3	
3	4	5	5	5	5	4	3	
3	4	5	5	5	5	4	3	
3	4	5	5	5	5	4	3	
3	4	5	5	5	5	4	3	
3	4	4	4	4	4	4	3	
3	3	3	3	3	3	3	3	



## Imagen contrastada

0	0	0	0	0	0	0	0	0
0	3	3	3	3	3	3	3	0
0	3	7	7	7	7	3	0	
0	3	7	7	7	7	3	0	
0	3	7	7	7	7	3	0	
0	3	7	7	7	7	3	0	
0	3	7	7	7	7	3	0	
0	3	3	3	3	3	3	0	
0	0	0	0	0	0	0	0	0

Se pueden obtener mejores resultados al dejar de considerar algunos píxeles, normalmente los píxeles cuyos niveles de gris se encuentran a los **extremos** del histograma (**recorte**):



```
% Programa para cambiar contraste de imagen f(x,y)

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

f=imread('3 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)
c=zeros(256,1); % inicializa en ceros contador c(i).
niveles=0:255; % niveles de gris.
```

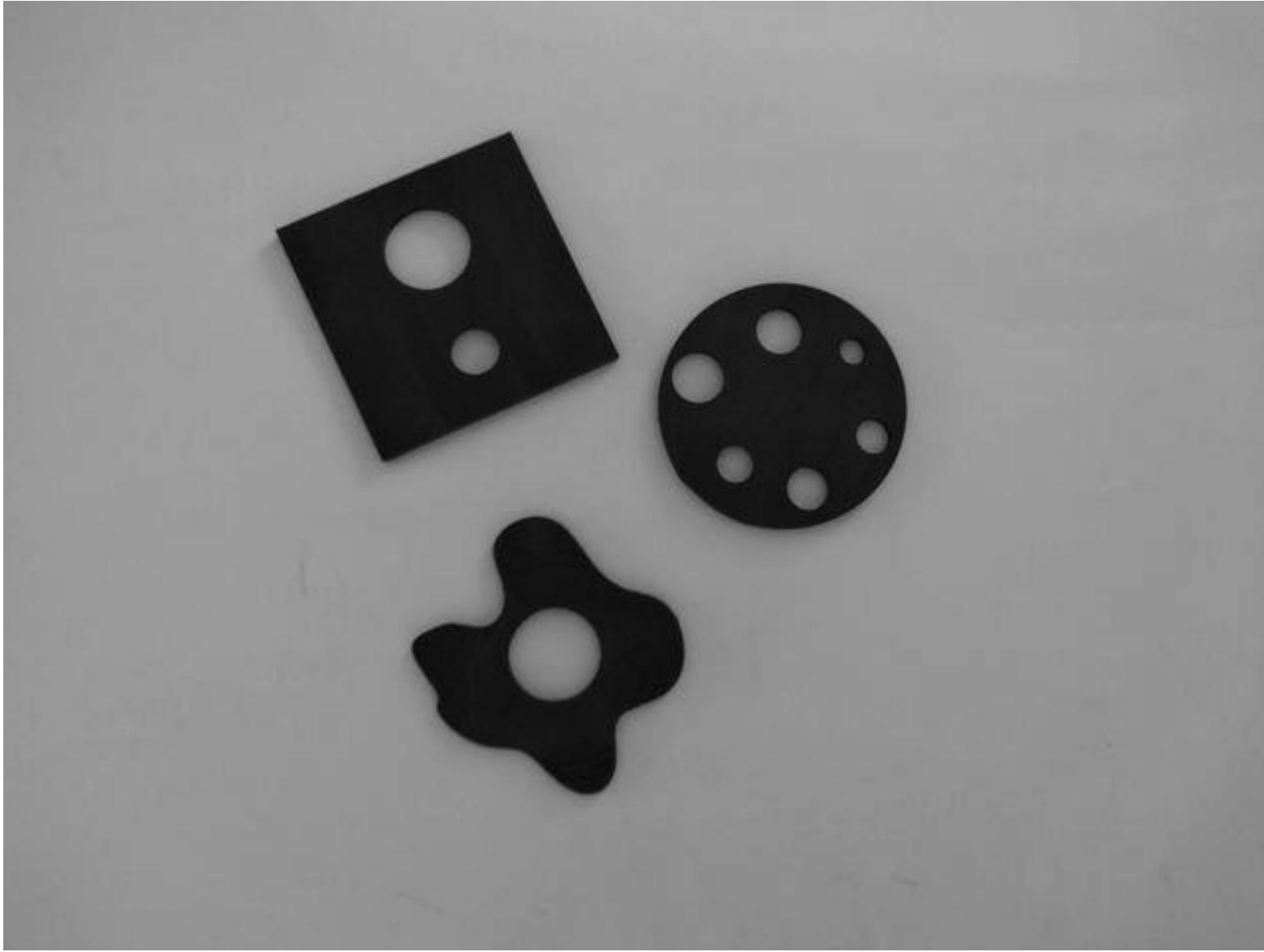


Imagen original

```
for x=1:n % se bare 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.  
title('Histograma de imagen original'); % se asigna título  
% disp(c);
```

```
for j=1:256
    i=c(j);
    if i>0
        break
    end
end
```

```
min_a=j-1;
```

```
for j=256:-1:1
    i=c(j);
    if i>0
        break
    end
end
```

```
max_a=j-1;
```

```
min_d=100; % se selecciona valor mínimo deseado.  
max_d=150; % se selecciona valor máximo deseado.  
  
for x=1:n % se barre imagen en renglones.  
    for y=1:m % se barre imagen en columnas.  
        p_in=f(x,y);  
        p_sal=(p_in-min_a) * ((max_d-min_d) / (max_a-  
min_a)) +min_d;  
        g(x,y)=p_sal;  
    end  
end  
  
figure(3) % se abre una ventana auxiliar.  
imshow(g) % se muestra imagen contrastada g(x,y).  
title('Imagen contrastada') % se asigna título de esta  
imagen.
```

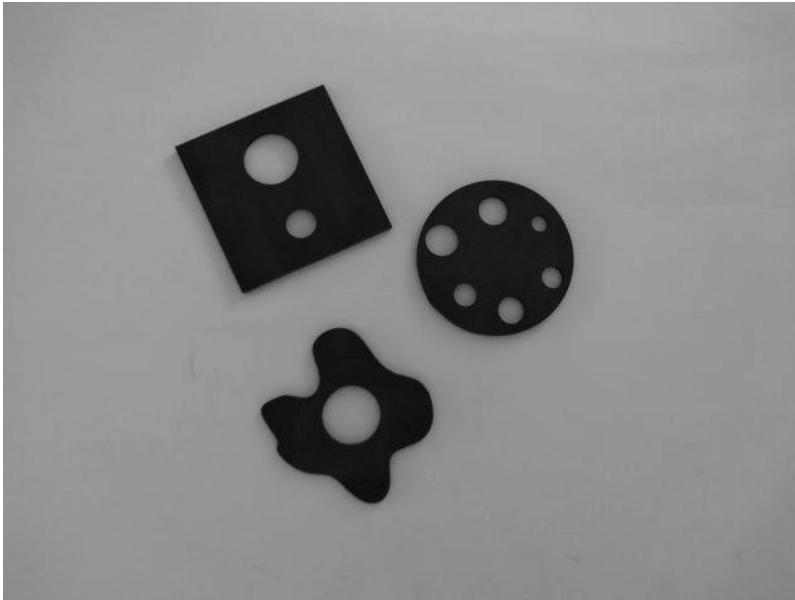
```

c=zeros(256,1); % inicializa en ceros 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=g(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 (4) % se abre una ventana auxiliar.
fill(niveles,c,'b' ); % dibuja histograma h(ri) de imagen.
title('Histograma de imagen contrastada'); % se asigna título.
% disp(c);

```



Imagen

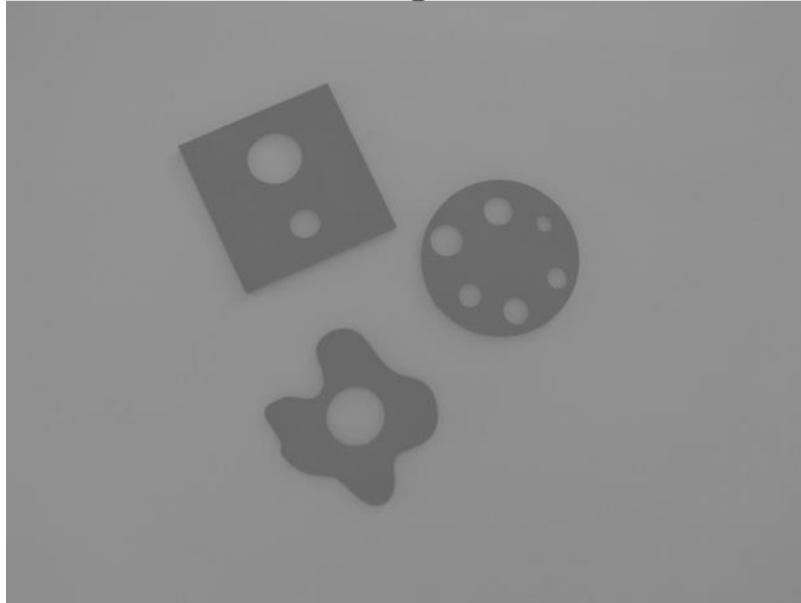
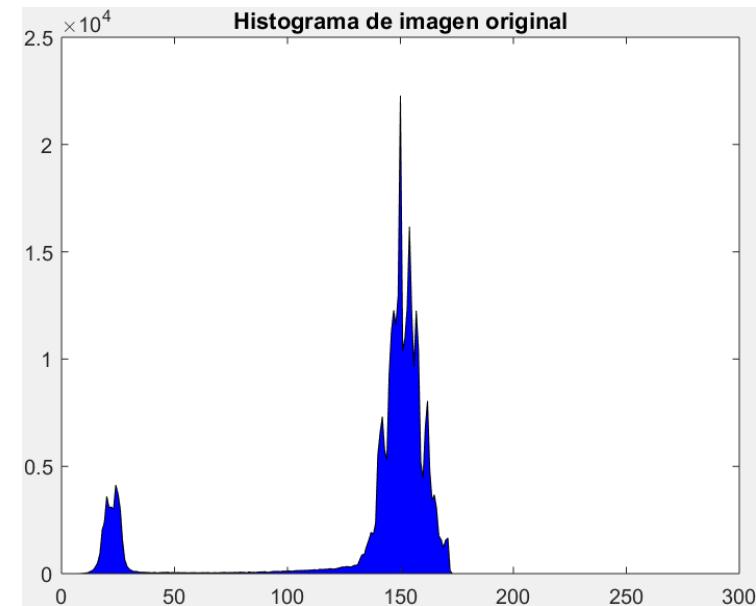
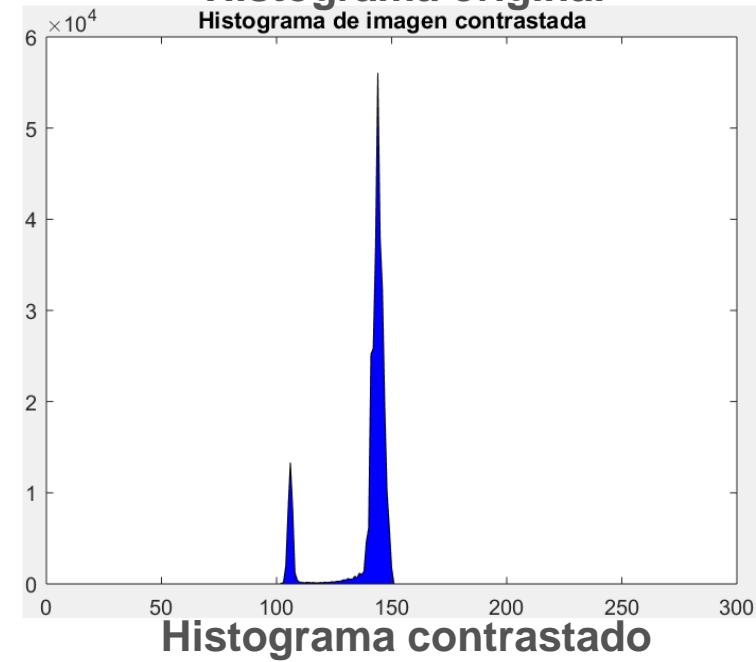


Imagen contrastada



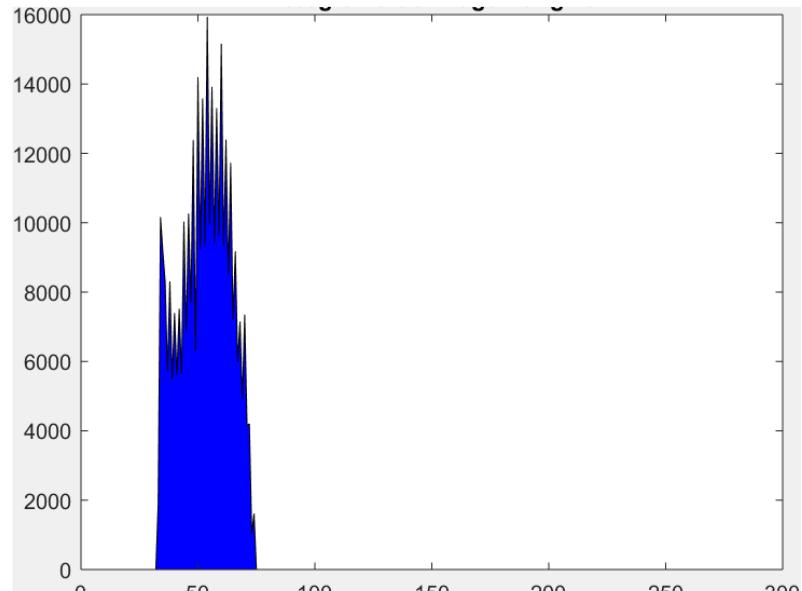
Histograma original



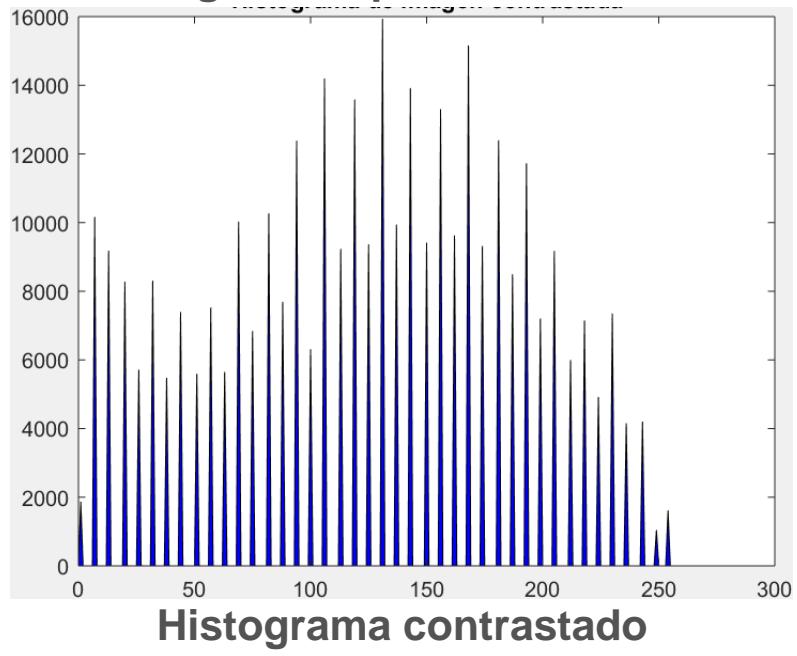
Histograma contrastado



**Imagen con poco contraste**



**Histograma original**



**Histograma contrastado**

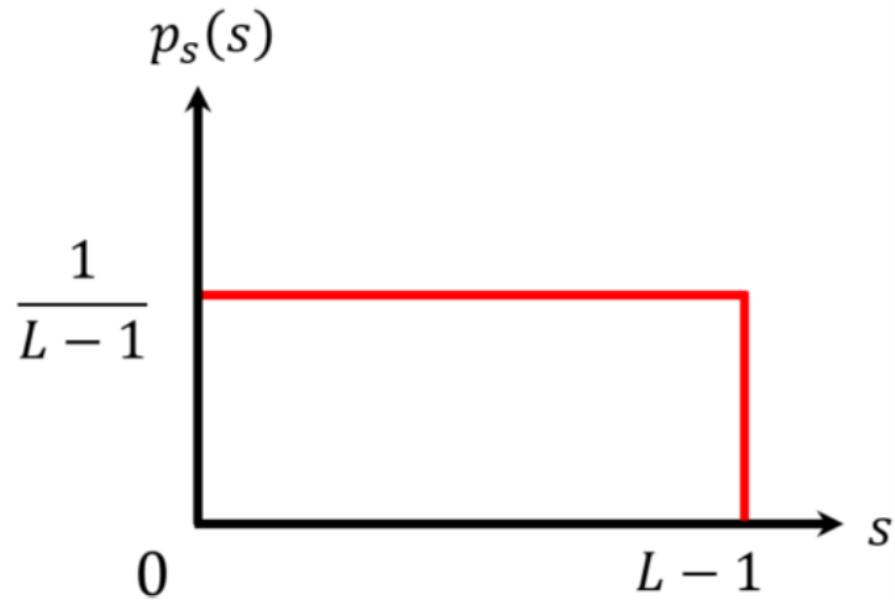
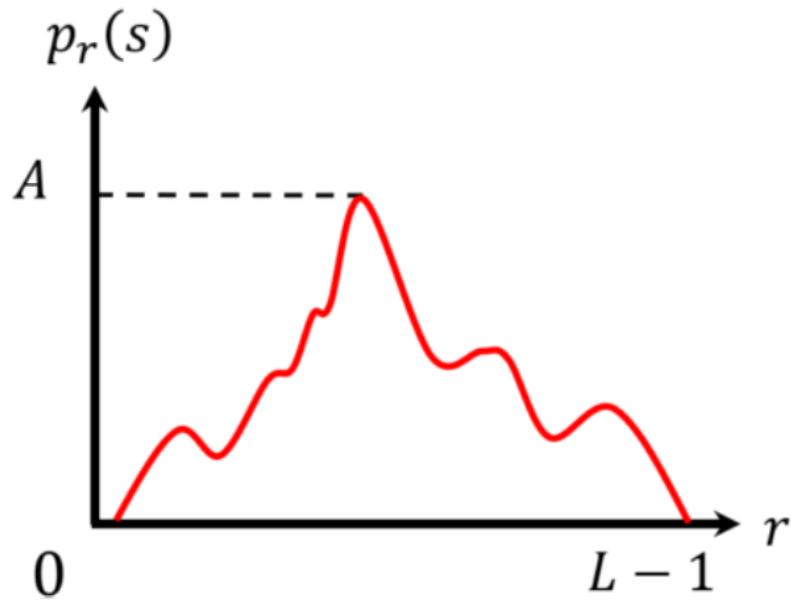


**Imagen contrastada**

## Ecualización del histograma de una imagen:

La EH es una transformación que busca obtener para una imagen de entrada  $f(x, y)$  un histograma con una **distribución uniforme**.

En otras palabras, que todos los niveles de gris tengan a la salida una distribución de probabilidad uniforme.



Dada una imagen  $M \times N$  píxeles, con  $n_k$  píxeles para cada nivel  $r_k$ .

La EH consiste en realizar la siguiente transformación sobre los niveles de intensidad de la imagen:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{M \times N} \sum_{j=0}^k n_j$$

Resulta en una dispersión del histograma en un rango mayor dentro del intervalo  $[0, L - 1]$ .

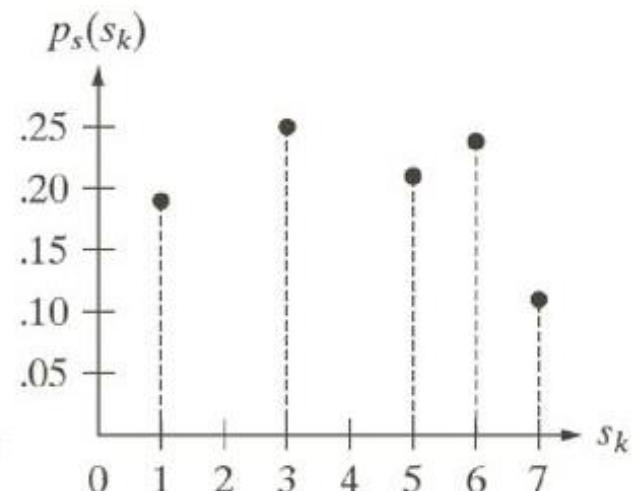
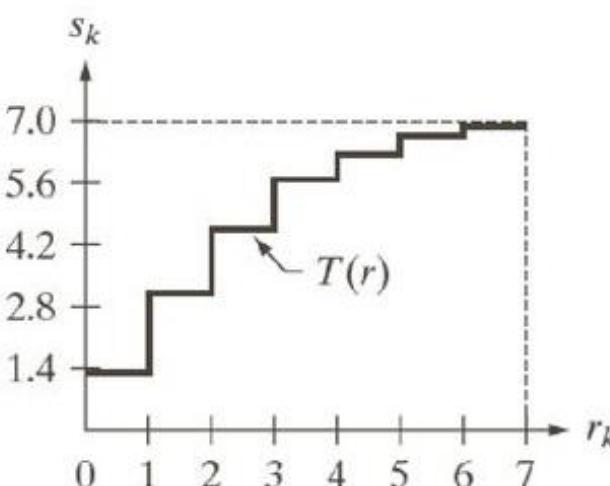
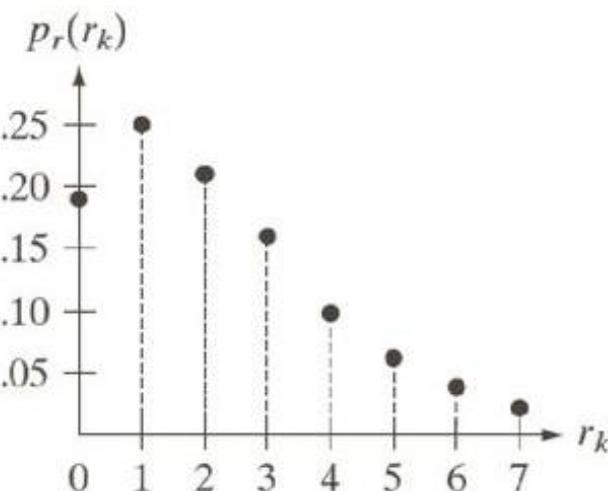
# Ejemplo:

$r_k$	$n_k$	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$s_k$

1.33  
3.08  
4.55  
5.67  
6.23  
6.65  
6.86  
7.00

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{M \times N} \sum_{j=0}^k n_j$$



```
% Programa para ecualizar una imagen f(x,y)

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

f=imread('Gato 01.JPG'); % se lee imagen de prueba
f(x,y)

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



Imagen original

```
g = histeq(f); % función para ecualizar imagen.

figure(2) % abre una ventana auxiliar.
imshow(g) % muestra imagen contrastada g(x,y).
title('Imagen ecualizada') % se asigna título de esta
imagen.

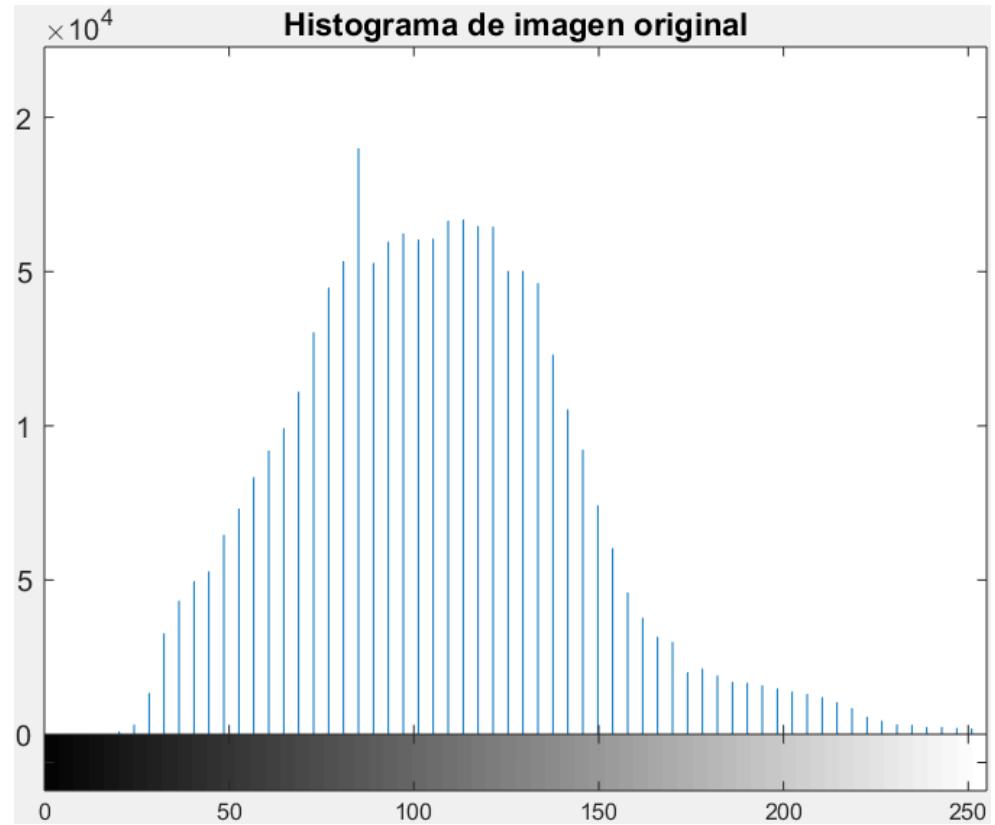
figure(3)
imhist(f, 64)
title('Histograma de imagen original') % asigna título.

figure(4)
imhist(g, 64)
title('Histograma de imagen ecualizada') % asigna
título.
```

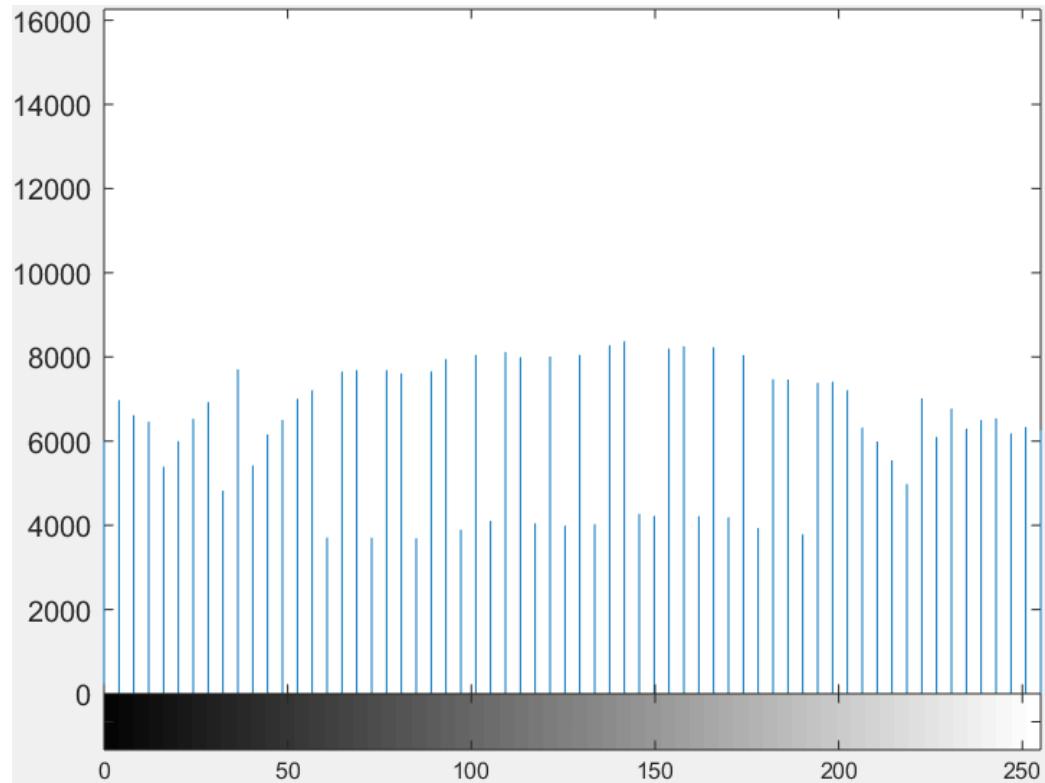
# Ejemplo:



Imagen original



su histograma



**Histograma ecualizado**



**Imagen ecualizada**

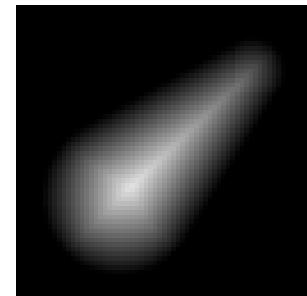
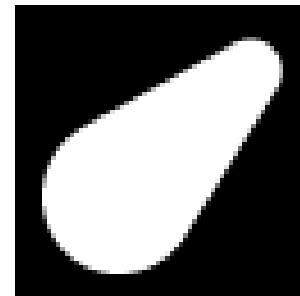
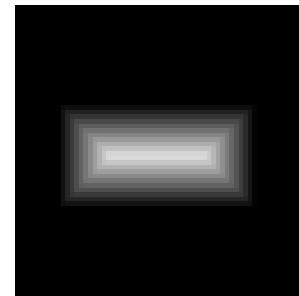
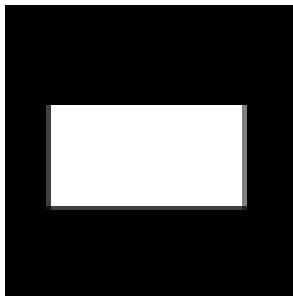
**Transformada  
distancia de una  
imagen:**

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

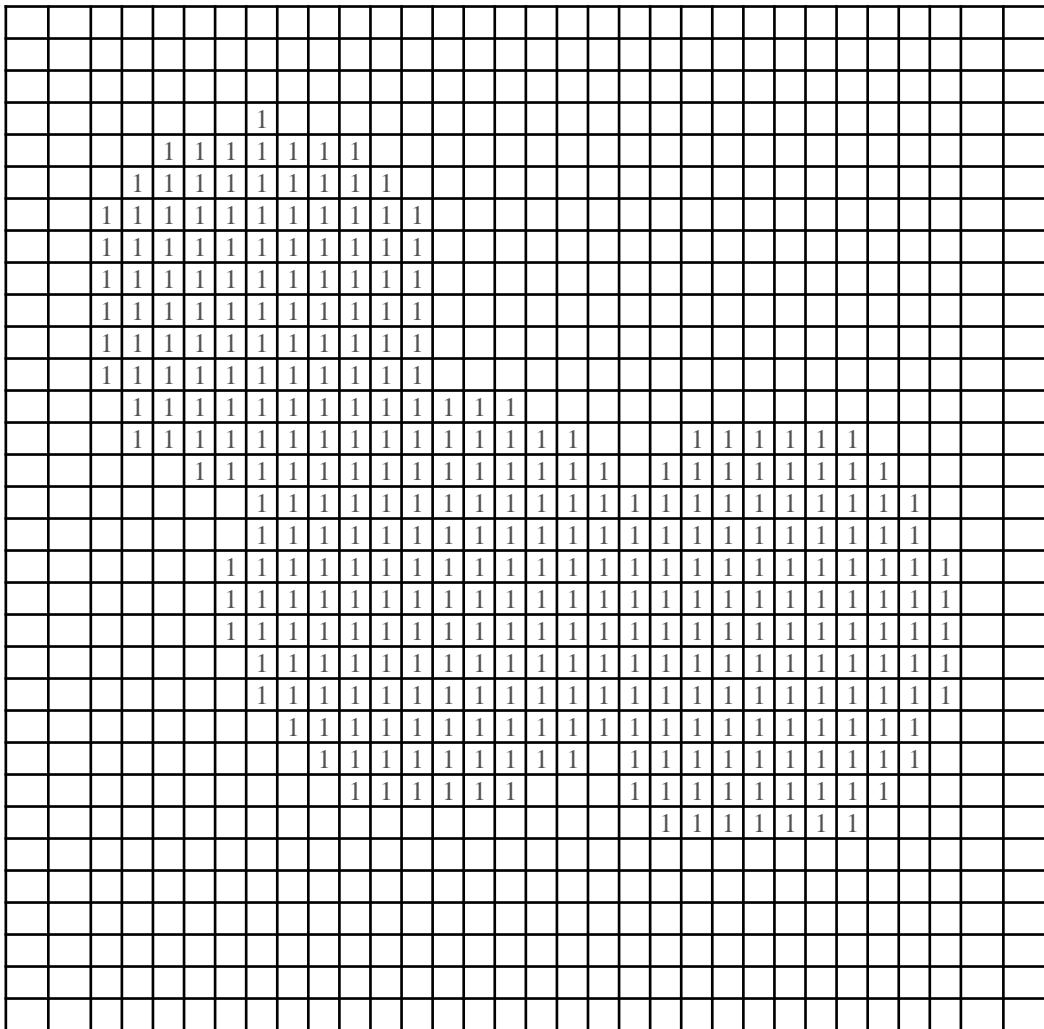
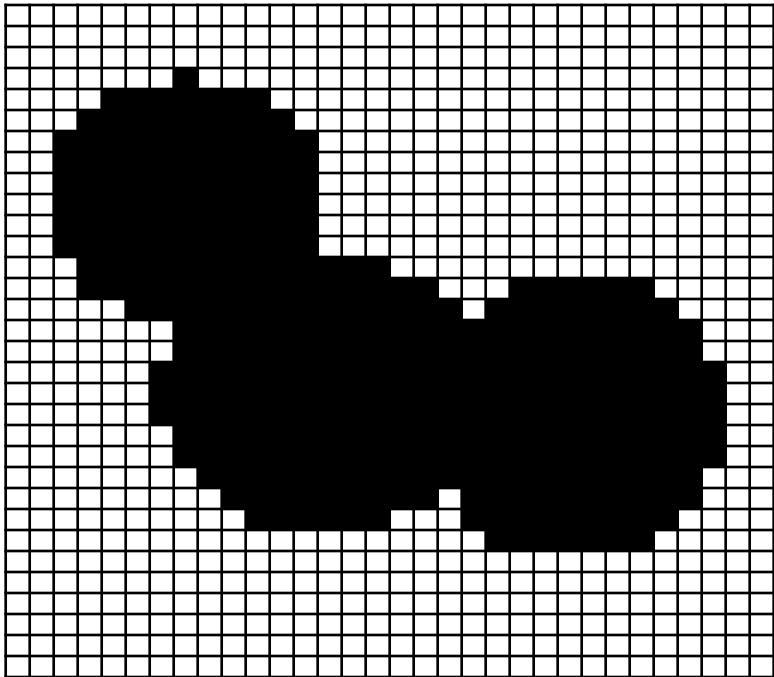
Binary Image

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

Distance transformation



La transformada distancia es la distancia más corta de un pixel hacia el fondo





# Applications of the Region Growing Euclidean Distance Transform: Anisotropy and Skeletons

Olivier Cuisenaire, Benoit Macq

Telecommunication and Remote Sensing Laboratory  
Université catholique de Louvain

## Abstract

A new region growing algorithm has recently been proposed for computing Euclidean Distance Maps in a time comparable to widely used chamfer DT. In this paper we show how this algorithm can be extended to more complex tasks such as the computation of distance maps on an-isotropic grids and the generation of a new type of Euclidean skeletons.

## 1. Introduction

A distance map is an image where the value of each pixel of the foreground is the distance to the nearest background pixel (NBP). These maps have numerous applications in pattern recognition, medical image registration [9], active contour models [8], ... Efficient algorithms to produce distance maps can be grouped in two categories: Vector Distance Transform (VDT) first proposed by Danielsson [1,4] or an approximation of it, the Chamfer Distance Transform (CDT) proposed by Borgerfors in [2,3]. This offers various trade-off between speed and quality of the approximation.

In [7], we improved the VDT by using a new order for scanning the image, thus significantly fastening the computations. In section 2 we will briefly summarize this method. Then, in section 3, we show how this can be applied to an-isotropic data, which is necessary for many 3D imaging systems for which the data is sampled with an inter-slice voxel size larger than the intra-slice voxel size. Finally, in section 4 we study one of the main applications of distance maps, the generation of morphological skeleton of an object. We show how the properties of the

propagation scheme of our DT algorithm can be used to produce a new type of skeleton.

## 2. The Region Growing EDT

The RGEDT algorithm, fully described in [7] is a variation of the Vector Distance Transform described in [4]. The vector localizing the NBP for each pixel is propagated to his neighboring pixels. The originality of RGEDT is that, instead of using a raster scan, the pixels are scanned by increasing value of the distance by using a hierarchical queue (HQ) data structure (fig 1).

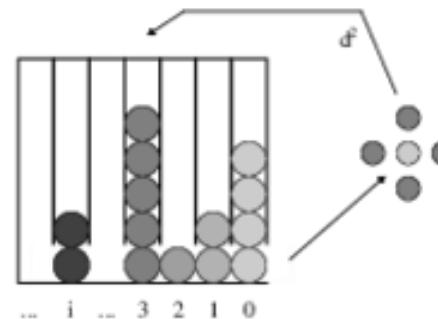


Figure 1: Hierarchical Queues are made of an ordered collection of FIFO queues. In-going elements are added to any of the queues, outgoing elements are taken from the smallest numbered non-empty queue. Each element includes the location of the pixel and of its NBP.

For each pixel we consider that its neighbors have the same NBP. If this leads to a distance lower than

# Watershed Segmentation based on Distance Transform and Edge Detection Techniques

Pinaki Pratim Acharya  
Assistant Professor, Department of CSE  
Bengal Institute of Technology and Management  
Santiniketan, West Bengal, Pin - 731236

Dibyendu Ghoshal  
Associate Professor, Department of ECE  
National Institute of Technology  
Agartala, Tripura, Pin - 799001

## ABSTRACT

An edge detection algorithm for digital images is proposed in this paper. Edge detection is one of the important and most difficult tasks in image processing and analysis. In images edges can create major variation in the picture quality where edges are areas with strong intensity contrasts. Edges in digital images are areas with strong intensity contrasts and a jump in intensity from one pixel to the next can create major variation in the picture quality. This paper proposed an effective edge detection algorithm based morphological edge detectors and watershed segmentation algorithm using distance transform. The result confirms that the proposed algorithm is found to yield satisfactory and efficient segmentation of the digital images for edge detection. Experimental result presented in this paper is obtained by using MATLAB.

## Keywords

Edge detection, Segmentation, Distance Transform, Watersheds.

## 1. INTRODUCTION

Image segmentation has been an essential step in digital image processed for most subsequent image analysis and image understanding activities. In specific, many of the prevailing method for image description and recognition [1-2], image visualization [3-4], and object based image compression [5-7], highly depend on the segmentation result obtained from previous stage. The segmentation problem [10], [12] involves separating or partitioning of an image into a number of homogenous and spatially connected group of pixels called super pixels, so it has addition of two neighborhood segments produces a heterogeneous segment. As an alternative approach, segmentation can be thought of as a pixel labeling cum mathematical morphological process in the same that all the pixels belonging to the same homogeneous region are assigned the same label and theme and several techniques to define homogeneity of a region based on a particular objective of image segmentation process.

Stellar images of various planets, stars, galaxies, comets, nebula and other various extra terrestrial moving objects have occupied a significant place in radio astronomy, space research and preparation of distance space vehicle movements (manual or robotized) [8]. The images of stellar objects have been acquired either through satellite imagery or Hubble space telescope. These images are very much prone to be affected by a verity of noise like Gaussian noise, Rayleigh noise, Impulse noise and Speckle noise. At the same time, the contrast of these images have been generally found to be low to very long distance and atmosphere turbulence and disturbance and variation of temperature during the

transmission of them through space media. The segmentation of low contrast images have always posed problem due to processing problem arising from inability to directly apply the watershed algorithm on the image. Rather it has been found to be more suitable to apply watershed to the gradient of the image instead of the original image [20]. In this approach, the original minima of catchment basins correlate with the small values of the image gradient value. Thus the study of the effect of application of varies modified version or combination of morphological based watershed algorithm [9], [14-19] on low contrast images of stellar objects have become worthwhile and interesting.

A good number of methods have been proposed and applied to solve the problem related to digital image segmentation [11], [13]. These techniques can be broadly classified into the following categories as histogram based techniques, edge based techniques, region based techniques, markov random field based techniques and hybrid techniques. All the above techniques have got certain advantages as well as some complexity compared to the others, the present technique would fall within the group of processing having adaptive histogram equalization as a preprocessing step for low contrast stellar images.

## 2. THE WATERSHED SEGMENTATION

Watershed algorithm is based on morphological process although it can be mixed up with edge based segmentation to yield a hybrid technique. In geography, watersheds are the ridge line that divides different areas called catchment basins drained by different river systems. In mathematical morphology, a gradient image may be considered as a topographical surface where the numerical value of each pixel indicates the elevation of their points. The set of pixels along which the gray levels changes sharply gives rise to an edge. The watershed algorithm applies these ideas to solve a variety of image segmentation problems.

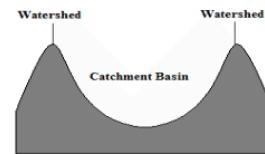


Fig 1: Watershed segmentation-local minima of gray level yield catchment basins, local maxima define the watershed lines.

0 0 0 0 1 1 1 1 1 1 0 0 0 0

0 0 0 0 1 2 3 4 3 2 1 0 0 0 0

0 0 0 0 1 1 1 1 1 1 0 0 0 0

→ 0 0 0 0 1 2 3 4 5 6 7 0 0 0 0

0 0 0 0 1 1 1 1 1 1 0 0 0 0

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

0 0 0 0 1 2 3 4 3 2 1 0 0 0 0



# Otro ejemplo:

0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0

## Otro ejemplo:

0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0

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

## Otro ejemplo:

0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0

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

0 0 1 2 3 2 1 0 0 0 0 1 2 3 4 3 2 1 0 0 0



¿Cómo hago esto mismo en 2-D?

# TWO ALGORITHMS FOR COMPUTING THE EUCLIDEAN DISTANCE TRANSFORM

MARINA L. GAVRILOVA

*Dept of Computer Science, University of Calgary  
Calgary, AB, Canada, T2N1N4  
E-mail: marina@cpsc.ucalgary.ca*

MUHAMMAD H. ALSUWAIYEL

*Department of Information and Computer Science  
King Fahd University of Petroleum & Minerals  
Dhahran 31261, Saudi Arabia  
E-mail: suwaiyel@ccse.kfupm.edu.sa*

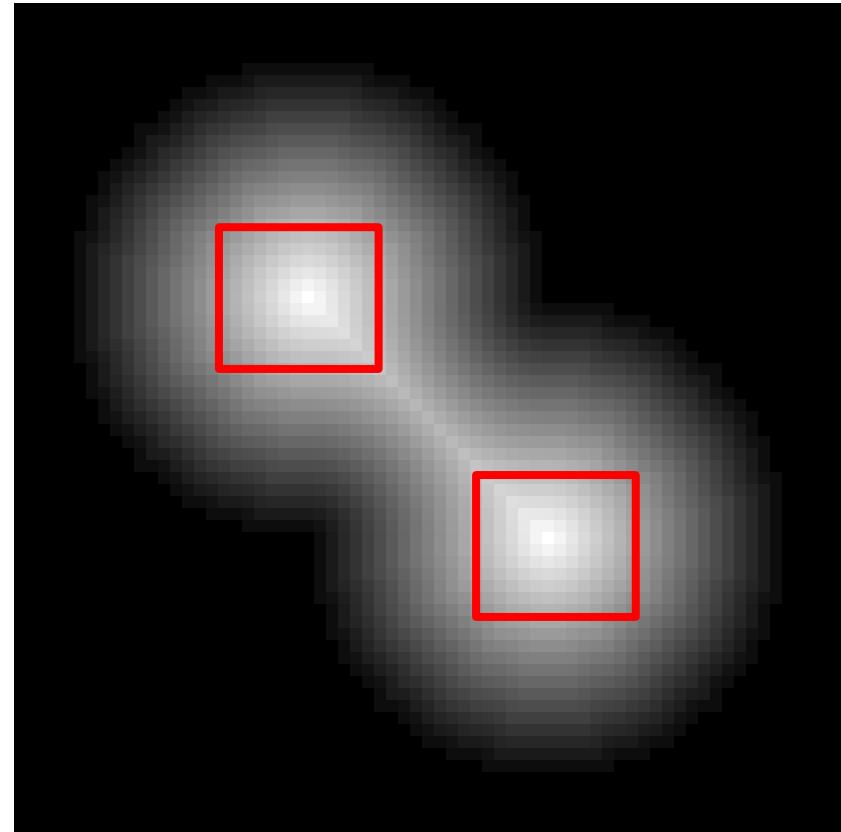
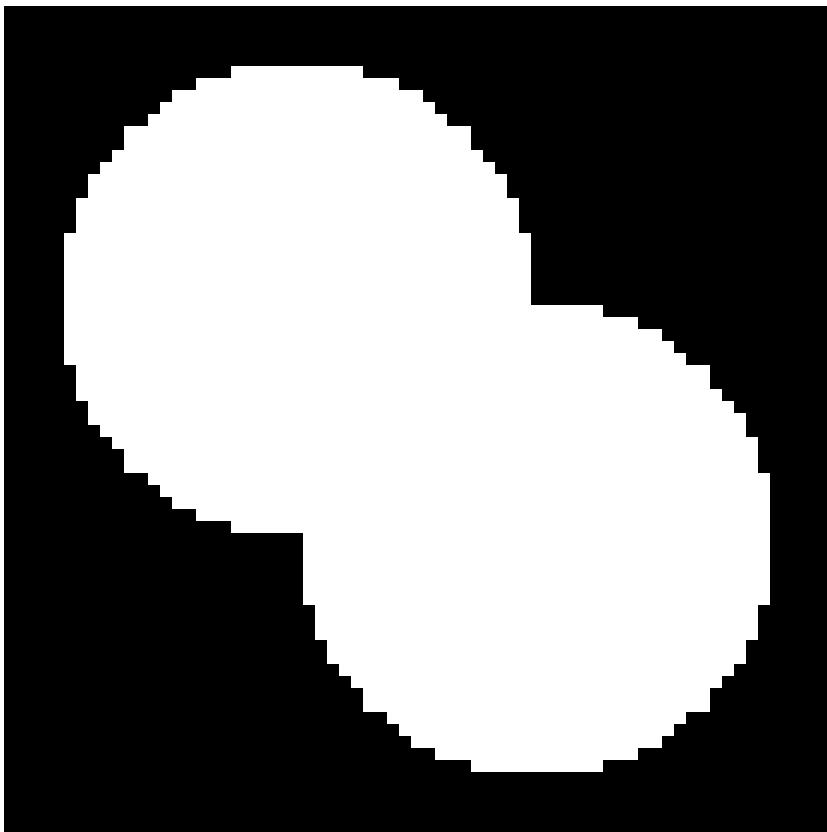
**Abstract.** Given an  $n \times n$  binary image of white and black pixels, we present two optimal algorithms for computing the distance transform and the nearest feature transform using the Euclidean metric. The first algorithm is a fast sequential algorithm that runs in linear time in the input size. The second is a parallel algorithm that runs in  $O(n)$  time on a linear array of  $n$  processors.

**Keywords:** Feature transform; Distance transform; Euclidean distance; Parallel algorithms; Image processing.

## 1. Introduction

Given an  $n \times n$  binary image  $\mathcal{I}$  of white and black pixels, the distance transform of  $\mathcal{I}$  is a map that assigns to each pixel the distance to the nearest black pixel, referred to as *feature*. The feature transform of  $\mathcal{I}$  is a map that assigns to each pixel the feature that is nearest to it. The distance transform was first introduced by Rosenfeld and Pfaltz [1], and it has a wide range of applications in image processing, robotics, pattern recognition and pattern matching [?]. The distance metrics used to compute the distance transform include the  $L_1$ ,  $L_2$  and  $L_\infty$  metrics, with the  $L_2$  (Euclidean) metric being the most natural, and rotational invariant.

Several algorithms have been proposed for these metrics. One approach is to grow clusters or neighborhoods around each feature consisting of those pixels whose nearest feature is  $p$ . This approach has been taken in [2] and [3] to obtain sequential and parallel algorithms, respectively. Daniellson [2] describes



% Transformada Distancia de una imagen.

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

```
f=imread('Circulos_01.BMP');  
figure(1)  
imshow(f)  
title('Imagen original')
```

# Imagen con 19 objetos con solapamiento:

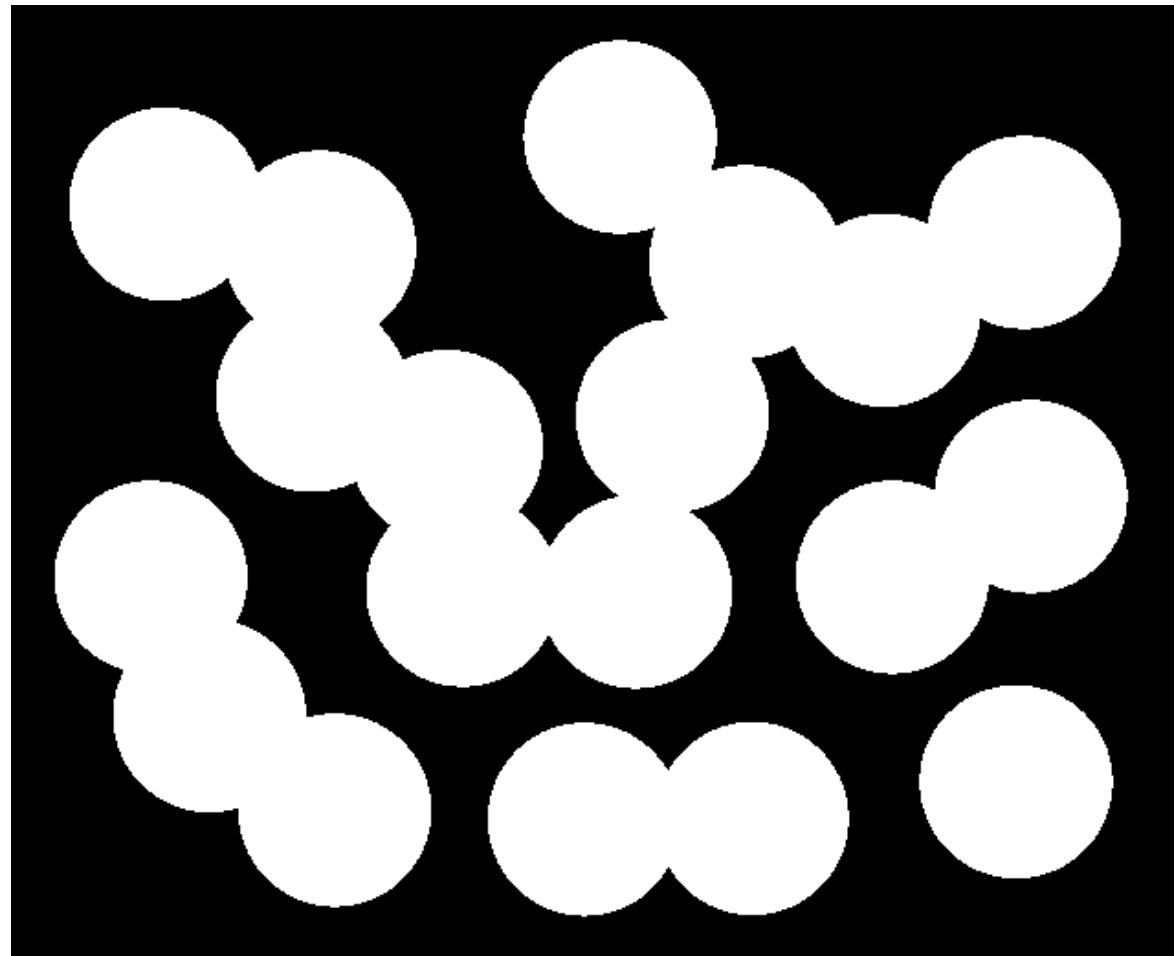
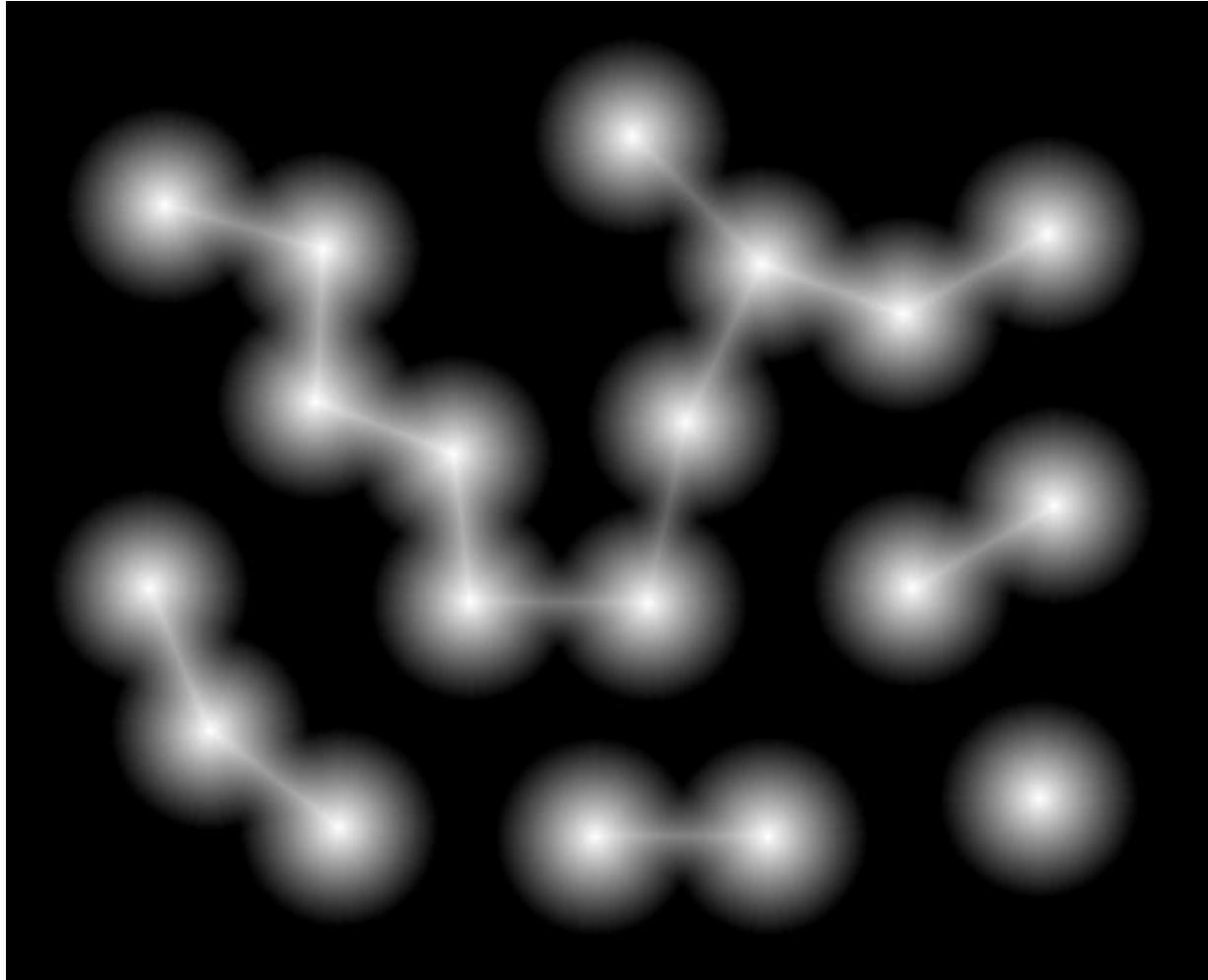


Imagen original

```
D = bwdist(~f); % Transformada distancia de  
imagen binaria.  
% Para cada píxel en ~f, la TD asigna un  
número que es la distancia  
% entre el píxel y su píxel más cercano 0 en  
~f.  
figure(2)  
imshow(D, [ ])  
title('Imagen DT')
```



**Imagen DT:**

```

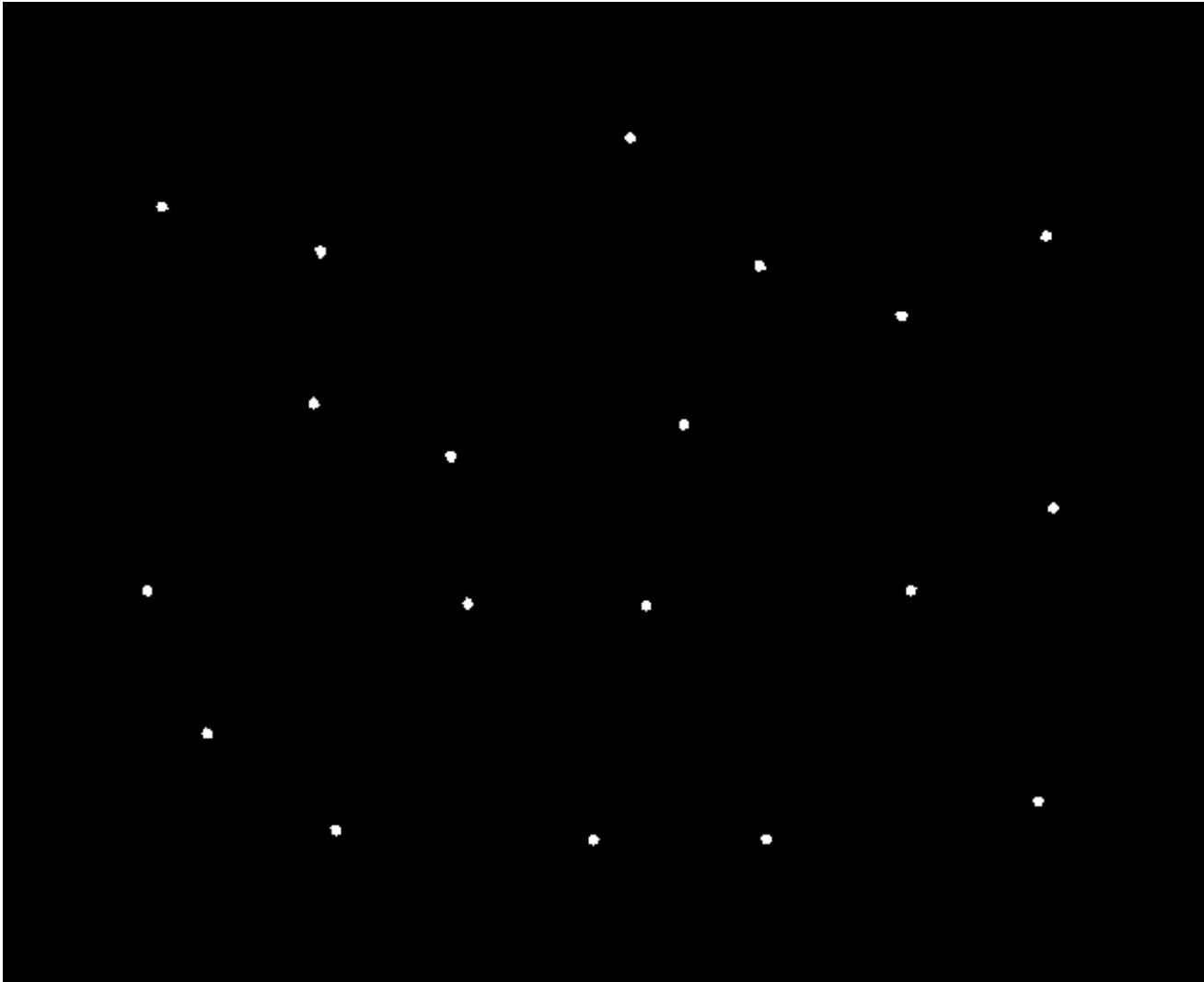
[n,m]=size(f); % dimensiones de la imagen f(x,y)
th=50.0; % umbral para binarizado de imagen.

for x=1:n % se barre imagen en renglones.
    for y=1:m % se barre imagen en columnas.
        i=D(x,y); % se obtiene el nivel de gris de f(x; y) .
        if i>=th
            % disp(i);
            g(x,y)=255;
        else
            g(x,y)=0;
        end
    end
end

figure(3)
imshow(g)
title('Imagen binaria')

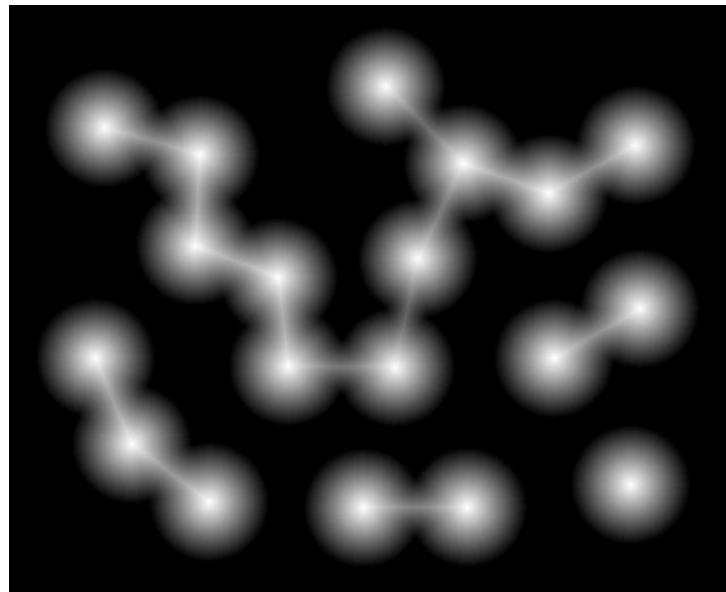
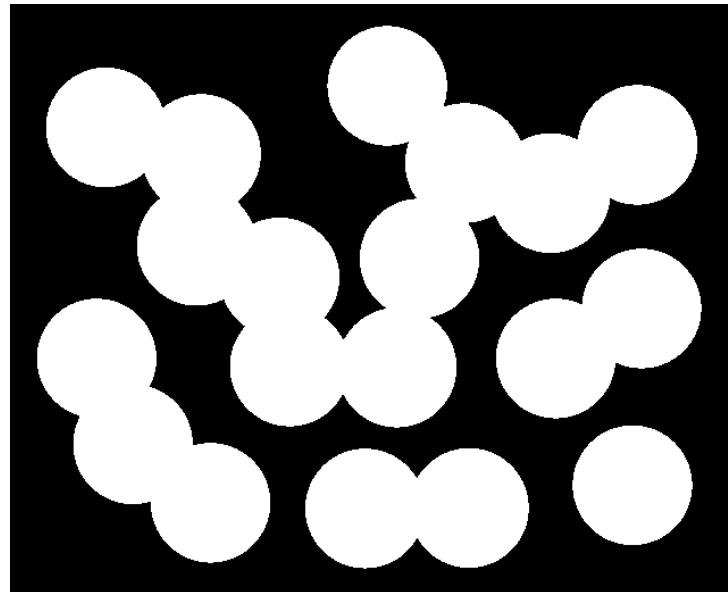
[L,n] = bwlabel(g);
X = ['Número de objetos en imagen: ',num2str(n)];
disp (X); % Despliega el número de objetos etiquetados.

```



**Imagen umbralada mostrando el número de objetos:**

**Imagen con 19  
objetos con  
solapamiento:**



**Imagen DT:**

**Imagen umbralada mostrando el  
número de objetos: 19**

# A clustering technique based on the distance transform\*

V. Starovoitov\*

Institute of Engineering Cybernetics, Belarussian Academy of Sciences, 6, Surganov St., 220012 Minsk, Belarus

Received 10 December 1994; revised 7 December 1995

## Abstract

We present a new variant of a geometric approach to unsupervised clustering. It is based on the digital distance and cost function transforms. We map the given set of real continuous data onto an  $n$ -dimensional binary image, where black pixels correspond to the observations. A way of such discretization is suggested. Domains with some concentration of black pixels are extracted as cluster cores. Clusters are detected by the mentioned transforms.

**Keywords:** Cluster analysis; Discretization; Binary image; Digital morphology; Distance transform; Digital metric; Cost function

## 1. Introduction

Clustering divides a set of feature points or patterns into subsets. A cluster is a homogeneous group of patterns which are "like" one another. "Likeness" between objects can be determined by the distance between the patterns. The following four conditions may be employed under which a pair of data groups can be considered as distinct clusters (Mizoguchi and Kakusho, 1982).

**Condition A.** The two groups of data are not in touch with each other.

**Condition B.** The difference in density between the two groups of data is larger than a certain index.

**Condition C.** The size of the touching region of the two groups of data is smaller than a certain index.

\*This work was supported in part by a grant from Deutscher Akademischer Austauschdienst (DAAD).

\* Email: muhan@newman.basnet.minsk.by

**Condition D.** The difference in density between the touching regions and each group is greater than a certain index.

Unsupervised clustering solves two main problems:  
 (1) detection of the clusters,  
 (2) construction of the cluster criterion.

The geometric approach to unsupervised cluster extraction in the space of numerical features solves the first problem by a distance threshold  $d_{th}$  (Mandel and Mirkin, 1987), and the second one by attaching feature points not falling in any detected cluster to the nearest cluster.

Postaire et al. (1993) reported that the observed data can be mapped into a discrete space and can be clustered by the morphological operations of erosion or opening. Binary Mathematical Morphology (MM) (Serra, 1982) and the Distance Transform (DT) (Rosenfeld and Pfaltz, 1968) are used for the analysis of geometrical structures in image processing. A variant of the geometric approach based on the use of the DT is developed. The Euclidean and Maha-

# EVALUATION OF DISTANCE TRANSFORM BASED ALTERNATIVES FOR IMAGE SEGMENTATION OF OVERLAPPING OBJECTS

## EVALUACIÓN DE ALTERNATIVAS BASADAS EN LA TRANSFORMADA DE DISTANCIA PARA LA SEGMENTACIÓN DE IMÁGENES DE OBJETOS QUE SE SOLAPAN

Chinea Valdés, Lyanett<sup>1</sup>, Lorenzo Ginori, Juan Valentín<sup>1</sup>

<sup>1</sup> Centro de Estudios de Electrónica y Tecnologías de la Información, Facultad de Ingeniería Eléctrica, Universidad Central "Marta Abreu" de Las Villas, Cuba, juan@uclv.edu.cu

## ABSTRACT

In the field of image segmentation, the case of objects that are touching or overlapping is frequently found. These require the use of appropriate techniques to achieve a good segmentation. Examples of this are found in digital image processing for cellular imaging, in the microscopy analysis of rocks and others. One way to address this problem is by combining the use of the distance transform and the watershed transform. In this case, it is particularly important to build adequate markers for the watershed transform. The ways of obtaining and using the markers influence the quality of segmentation, as well as the level of human participation in the process. The latter is determinant in regard of time consumption, and is particularly important when a large number of images are to be processed. In this work, various alternatives for building the markers are evaluated, which are oriented to avoid the human participation in the process. To evaluate the quality of the segmentation results, these were compared through calculation of the Jaccard index, with those obtained by means of manual segmentation of the selected images. Tables with the quantitative results are exhibited and graphical examples are shown as well. Application of the classical distance transform using as markers the centroids of the extended minima obtained from the application of the H-minima transform, exhibited the best results.

**Keywords:** Segmentation, distance transform, overlapped images, watersheds.

## RESUMEN

En el campo de la segmentación de imágenes, es frecuente encontrar casos de objetos que se tocan o se solapan. Estos requieren del uso de técnicas apropiadas para lograr una buena segmentación. Ejemplos de esto se encuentran en el procesamiento digital de imágenes para imaginología celular, en el análisis microscópico de rocas y otros. Una forma de abordar este problema es combinando el uso de la transformada de distancia y la transformada watershed. En este caso, es particularmente importante construir marcadores adecuados para la transformada watershed. La forma de obtener y utilizar los marcadores, influye sobre la calidad de la segmentación y sobre el grado de participación humana en el proceso. Esto último es determinante en cuanto al consumo de tiempo, y tiene particular importancia cuando se requiere procesar grandes cantidades de imágenes. En el presente trabajo se evalúan varias alternativas para la construcción de los marcadores, orientadas a prescindir de la participación humana en el proceso. Para evaluar la calidad de los resultados de la segmentación, estos fueron comparados, mediante el cálculo del índice de Jaccard, tomando como referencia los obtenidos mediante la segmentación manual de las imágenes seleccionadas. Se exhiben tablas con los resultados cuantitativos y se muestran también ejemplos gráficos de los mismos. La aplicación de la transformada de distancia clásica, usando como marcadores los centroides de los mínimos regionales obtenidos mediante la transformada H-minima, exhibió los mejores resultados.

**Muchos problemas pueden ser  
resueltos mediante una red  
neuronal**

**¿Cuándo usar una red  
neuronal?**

**Si podemos a mano encontrar una solución analítica para un problema, no usar una red neuronal.**

**Inversamente, si nos es muy difícil encontrar una solución analítica, entonces usar a una RNA.**

**El éxito de las RNAs reside en el hecho que permiten resolver problemas complejos.**

**Un problema asociado es que no son interpretables y explicables al 100%.**

# **Operaciones de expansión y encogimiento:**

**Expansión:** Algunos píxeles adyacentes a las regiones son cambiados de 0 a 1. Las regiones son engrosadas.

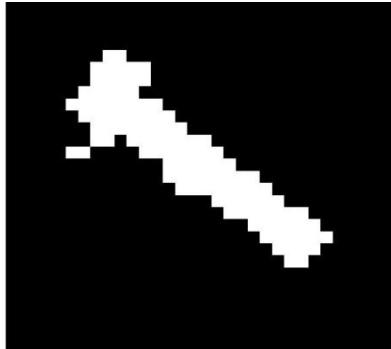
1	1	1
0	0	1
0	0	1

1	1	1
0	1	1
0	0	1

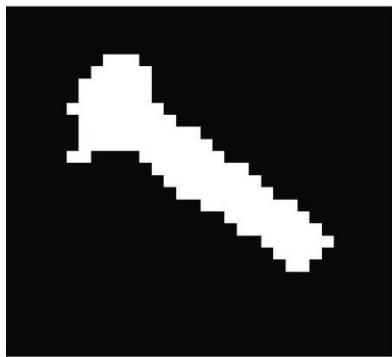
**Encogimiento:** Algunos píxeles adyacentes a las regiones son cambiados de 1 a 0. Las regiones son adelgazadas.

0	1	1
0	1	1
0	0	1

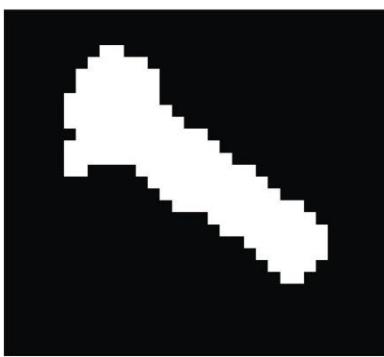
0	1	1
0	0	1
0	0	1



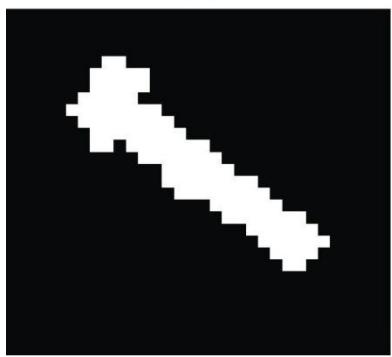
(a)



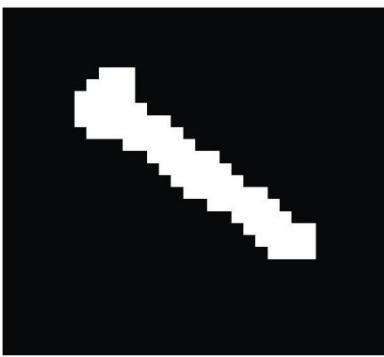
(b)



(c)



(d)

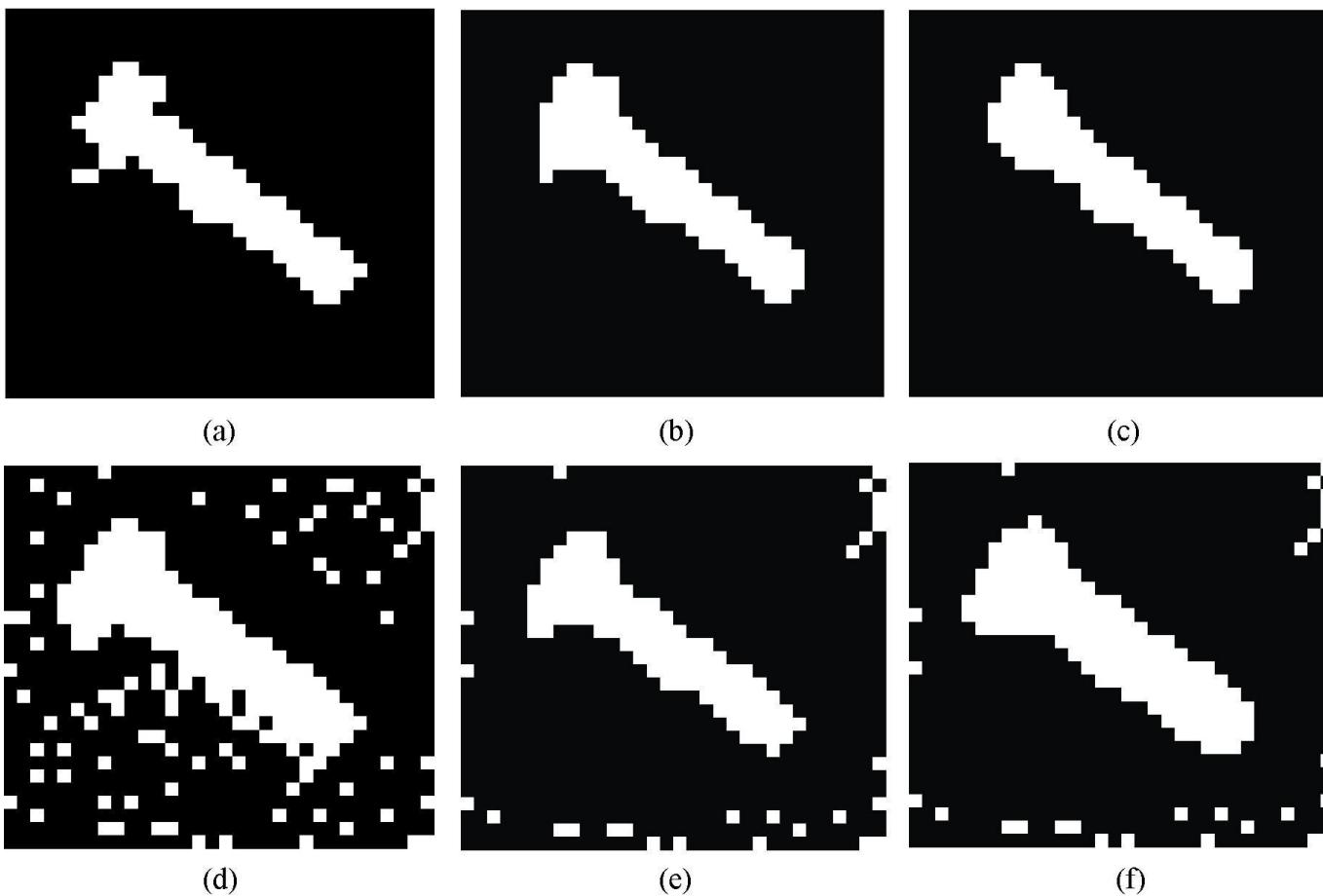


(e)

Aplicación de las operaciones de expansión y encogimiento sobre una imagen binaria.

- (a) Imagen original.
- (b) Expansión de (a) con  $n = 2$ .
- (c) Expansión de (a) con  $n = 6$ .
- (d) Encogimiento de (a) con  $n = 6$ .
- (e) Encogimiento de (a) con  $n = 4$ .

$n$  es el número de vecinos del píxel en cuestión.



Efecto de la aplicación de una **expansión** seguida de un **encogimiento** y un **encogimiento** seguido de una **expansión** sobre una imagen. (a) Imagen original. (b) Resultado de expandir (a). (c) Resultado de encoger (b). (d) Imagen original. (e) Resultado de encoger (d). (f) Resultado de expandir (e).

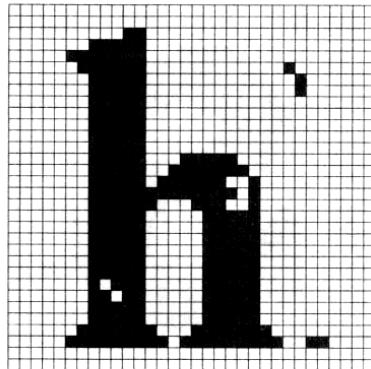
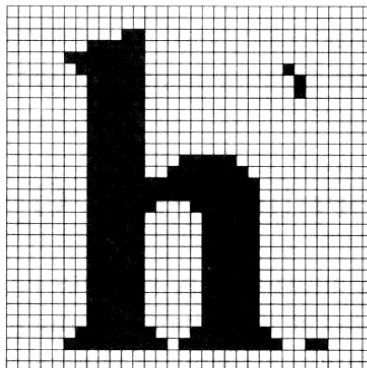
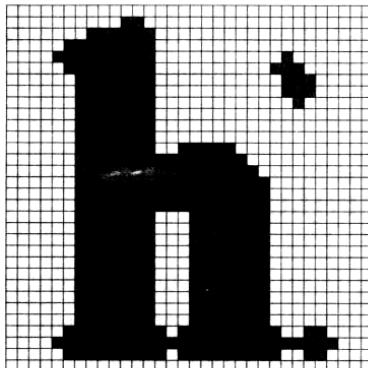
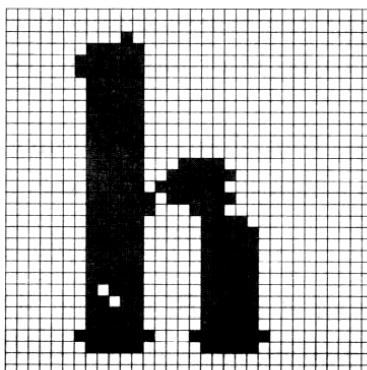
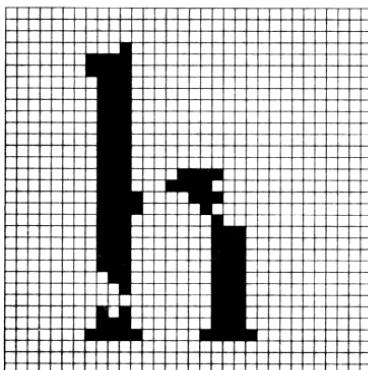


Imagen original



Expansión seguida de un encogimiento.

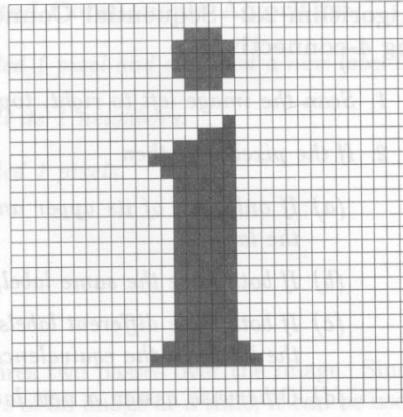
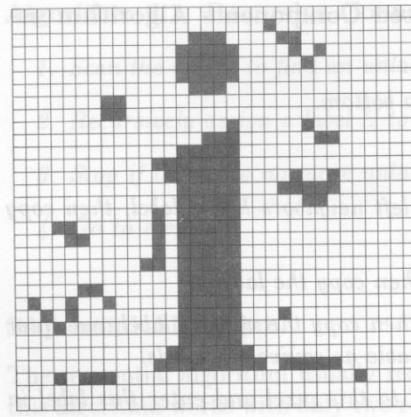


Encogimiento seguido de una expansión.

# FILTRADO POR TAMAÑO:

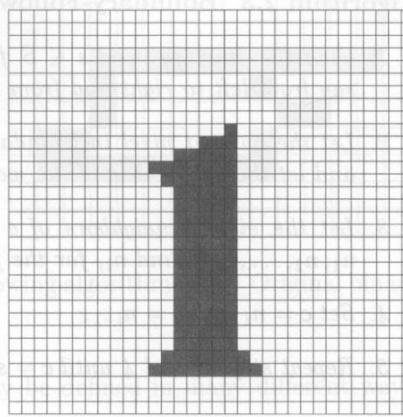
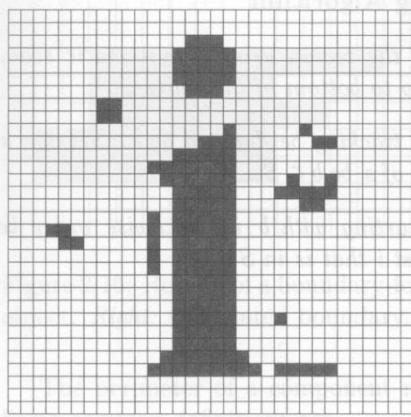
**Filtrado por tamaño:** Remover regiones pequeñas en la imagen cuyo tamaño sea inferior a un umbral dato  $T$ :

$T=10$ :



Adecuado.

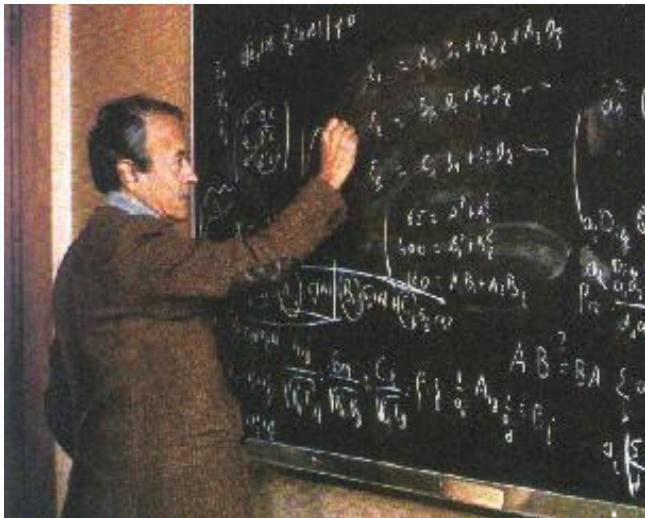
$T=25$ :



Muy grande.

# Morfología matemática:

# MORFOLOGÍA MATEMÁTICA



**PROF. GEORGES  
MATHERON**

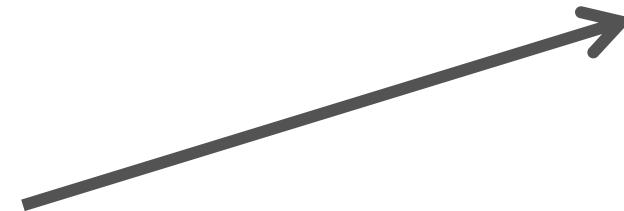


**PROF. JEAN  
SERRA**



**PROF. HERMANN  
MINKOSKY**

LA BASE DE LA MM SON LAS OPERACIONES  
ENTRE CONJUNTOS DESARROLLADAS POR



# MORFOLOGÍA MATEMÁTICA



1968

JEAN SERRA



CENTRO DE MORFOLOGÍA  
MATEMÁTICA

# **APLICACIONES DE LA MM**

**MINERALOGÍA,**

**PETROGRAFÍA,**

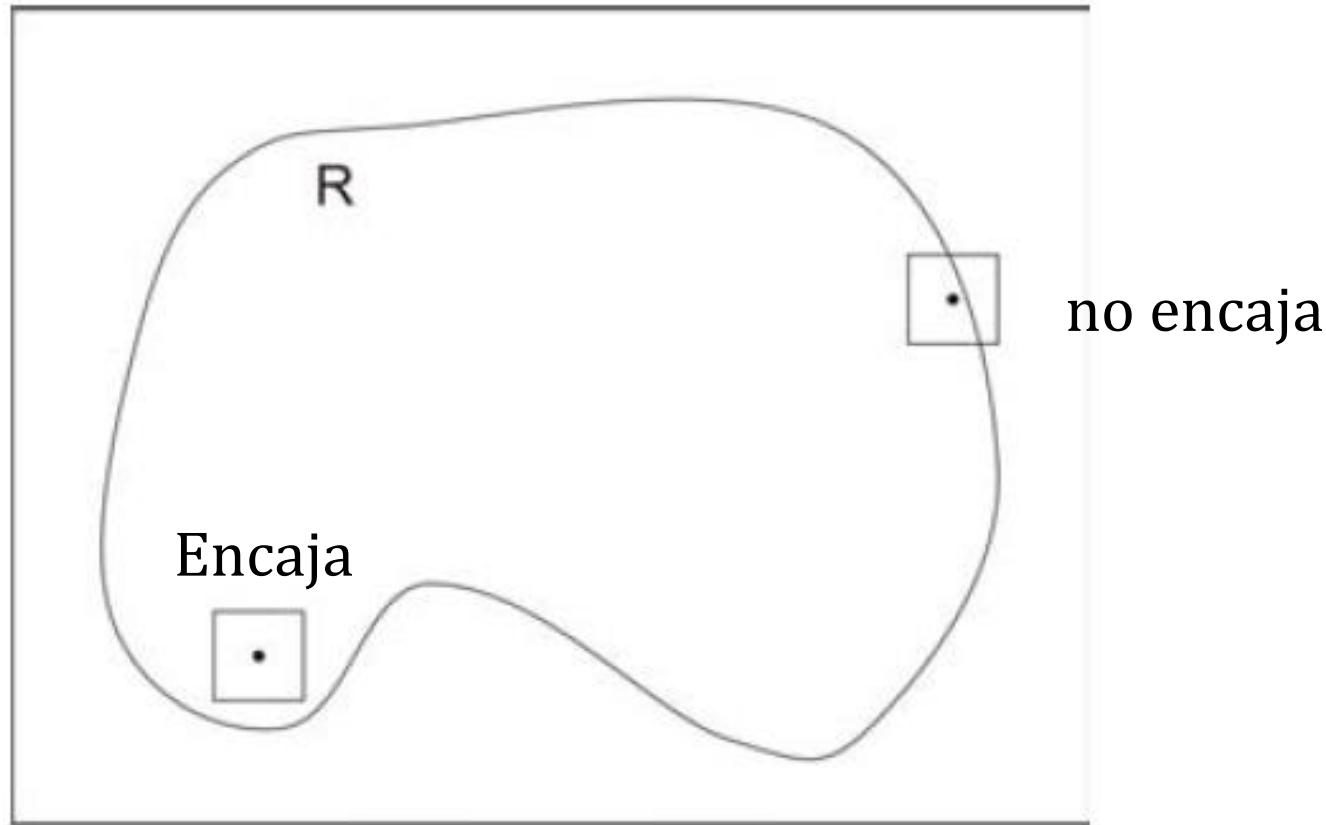
**CITOLOGÍA,**

**ANÁLISIS DE IMÁGENES,**

**ROBÓTICA,**

**ETC.**

# IDEA DE FUNCIONAMIENTO DE LA MM



La idea básica de la MM consiste en probar una imagen por medio de un probador llamado elemento de estructura (EE) y así cualificar la manera en que dicho EE encaja o no en dicha imagen.

# IDEA DE FUNCIONAMIENTO DE LA MM

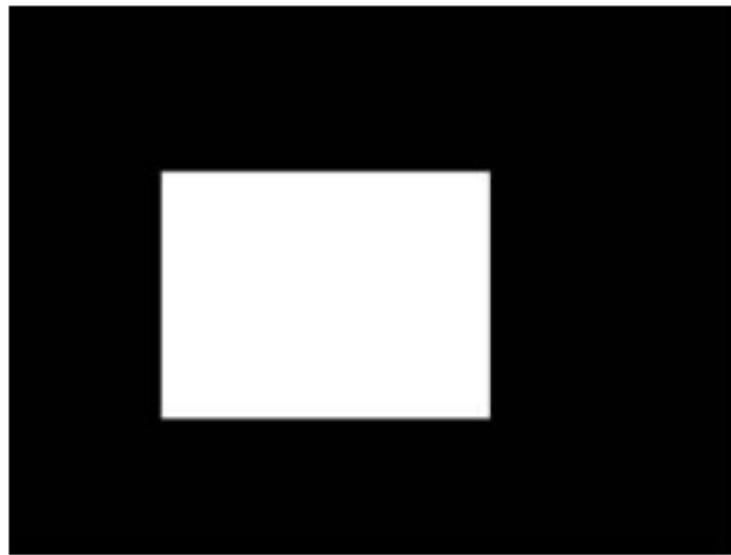
encaja, en la otra no. Al marcar las posiciones en las cuales el EE encaja en la imagen (objeto) uno es capaz de derivar información relacionada con la estructura relativa a la imagen y objeto. Esta información depende tanto del tamaño del EE como de su forma, y como fue enfatizado por Matheron, “*La naturaleza de la información depende, por tanto, de la selección del EE*”.

Todos los métodos morfológicos dependen del concepto de encajar el EE. Existen dos operaciones de base, **la erosión y la dilatación**. A partir de éstas se derivan las demás operaciones.

# DEFINICIONES

**Definición 3.1.** Sea  $A$  un conjunto en  $\mathbb{Z}^2$  con componente  $a = (a_1, a_2)$ , la traslación de  $A$  por  $x = (x_1, x_2) \in \mathbb{Z}^2$  denotada como  $(A_x)$ , se define como:

$$(A)_x = \{c \mid c = a + x, \forall a \in A\}. \quad (3.20)$$



(a)

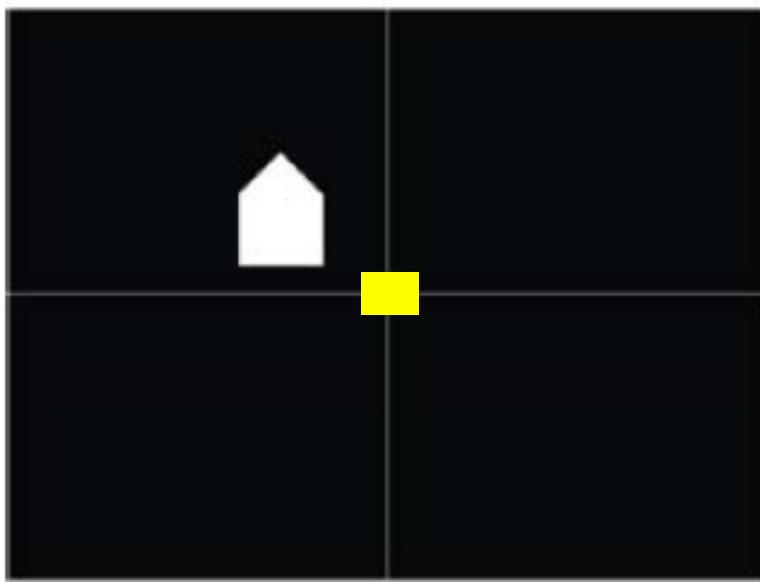


(b)

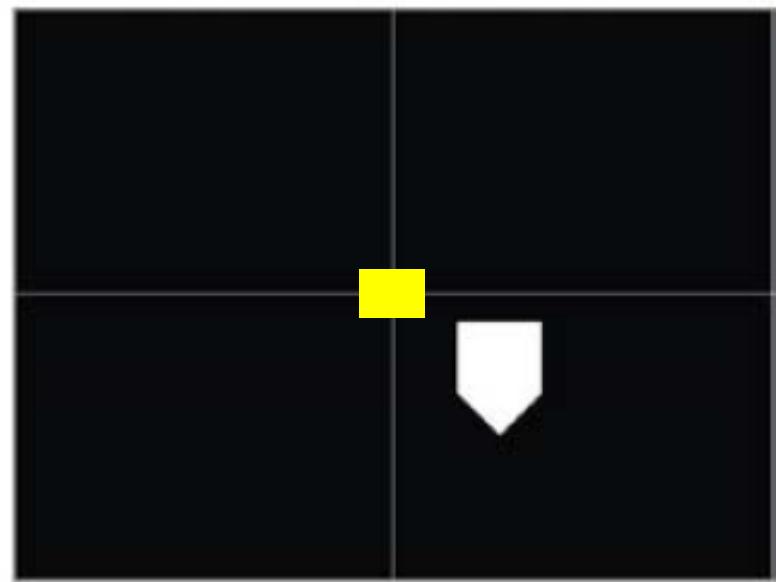
# DEFINICIONES

**Definición 3.2.** Sea  $B$  un conjunto en  $Z^2$  con componente  $b = (b_1, b_2)$ , la reflexión de  $B$ , denotada  $\hat{B}$ , se define como:

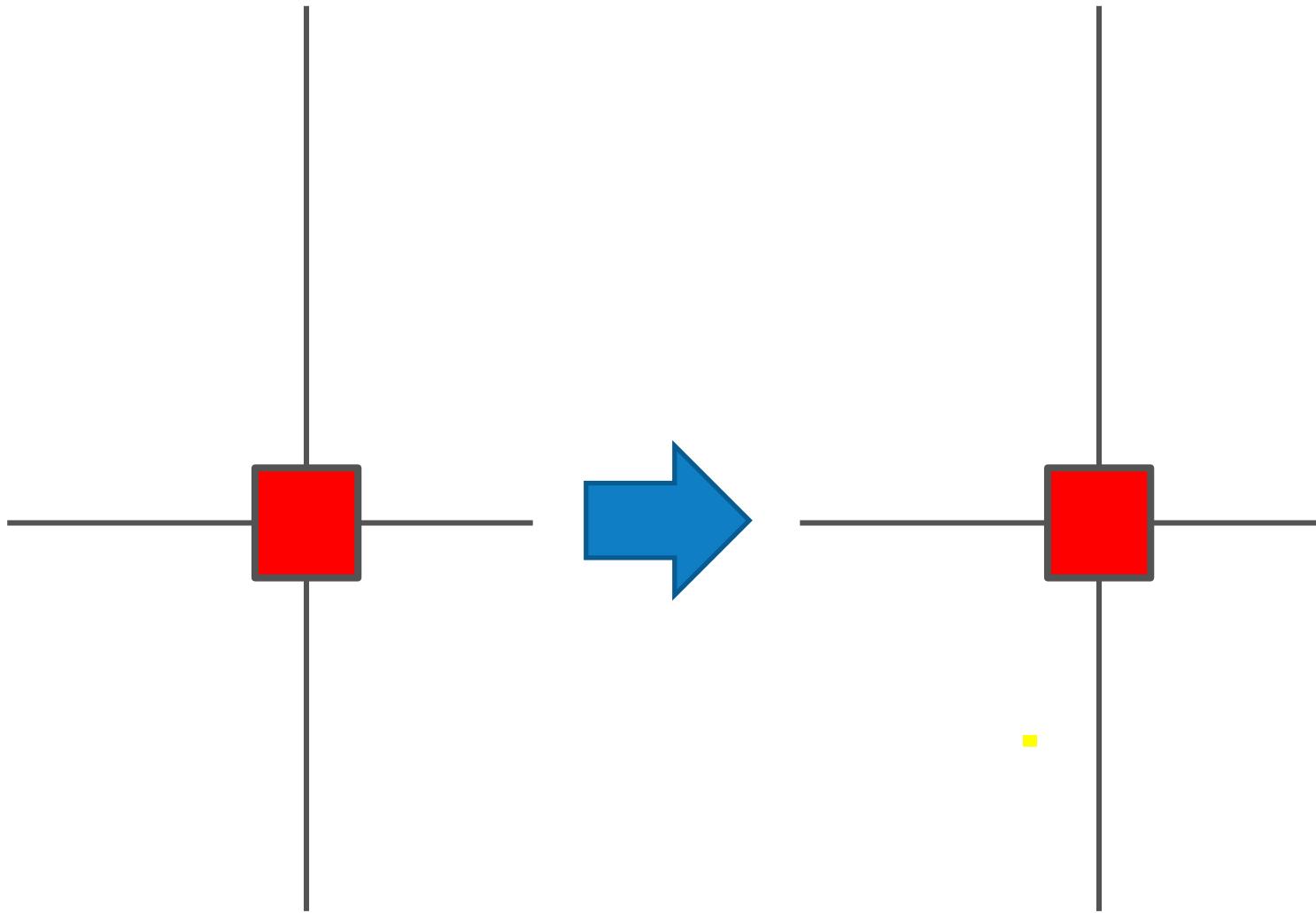
$$\hat{B} = \{b \mid b = -x, \forall b \in B\}. \quad (3.21)$$



(c)



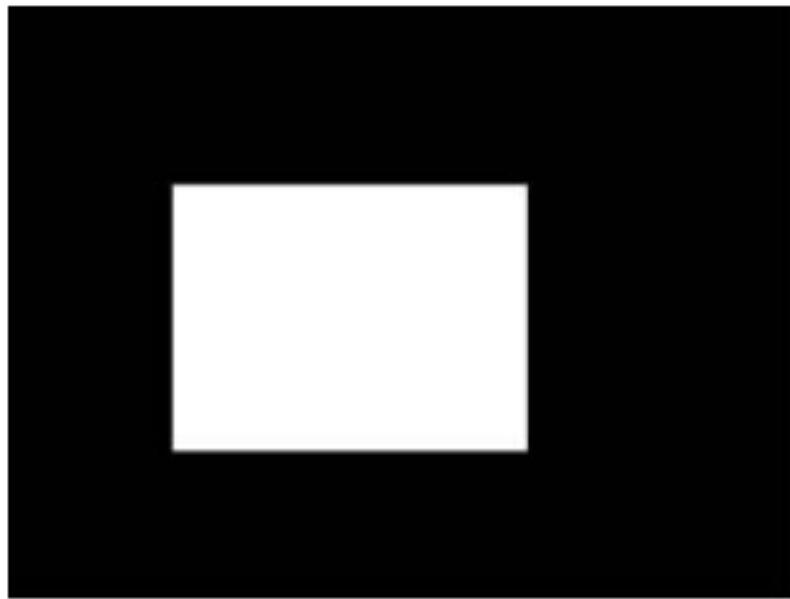
(d)



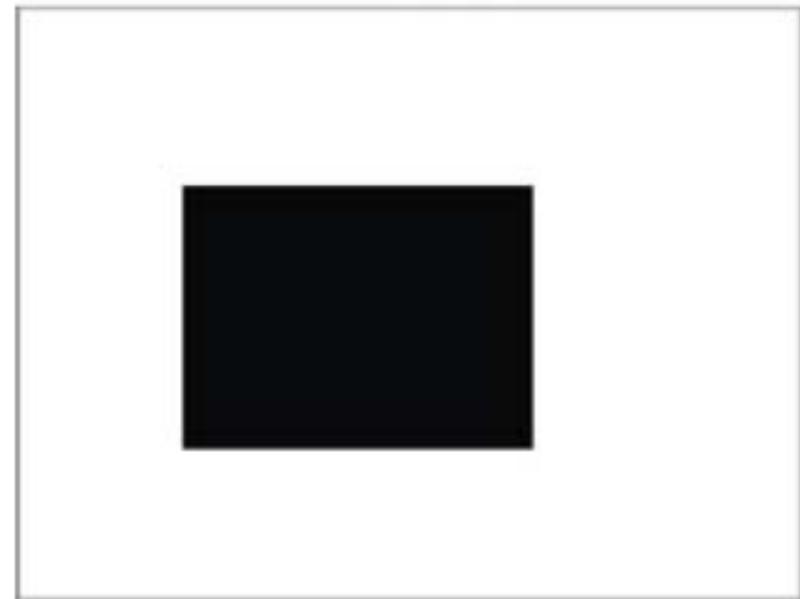
# DEFINICIONES

**Definición 3.3.** El complemento de  $A$ , denotado como  $A^c$ , se define como:

$$A^c = \{x \mid x \notin A\}. \quad (3.22)$$



(a)



(c)

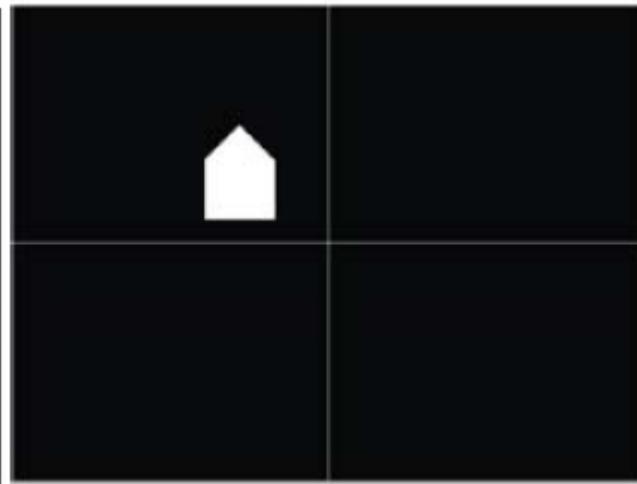
# DEFINICIONES

**Definición 3.4.** Sean  $A$  y  $B$  dos conjuntos en  $Z^2$ , la diferencia entre  $A$  y  $B$ , denotada como  $A-B$ , se define como:

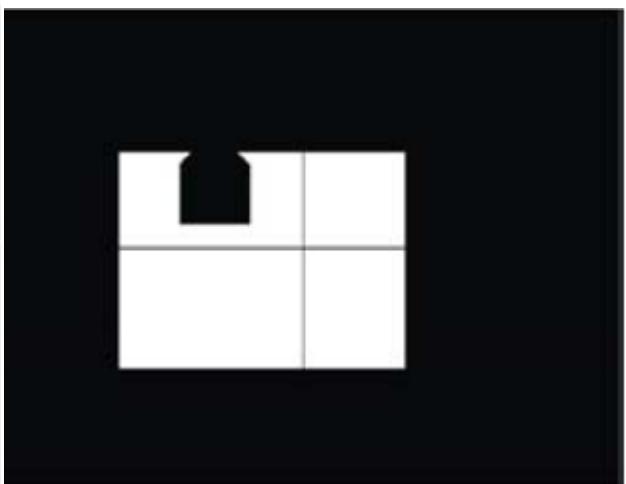
$$A - B = \{x \mid x \in A, x \notin B\}. \quad (3.23)$$



(a)



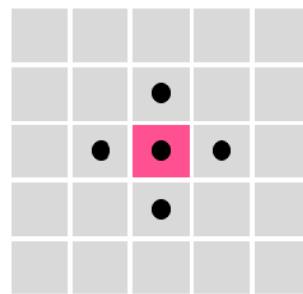
(c)



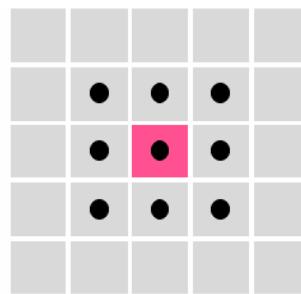
(f)

# El elemento estructural:

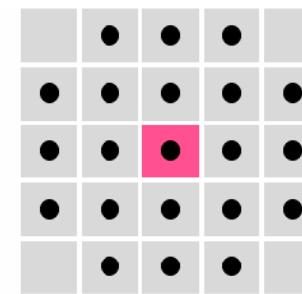
Al tamaño y la forma del vecindario utilizado en las operaciones de erosión y dilatación se conoce como elemento estructural. Así, la figura a representa un elemento estructural equivalente a considerar el vecindario de los 4 vecinos más próximos, la figura b el de los 8 vecinos más próximos y la figura c representa un vecindario aproximadamente circular.



(a)



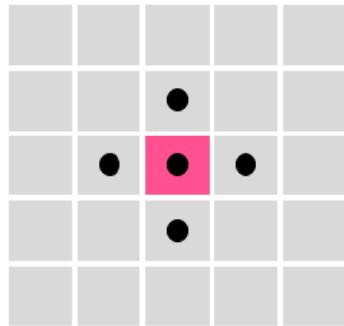
(b)



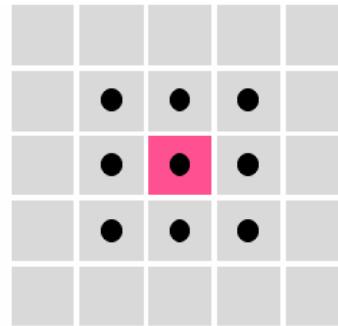
(c)

Ejemplos de elementos estructurales. Los puntos negros representan los píxeles del vecindario que constituyen el elemento estructural. Los píxeles en rojo representan el punto de prueba.

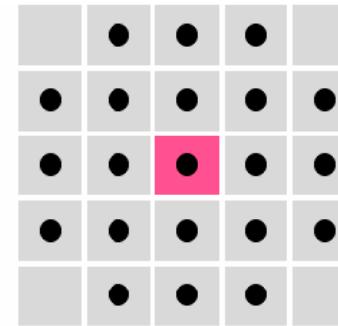
Cada elemento estructural tiene un punto de **prueba** (**inserción**), es decir, la posición del vecindario que se hace corresponder con el píxel de la imagen al aplicar el filtro (píxel en **rojo** en la figura).



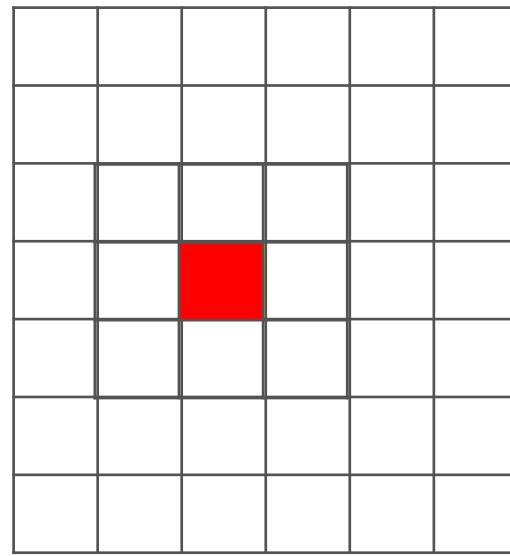
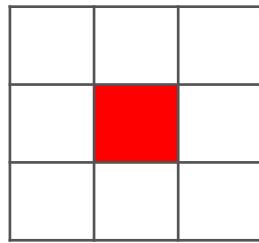
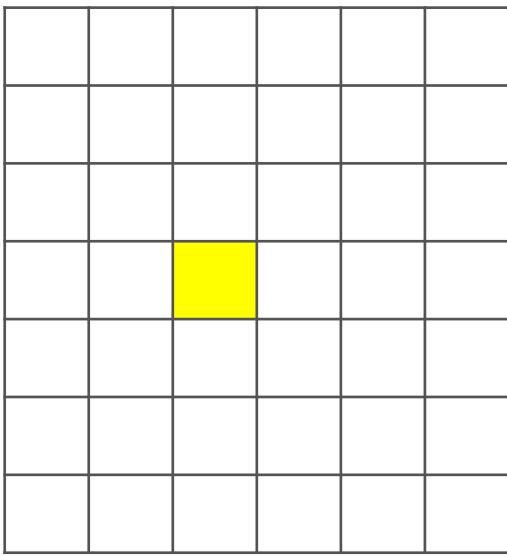
(a)



(b)



(c)

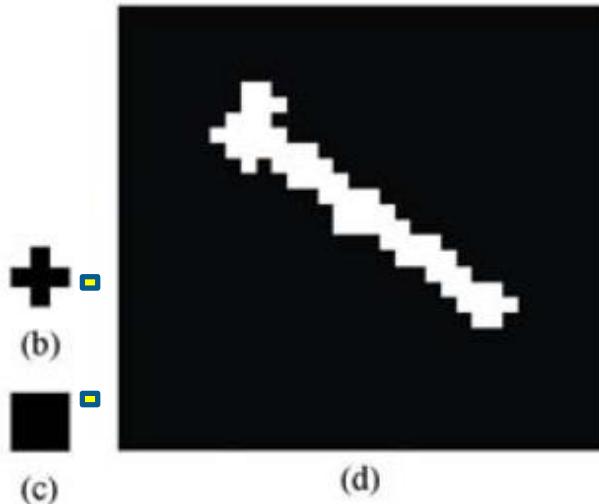


# OPERACIONES BÁSICAS: EROSIÓN

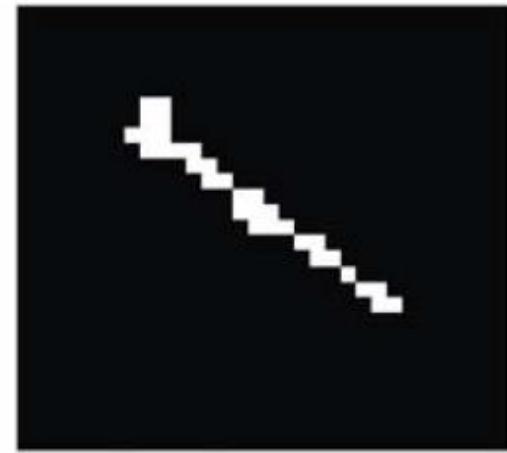
**Definición 3.5.** Sean  $A$  y  $B$  dos conjuntos en  $\mathbb{Z}^2$ , la erosión de  $A$  por  $B$ , denotada como  $A \Theta B$  se define como:

$$A \Theta B = \{x \mid (B)_x \subseteq A\}. \quad (3.24)$$

En otras palabras, la erosión de  $A$  por  $B$  es el conjunto de todos los puntos  $x$  tal que  $B$  trasladada por  $x$  está contenida en  $A$ .



(d)



(e)

## Ejemplo (erosión):

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1  
1 1 1  
1

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1  
1 1 1  
1 1 1

## Ejemplo (erosión):

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1  
1 1 1  
1

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1  
1 1 1  
1 1 1

## Ejemplo (erosión):

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1  
1 1 1  
1  
1

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1  
1 1 1  
1 1 1

## Ejemplo (erosión):

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1  
1 1 1  
1  
1 1

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1  
1 1 1  
1 1 1

## Ejemplo (erosión):

1 1 1 1 1

1 1 1

1 1 1 1 1

1 1 1 1

1

1 1 1

1

1 1 1 1 1

1 1 1

1 1 1

1 1 1

1 1 1 1 1

1 1 1

1 1 1 1

## Ejemplo (erosión):

1 1 1 1 1

1 1 1

1 1 1 1 1

1 1 1 1

1  
1 1 1

1

1 1 1 1 1

1 1 1

1 1 1 1 1

1 1 1 1

1 1 1

1 1 1

1

1 1 1

## Ejemplo (erosión):

1 1 1 1 1

1 1 1

1 1 1 1 1

1 1 1 1

1

1 1 1

1

1 1 1 1 1

1 1 1

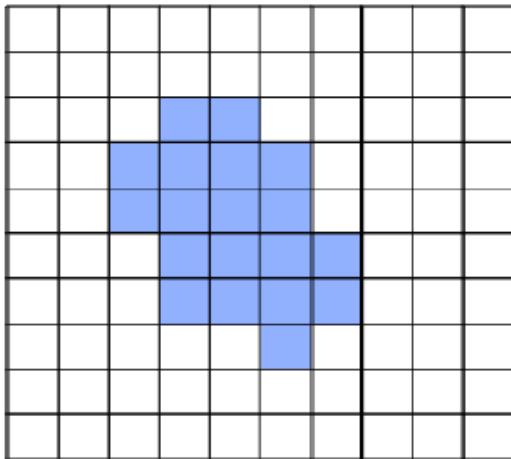
1 1 1

1

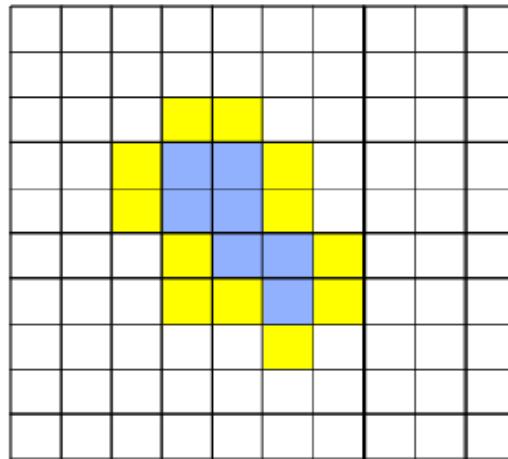
1 1 1 1 1

1

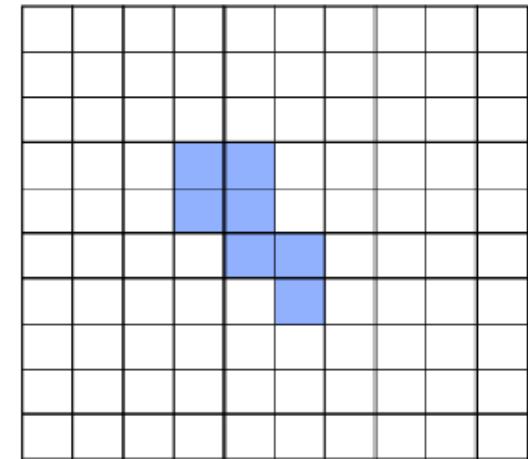
1 1 1 1



(a)

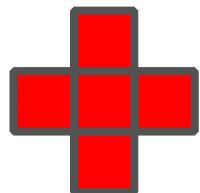


(b)



(c)

Erosión morfológica: (a) objeto original (en azul), (b) píxeles eliminados en la operación de erosión (en amarillo), y (c) resultado final.



# OPERACIONES BÁSICAS: DILATACIÓN

**Definición 3.6.** Sean  $A$  y  $B$  dos conjuntos en  $\mathbb{Z}^2$ , la dilatación de  $A$  por  $B$ , denotada como  $A \oplus B$  se define como:

$$A \oplus B = \{x \mid \hat{B}_x \cap A \neq \emptyset\}. \quad (3.25)$$

En otras palabras, el proceso de dilatación consiste en obtener la reflexión del elemento estructurante  $B$  alrededor de su origen y después desplazar esta reflexión por  $x$ .  $A \oplus B$  es entonces el conjunto de todos los desplazamientos en  $x$  tales que  $\hat{B}$  y  $A$  compartan al menos un elemento no igual a cero.

# OPERACIONES BÁSICAS: DILATACIÓN

**Definición 3.6.** Sean  $A$  y  $B$  dos conjuntos en  $\mathbb{Z}^2$ , la dilatación de  $A$  por  $B$ , denotada como  $A \oplus B$  se define como:

$$A \oplus B = \{x \mid \hat{B}_x \cap A \neq \emptyset\}. \quad (3.25)$$



(a)



(b)



(d)



(e)

## Ejemplo (dilatación):

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1  
1 1 1  
1

1 1 1 1 1  
1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1  
1 1 1  
1 1 1

## Ejemplo (dilatación):

1 1 1 1 1

1 1 1

1 1 1 1 1

1 1 1 1

1  
1 1 1

1

1 1 1 1 1  
1 1 1 1 1 1 1  
1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1  
1 1 1 1

1 1 1 1 1

1 1 1

1 1 1 1 1

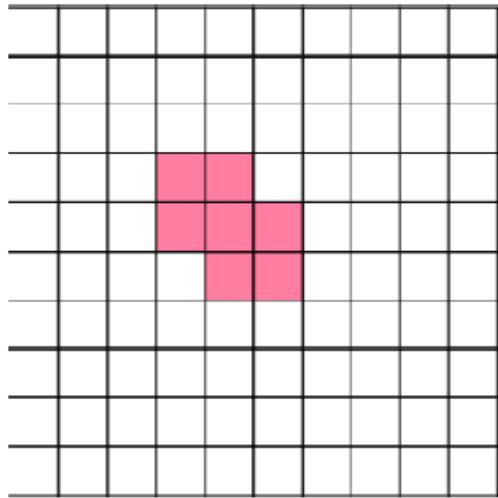
1 1 1 1

1 1 1

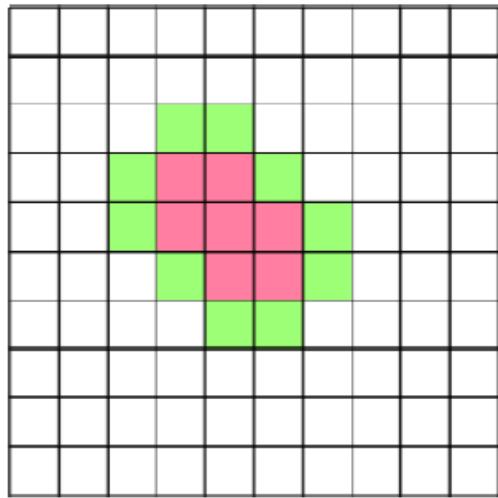
1 1 1

1 1 1

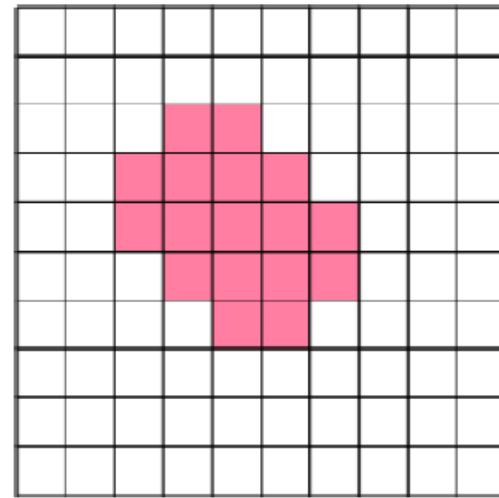
1 1 1 1 1 1  
1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1  
1 1 1 1 1



(a)

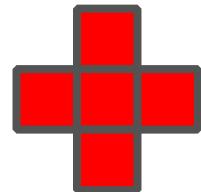


(b)



(c)

Dilatación morfológica: (a) objeto original (en rojo), (b) píxeles convertidos en objeto al aplicar la dilatación (en verde), y (c) resultado final.



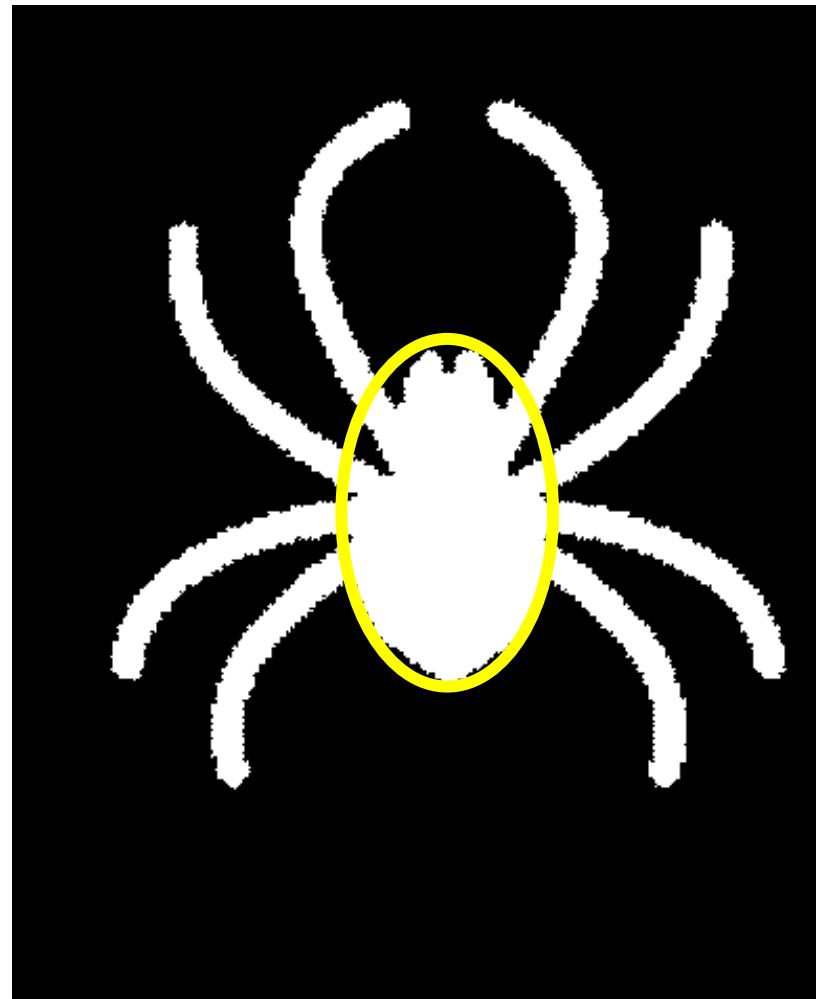
# Efecto de encadenar filtros:

Con objeto de aumentar su efecto, se pueden diseñar filtros consistentes en la **iteración** de varias operaciones de erosión o de dilatación.

En ocasiones no es suficiente la aplicación de una erosión simple para eliminar salientes del contorno de los objetos, o de una dilatación para eliminar agujeros o huecos en un objeto de un cierto tamaño.

En estas ocasiones es necesario el encadenamiento de  $n$  iteraciones de filtrado para obtener el resultado buscado.

**Ejemplo:** Eliminar de forma automática las extremidades de la araña de la figura (a) y dejar únicamente con el cuerpo:

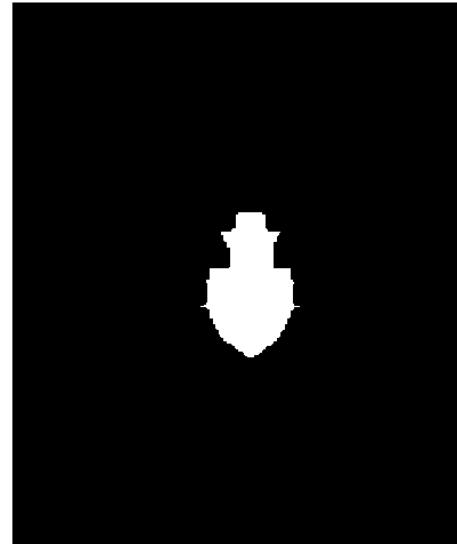
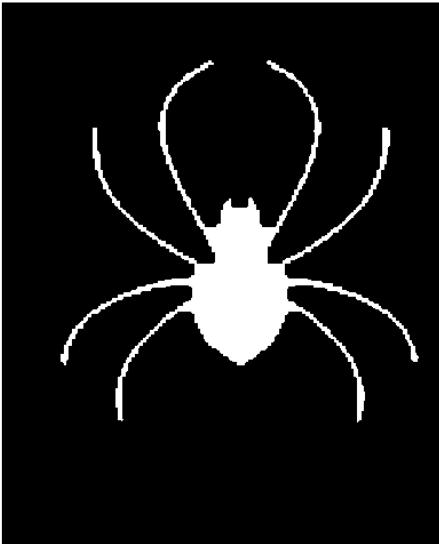
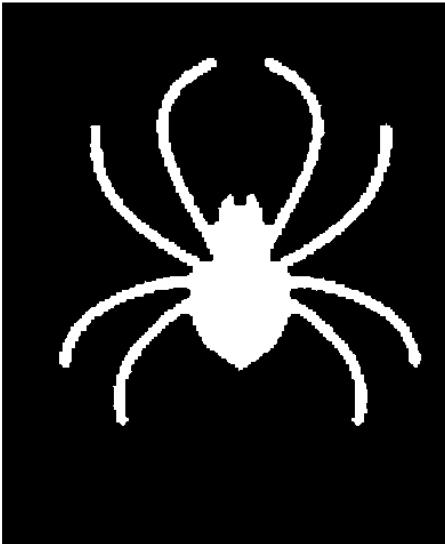


(a)

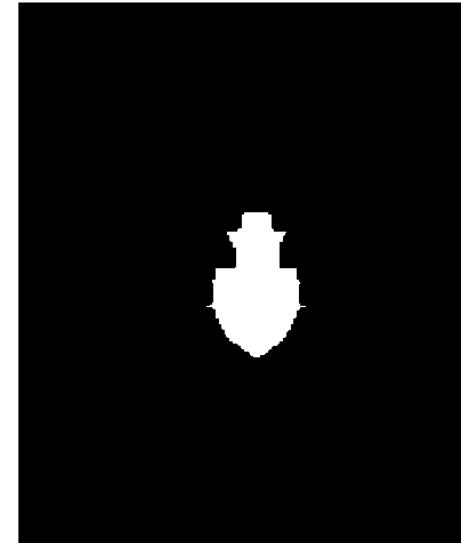
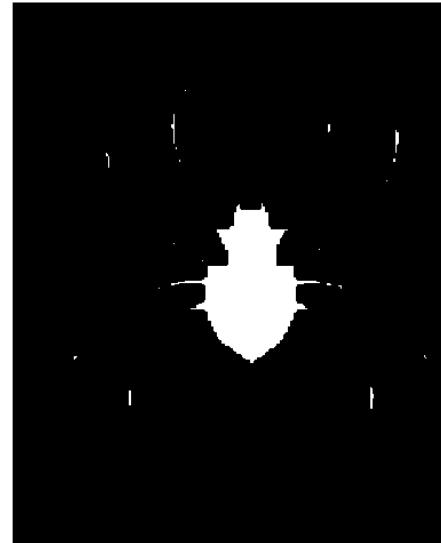
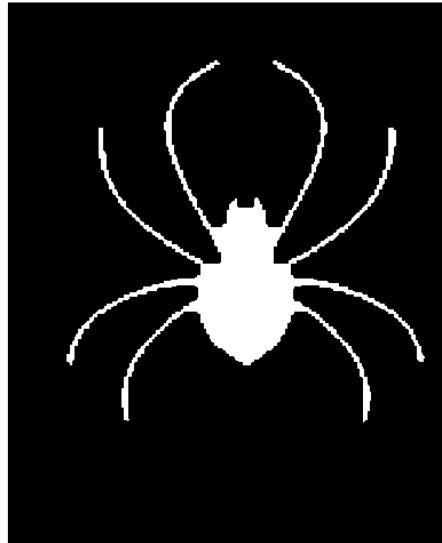
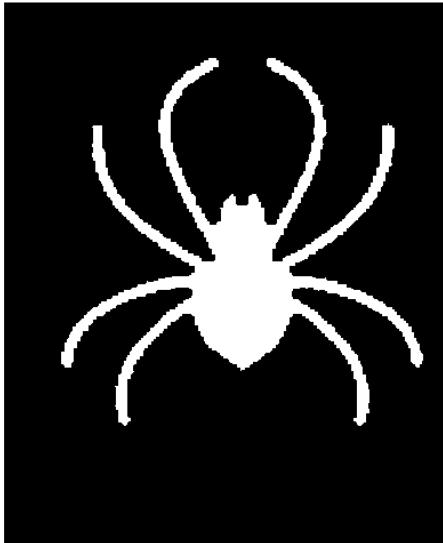
Para esto, se debe determinar **el número de iteraciones de erosión necesarias** hasta que las extremidades hayan sido eliminadas

Después realizar el mismo número **de iteraciones de dilatación** para recuperar el tamaño del cuerpo.

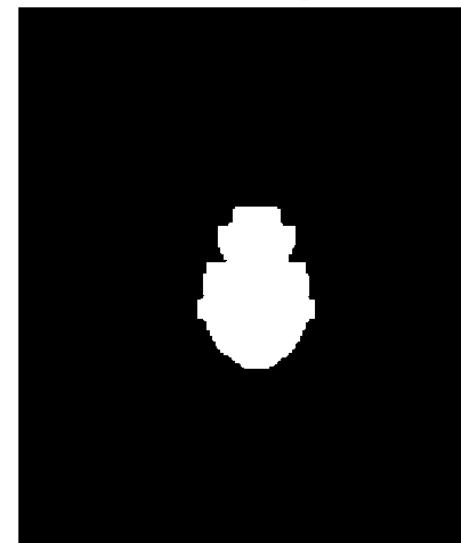
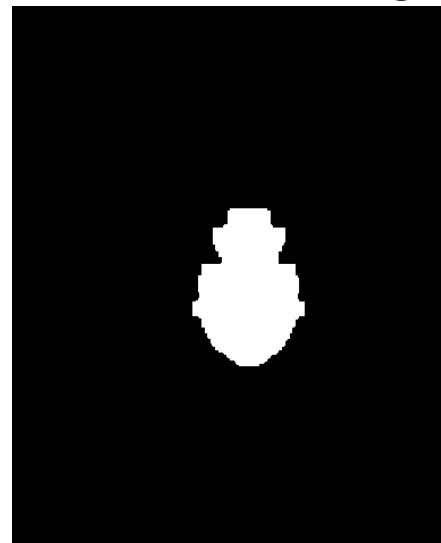
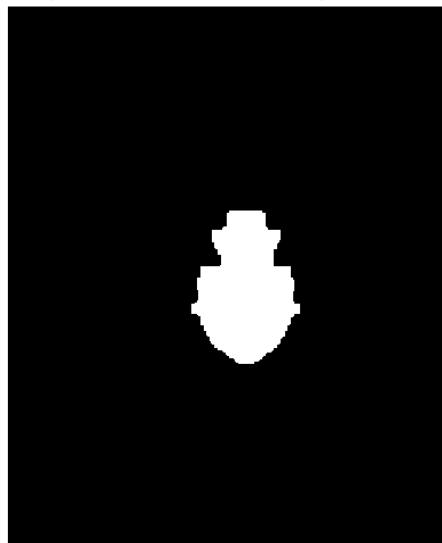
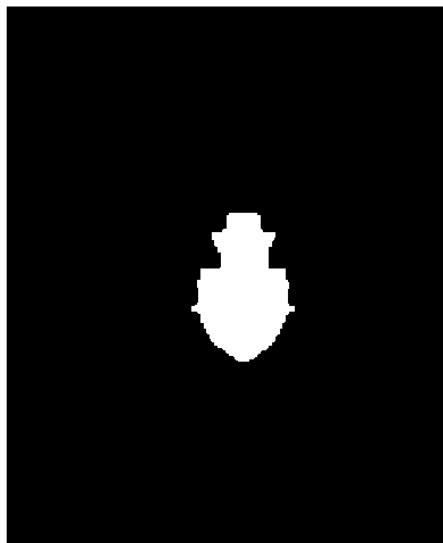
## 4 erosiones encadenadas con un vecindario de $3 \times 3$ :



**4 erosiones encadenadas con un vecindario de  $3 \times 3$ :**



**4 dilataciones para recuperar el tamaño original del cuerpo:**



% Efecto de la aplicación iterada de erosiones  
y dilataciones:

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

f=imread('aragna 01.BMP');
figure(1)
imshow(f)
title('Imagen original')
```

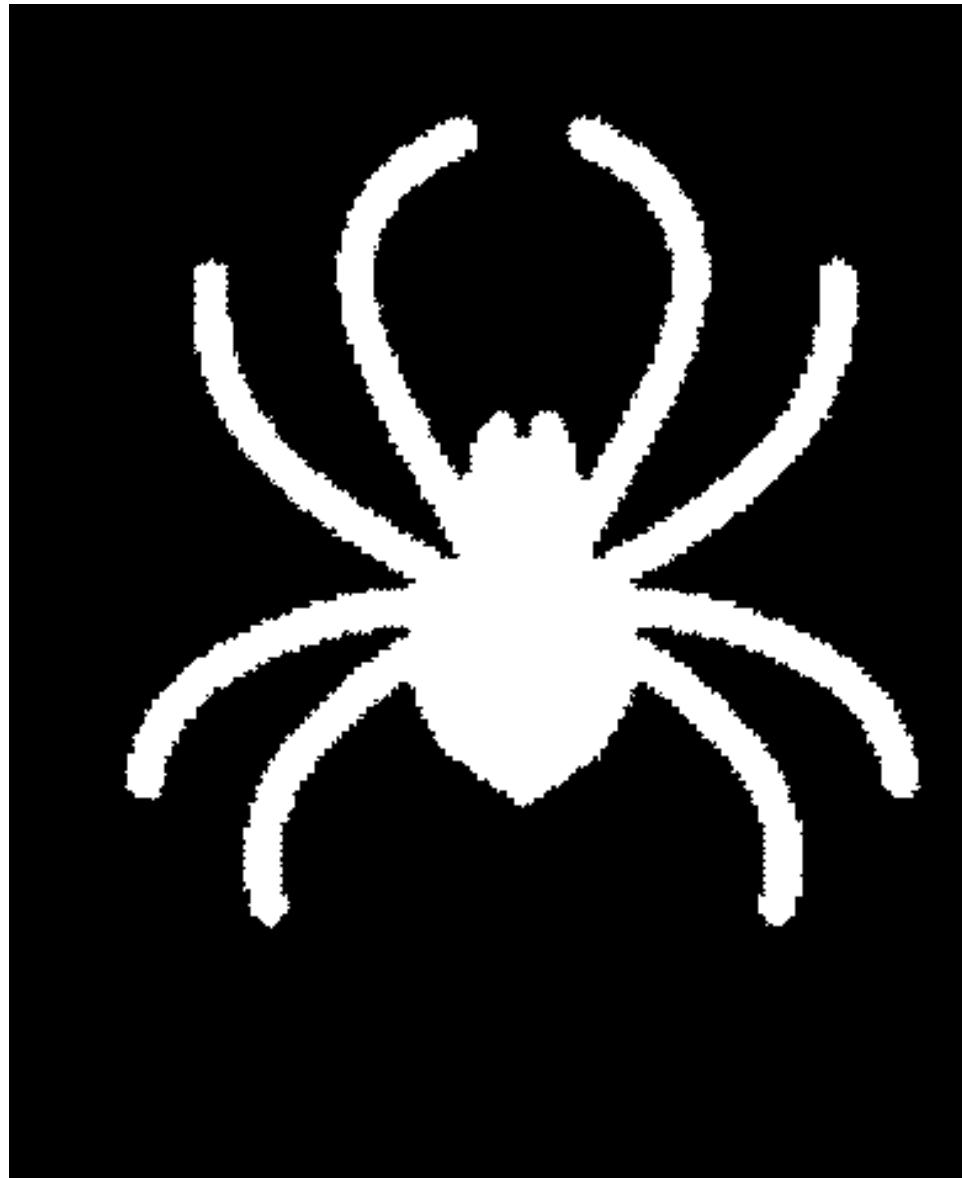
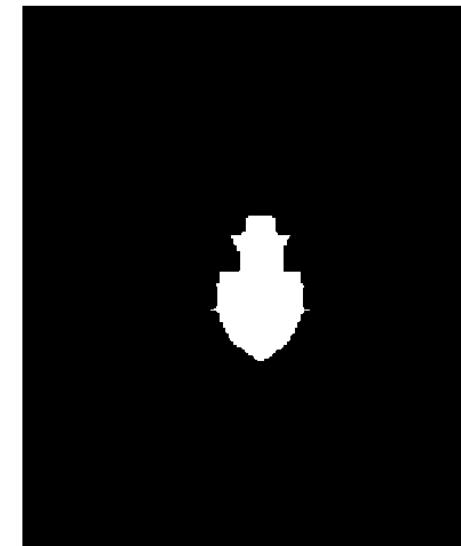
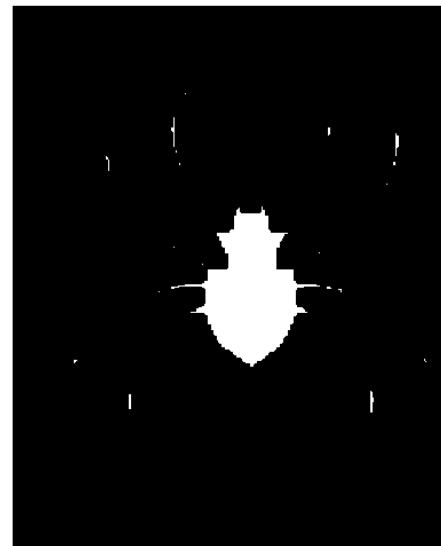
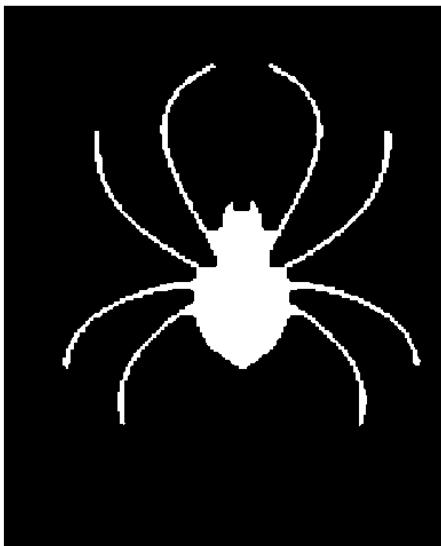
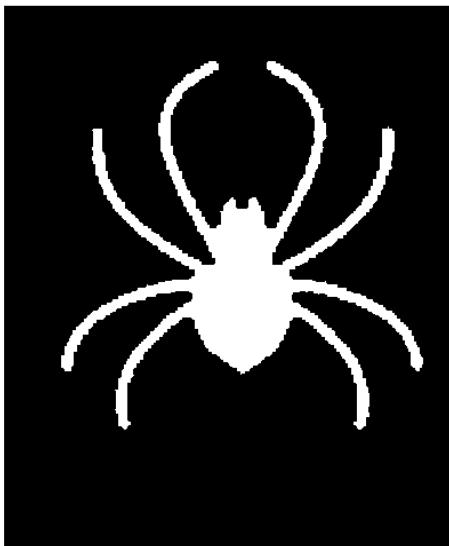


Imagen original

```
se = strel('disk',3); % square 3x3, line lenght y angulo,  
disk  
  
ge1 = imerode(f,se); % erosion de imagen  
figure(2)  
imshow(ge1)  
title('Imagen erosionada una vez')  
  
ge2 = imerode(ge1,se); % erosion de imagen  
figure(3)  
imshow(ge2)  
title('Imagen erosionada dos veces')  
  
ge3 = imerode(ge2,se); % erosion de imagen  
figure(4)  
imshow(ge3)  
title('Imagen erosionada tres veces')  
  
ge4 = imerode(ge3,se); % erosion de imagen  
figure(5)  
imshow(ge4)  
title('Imagen erosionada cuatro veces')
```

**4 erosiones encadenadas con un vecindario de  $3 \times 3$ :**



```
gd1 = imdilate(ge4,se); % dilatacion de imagen
figure(6)
imshow(gd1)
title('Imagen dilatada una vez')

gd2 = imdilate(gd1,se); % dilatacion de imagen
figure(7)
imshow(gd2)
title('Imagen dilatada dos veces')

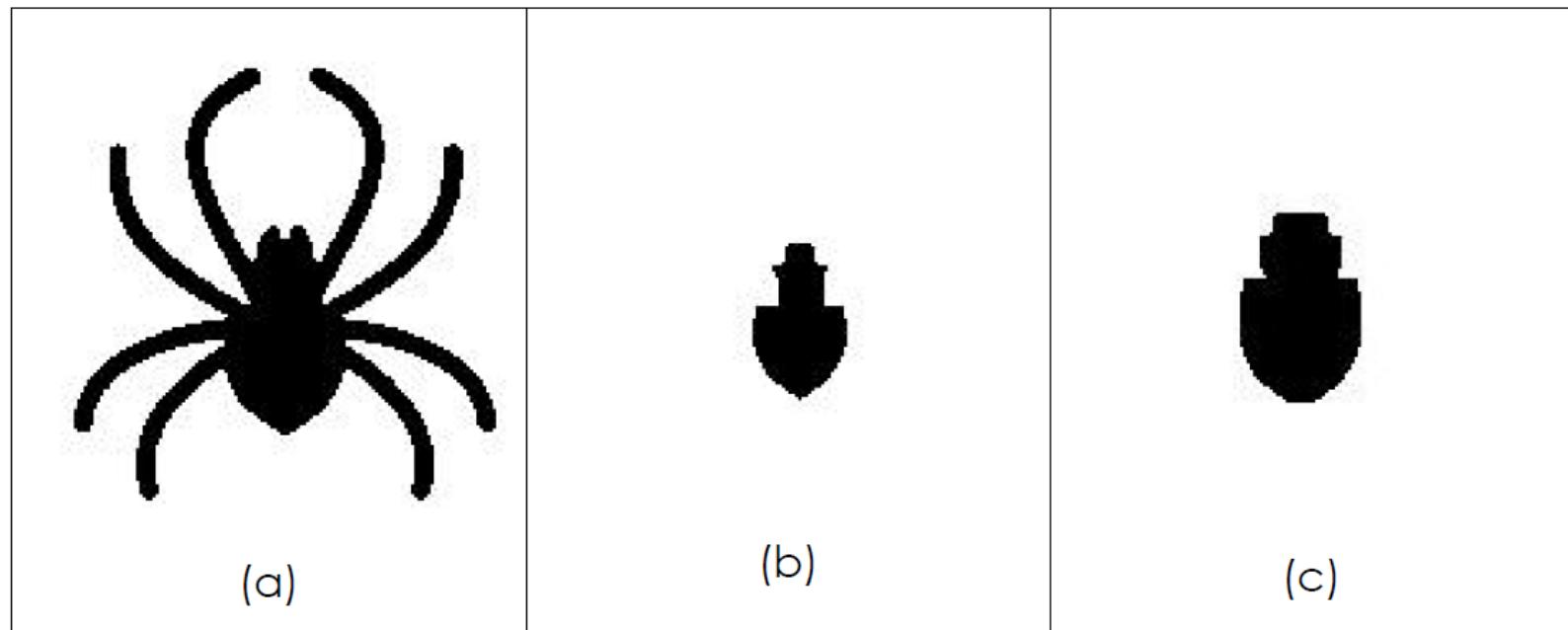
gd3 = imdilate(gd2,se); % dilatacion de imagen
figure(8)
imshow(gd3)
title('Imagen dilatada tres veces')

gd4 = imdilate(gd3,se); % dilatacion de imagen
figure(9)
imshow(gd4)
title('Imagen dilatada cuatro veces')
```

**4 dilataciones para recuperar el tamaño original del cuerpo:**



En este caso (figura b) se han aplicado **4 erosiones encadenadas con un vecindario de los 8 vecinos más próximos** hasta eliminar completamente las extremidades de la araña, seguidas de **4 dilataciones** para recuperar el tamaño original del cuerpo.



Encadenamiento de filtros de **erosión y dilatación**: a) figura original; b) resultado de aplicar **4 erosiones** encadenadas hasta eliminar las extremidades; c) resultado de aplicar **4 dilataciones** hasta recuperar el tamaño del cuerpo original

Uno pudiera pensar que las operaciones de erosión y dilatación son una inversa de la otra.

Esto, en general, no es cierto.

Esto es lo que le da el poder a las operaciones morfológicas para hacer procesamiento no lineal de señales.

Se dice que dos operaciones son una la dual de la otra cuando que?...

Esto se responde enseguida:

# APERTURA MORFOLÓGICA

**Definición 3.7.** Sean  $A$  y  $B$  dos conjuntos en  $Z^2$ , la apertura de  $A$  por  $B$ , denotada como  $A \circ B$ , se define como:

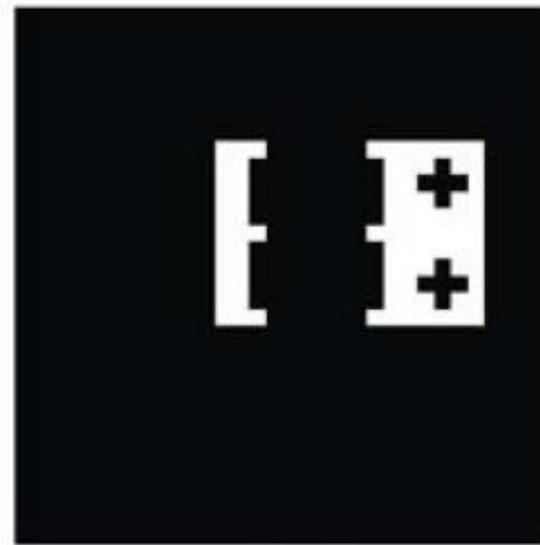
$$A \circ B = (A \ominus B) \oplus B.$$

En otras palabras, la apertura de  $A$  por  $B$  es simple y sencillamente la erosión de  $A$  por  $B$  seguida de la dilatación del resultado.



(a) Imagen A

Elemento de estructura  
 $B$



(b)

(c)  $A \ominus B$



(d)  $A \oplus B$

# CERRADURA MORFOLÓGICA

**Definición 3.8.** Sean  $A$  y  $B$  dos conjuntos en  $Z^2$ , la cerradura de  $A$  por  $B$ , denotada como  $A \bullet B$ , se define como:

$$A \bullet B = (A \oplus B) \ominus B .$$

En otras palabras, la cerradura de  $A$  por  $B$  es simple y sencillamente la dilatación de  $A$  por  $B$  seguida de la erosión del resultado.



(a) Imagen A

+  
Elemento  
de estructura  
 $B$



(b)

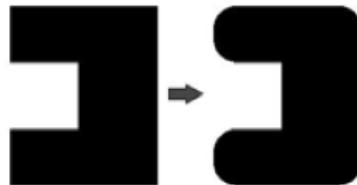
(c)  $A \oplus B$



(d)  $A \ominus B$

# OPERACIONES COMBINADAS: APERTURA Y CERRADURA

Apertura



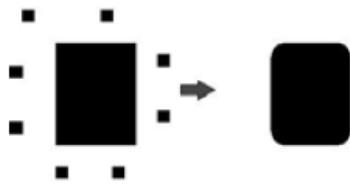
(a)



(b)

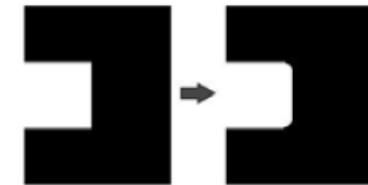


(c)

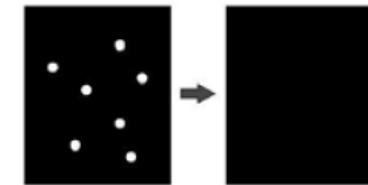


(d)

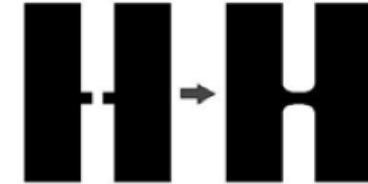
Cerradura



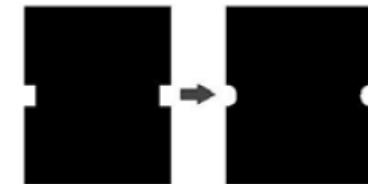
(e)



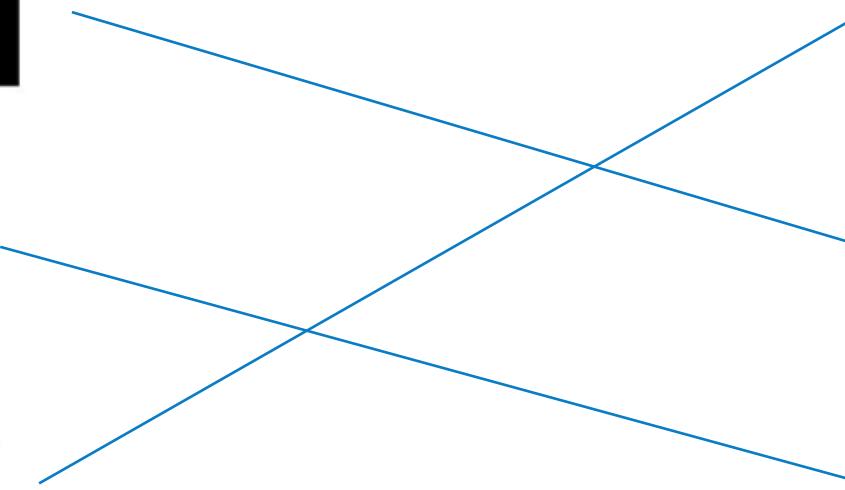
(f)



(g)



(h)

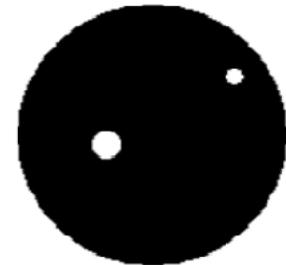


**Apertura**

$$I \circ H = (I \ominus H) \oplus H$$

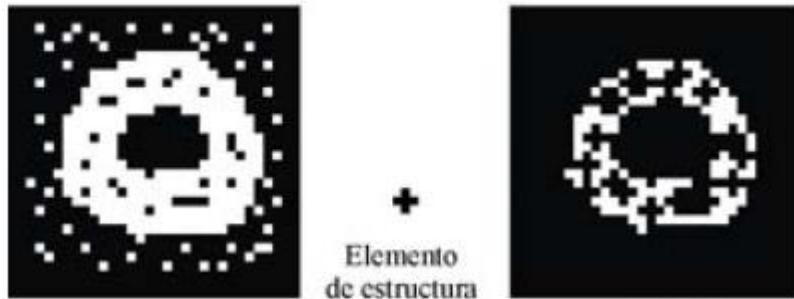
**Cierre**

$$I \bullet H = (I \oplus H) \ominus H$$

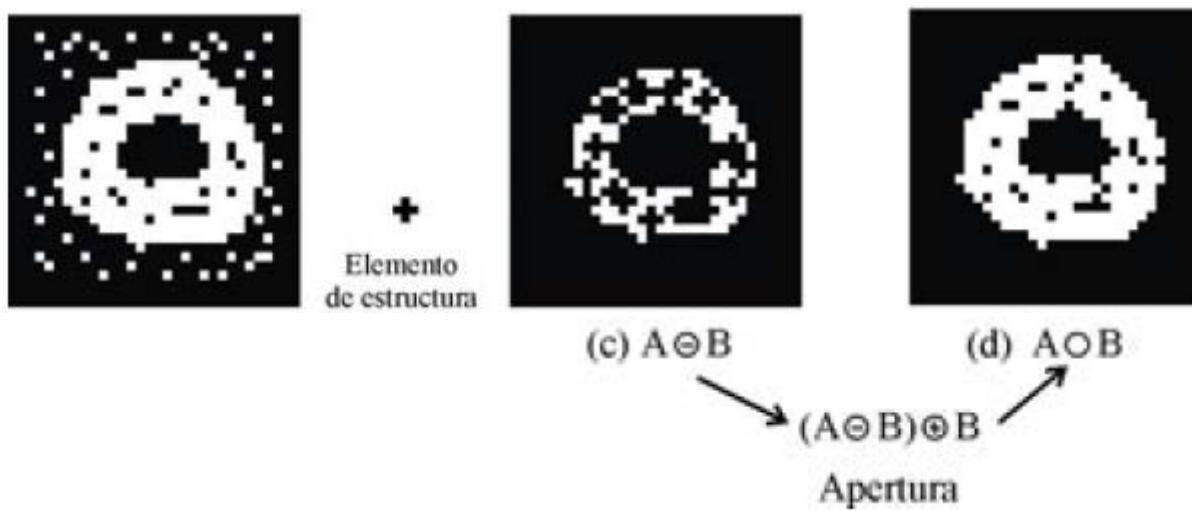


Ejemplos de operaciones de apertura (izquierda) y cierre (derecha).

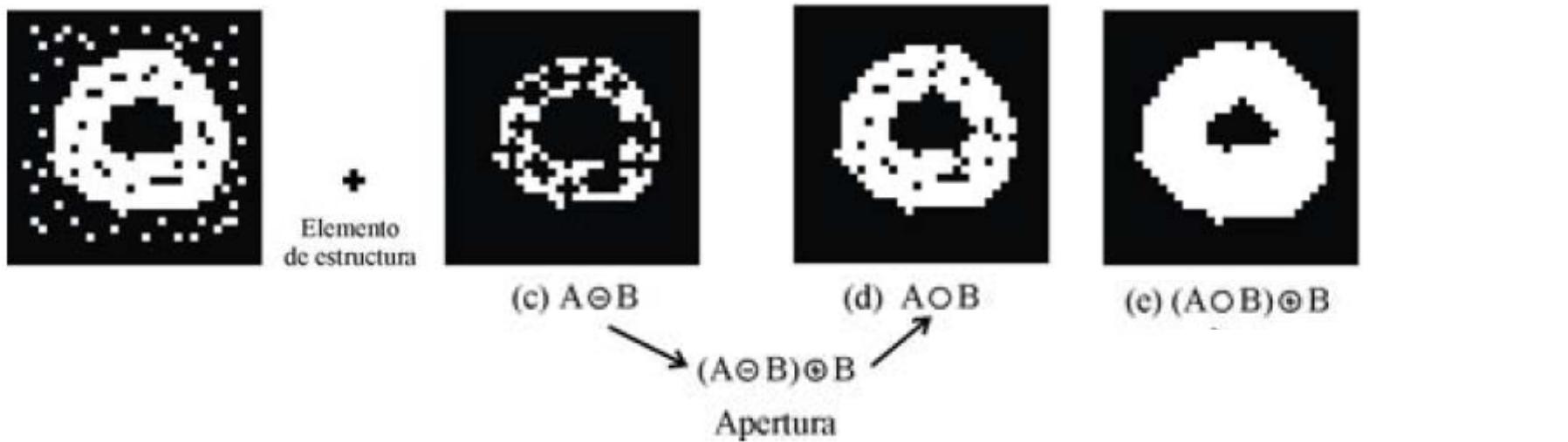
# FILTRADO MORFOLÓGICO



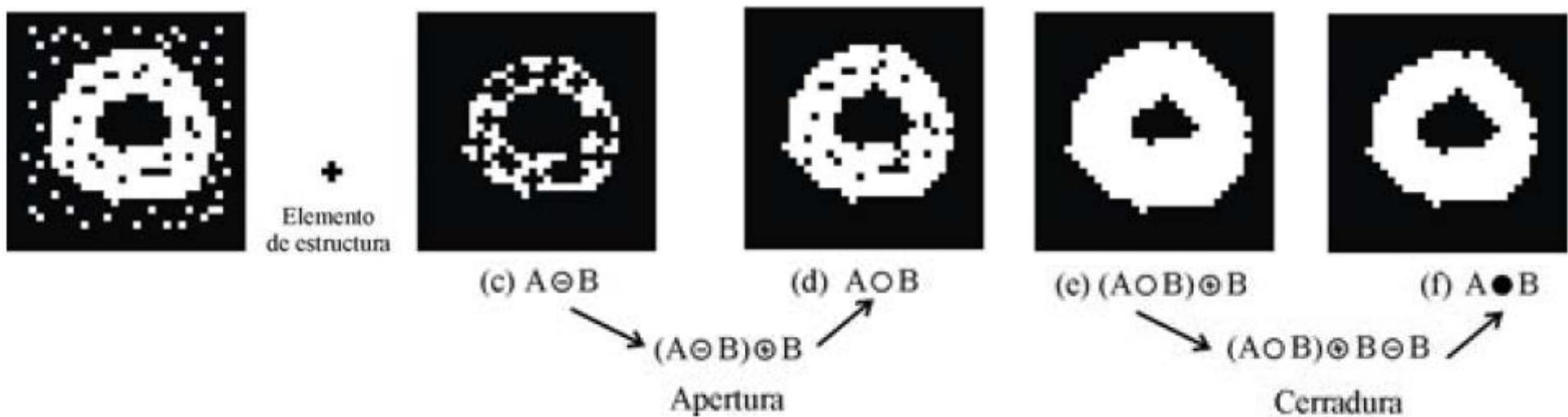
# FILTRADO MORFOLÓGICO



# FILTRADO MORFOLÓGICO



# FILTRADO MORFOLÓGICO



% Filtrado morfológico de imágenes.

```
clc; % limpia pantalla  
clearvars; %remueve todas las variables.  
close all; % cierra gráficas, archivos y recursos abiertos.  
  
f=imread('6 objetos.JPG');  
figure(1)  
imshow(f)  
title('Imagen original')  
  
fsp = imnoise(f, 'salt & pepper', 0.05);  
figure(2)  
imshow(fsp) % imagen con ruido.  
title('Imagen con ruido sal y pimienta')  
  
[n,m]=size(f);
```

# Ejemplo:

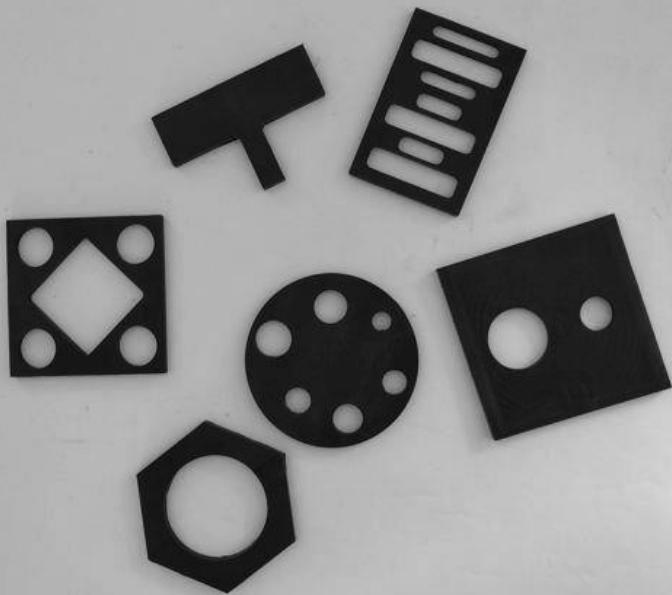


Imagen original

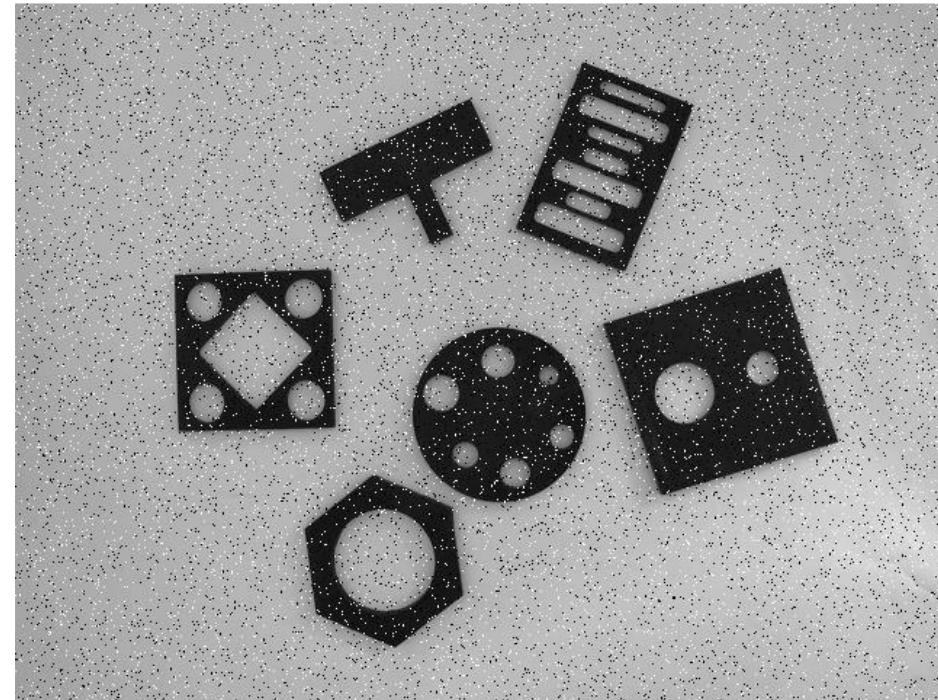


Imagen con ruido

```
th=90; % umbral

for y=1:m
    for x=1:n
        if fsp(x,y) >= th
            g(x,y)=0;
        else
            g(x,y)=255;
        end
    end
end

figure(3)
imshow(g)
title('Imagen binaria')
```

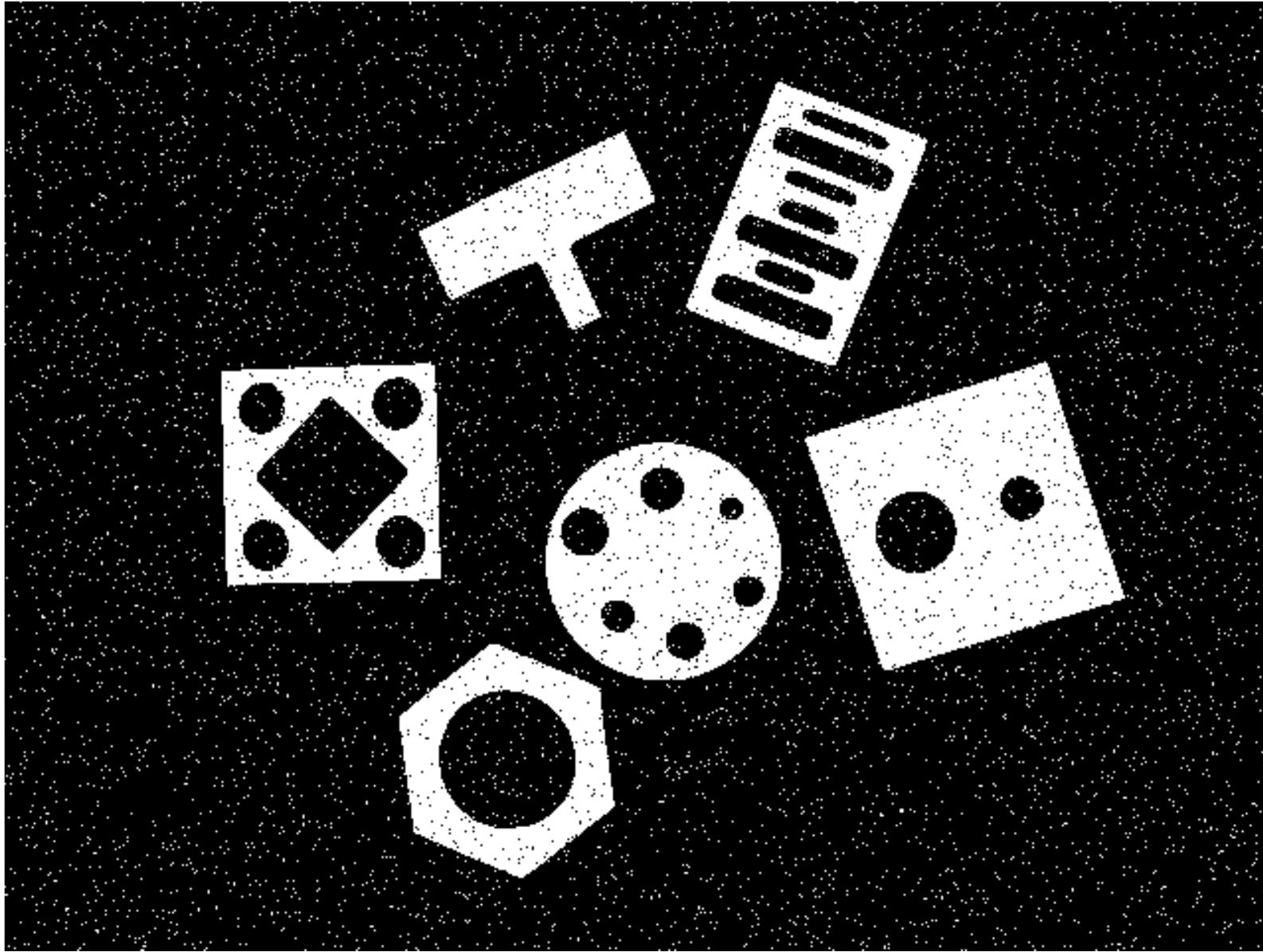


Imagen umbralada

% apertura:

```
se = strel('disk',1); % square 7x7, line lenght y angulo,  
disk  
gea = imerode(g,se); % erosion de imagen  
figure(4)  
imshow(gea)  
title('Imagen erosionada')  
  
gda = imdilate(gea,se); % dilatación de imagen  
figure(5)  
imshow(gda)  
title('Imagen dilatada')
```

## apertura

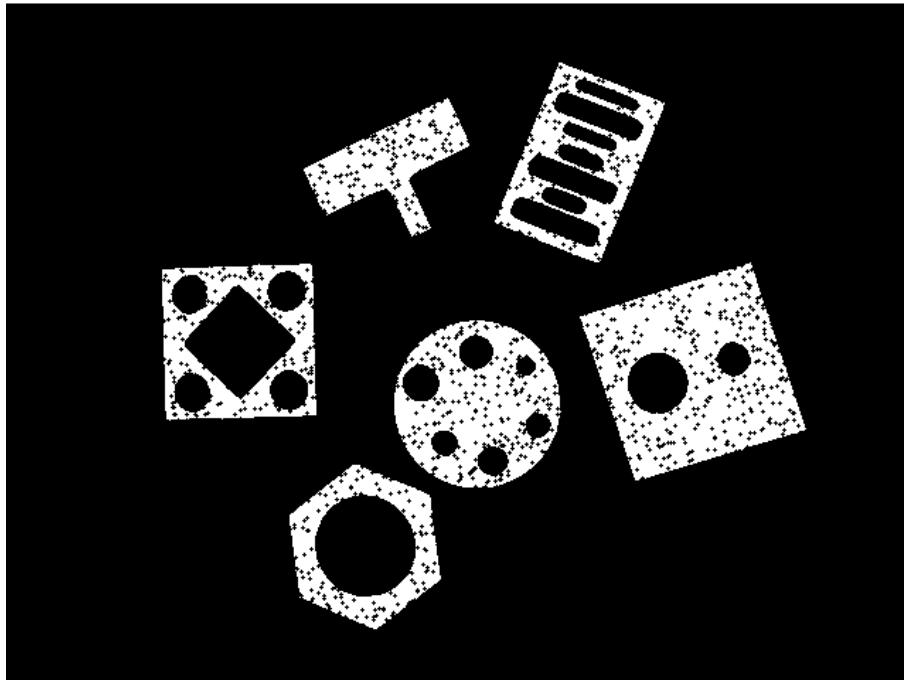


Imagen erosionada

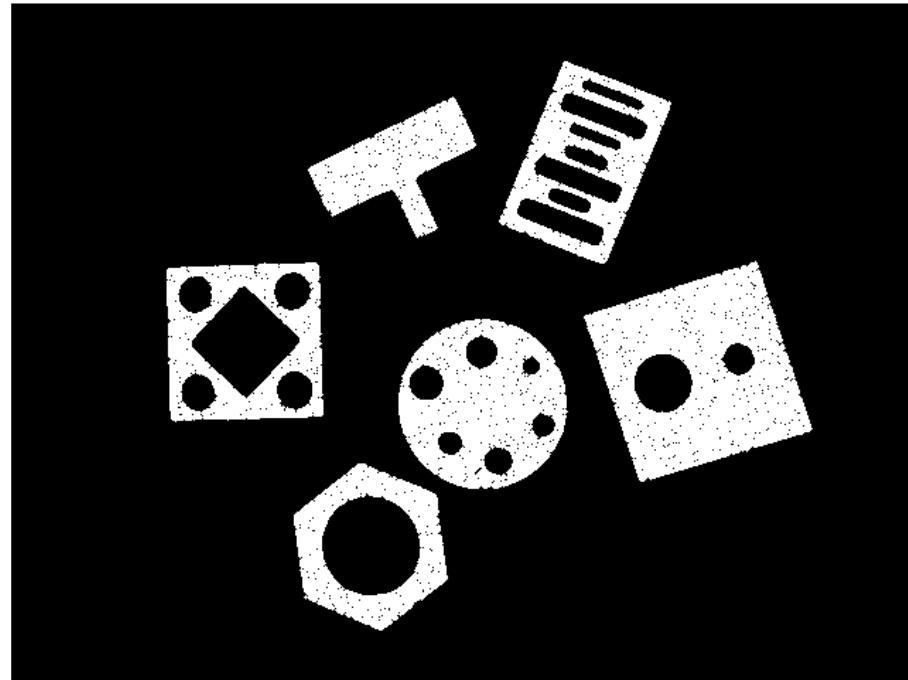


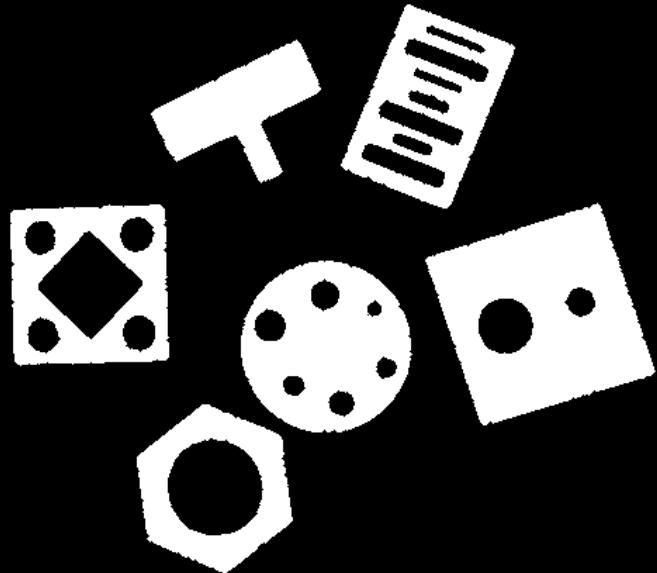
Imagen dilatada

% cerradura:

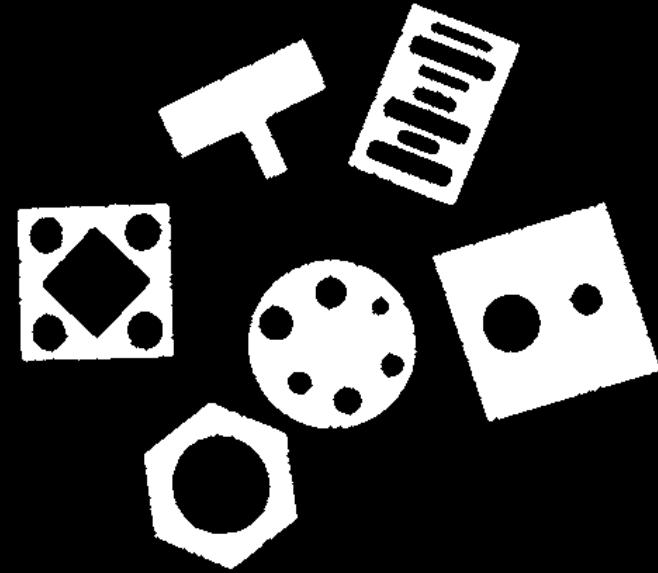
```
gdc = imdilate(gda,se); % dilatación de imagen  
figure(6)  
imshow(gdc)  
title('Imagen dilatada')
```

```
gec = imerode(gdc,se); % erosion de imagen  
figure(7)  
imshow(gec)  
title('Imagen erosionada')
```

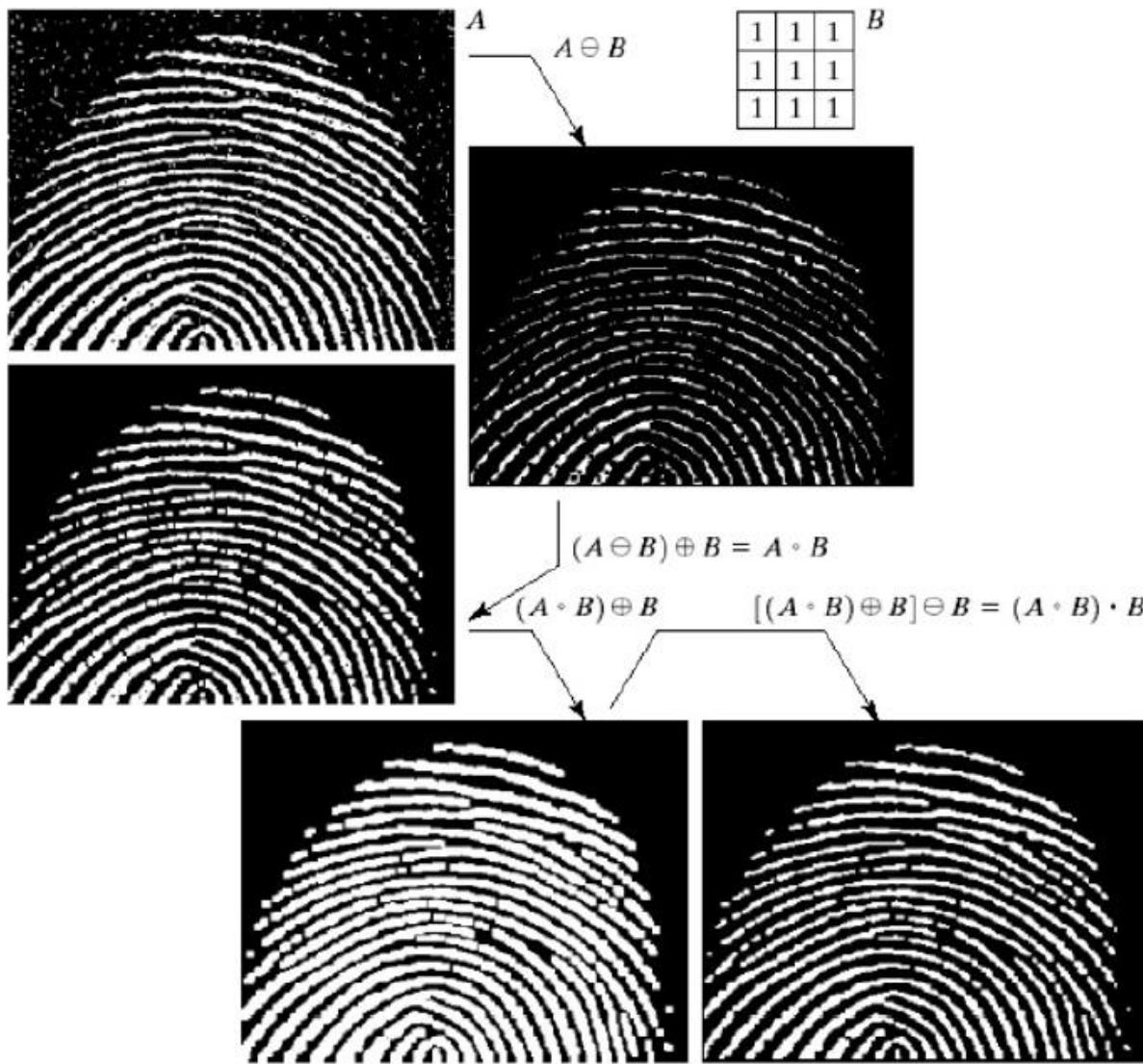
# **cerradura**



**Imagen dilatada**



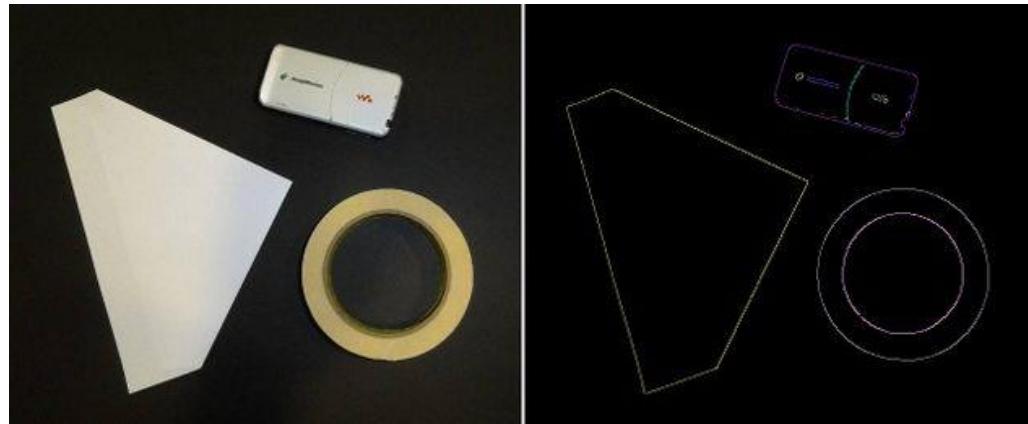
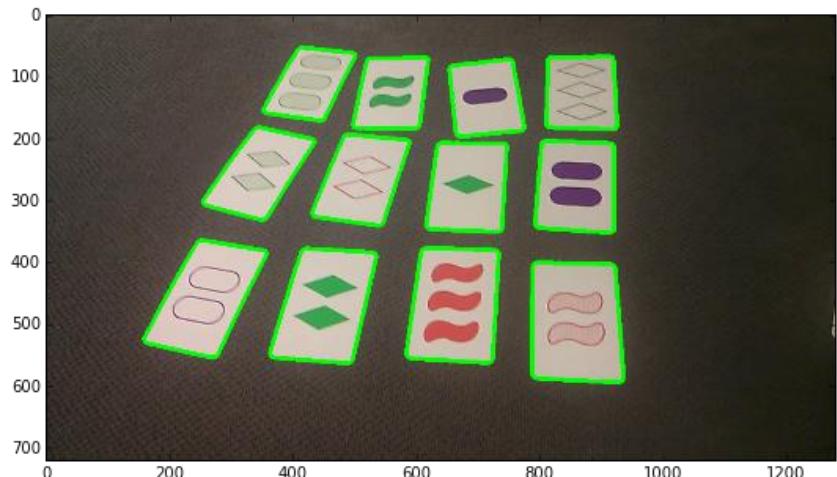
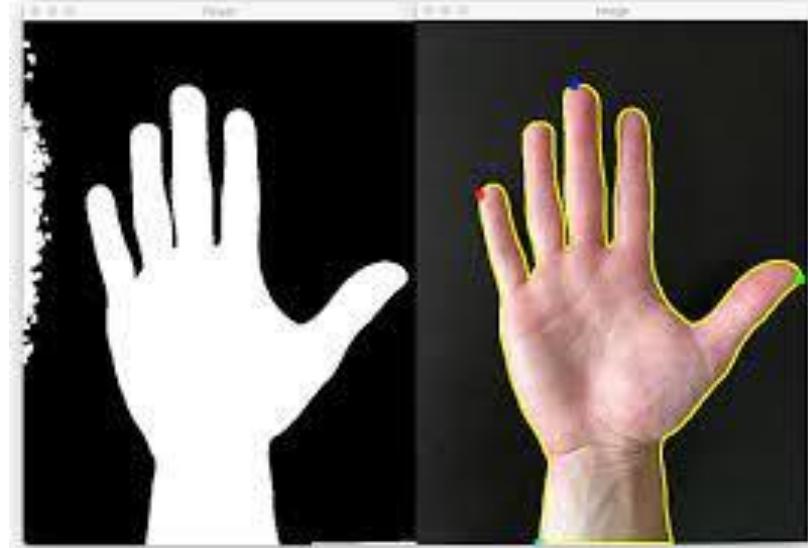
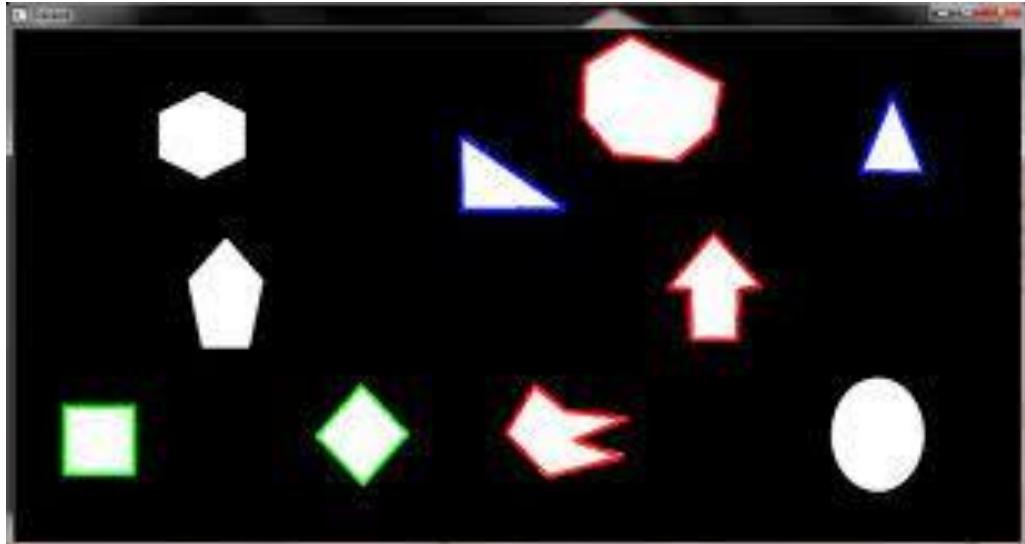
**Imagen erosionada**



Otro ejemplo de aplicación de filtros de apertura y cierre para reducir el ruido en imágenes binarias de huellas dactilares (González y Woods, 2006).

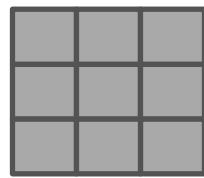
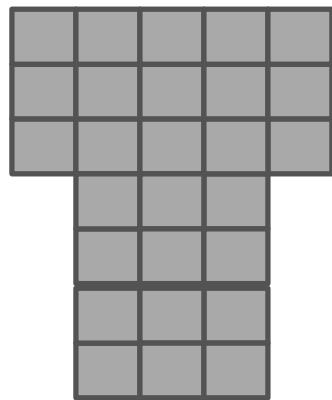
# Detección de contornos en imágenes:

# Detección de contornos:

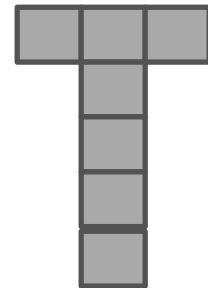


# Detección de contornos en imágenes:

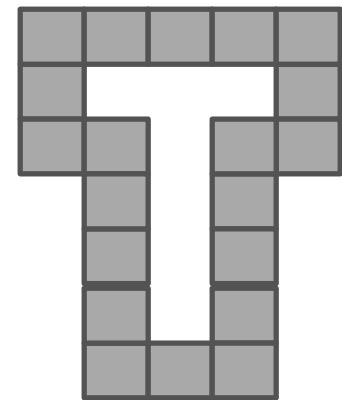
$A - (A \odot B)$



$(A \odot B)$



$A - (A \odot B)$



% Detección de contornos mediante operaciones morfológicas.

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

```
f=imread('6 objetos.JPG');  
figure(1)  
imshow(f)  
title('Imagen original')  
  
[n,m]=size(f);
```

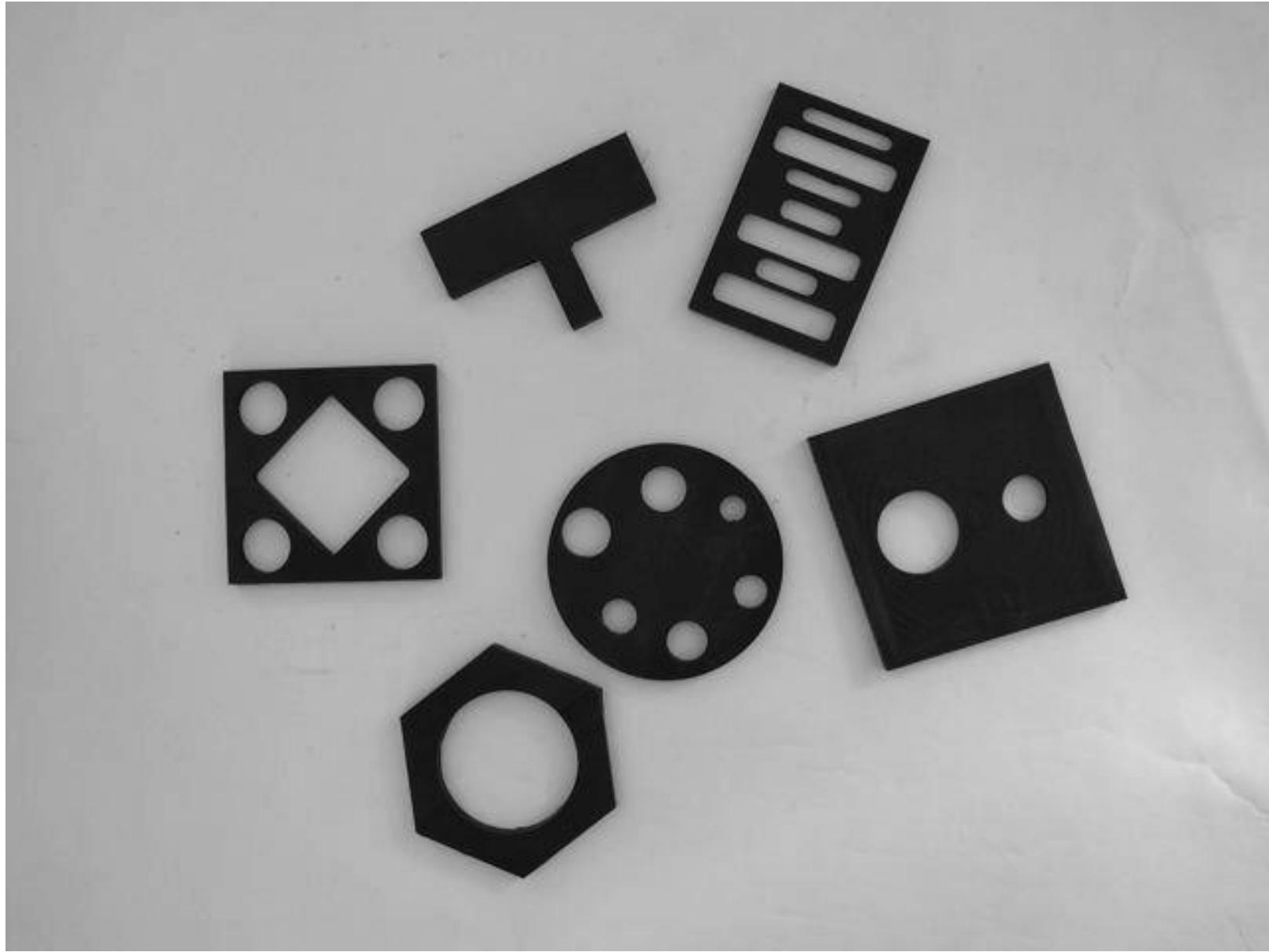


Imagen original

```
th=90; % umbral  
  
for y=1: m  
    for x=1: n  
        if f(x, y) >= th  
            g(x, y)=0;  
        else  
            g(x, y)=255;  
        end  
    end  
end  
  
figure(2)  
imshow(g)  
title('Imagen binaria')
```

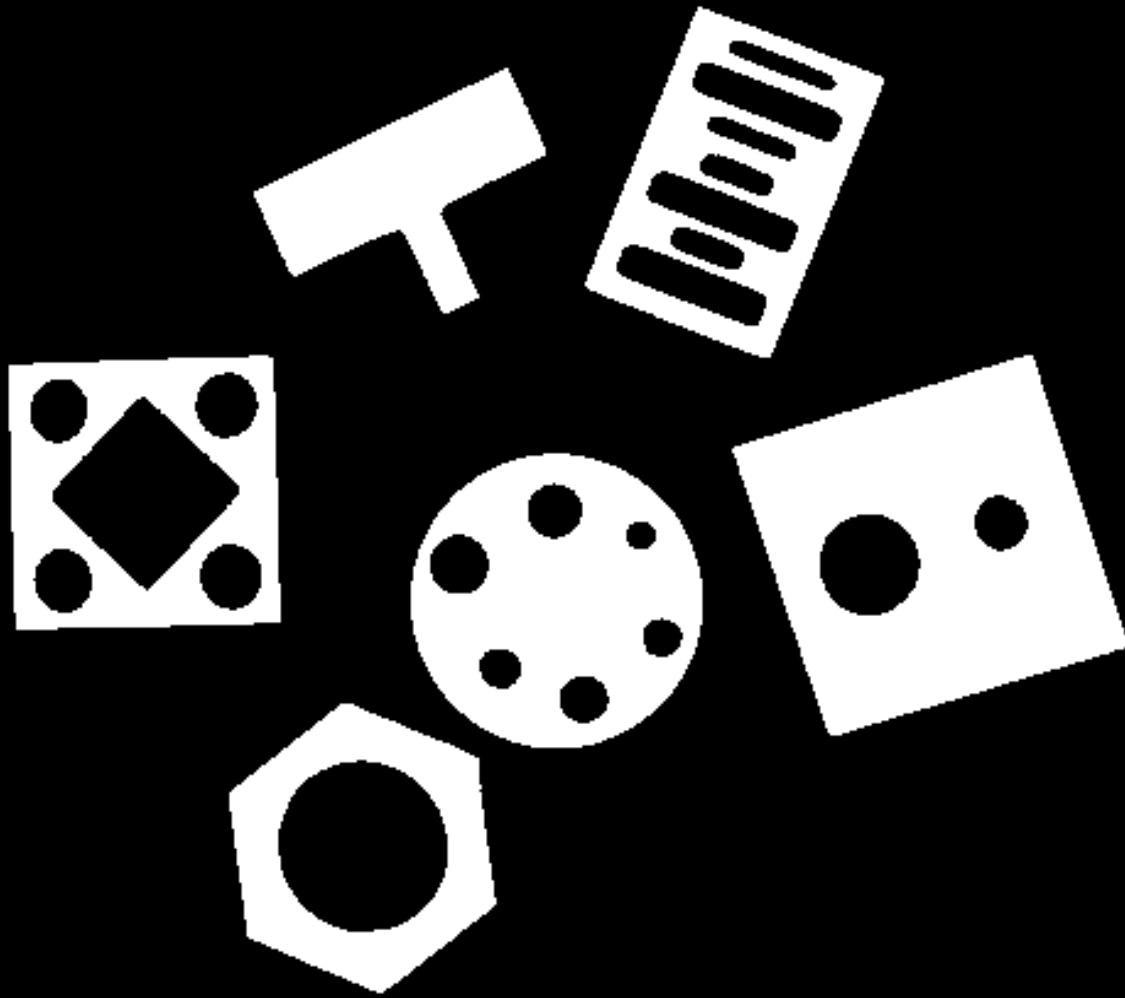


Imagen binaria

% erosión:

```
se = strel('disk',1); % square 7x7, line lenght  
y angulo, disk  
ge = imerode(g,se); % erosion de imagen  
figure(3)  
imshow(ge)  
title('Imagen erosionada')
```

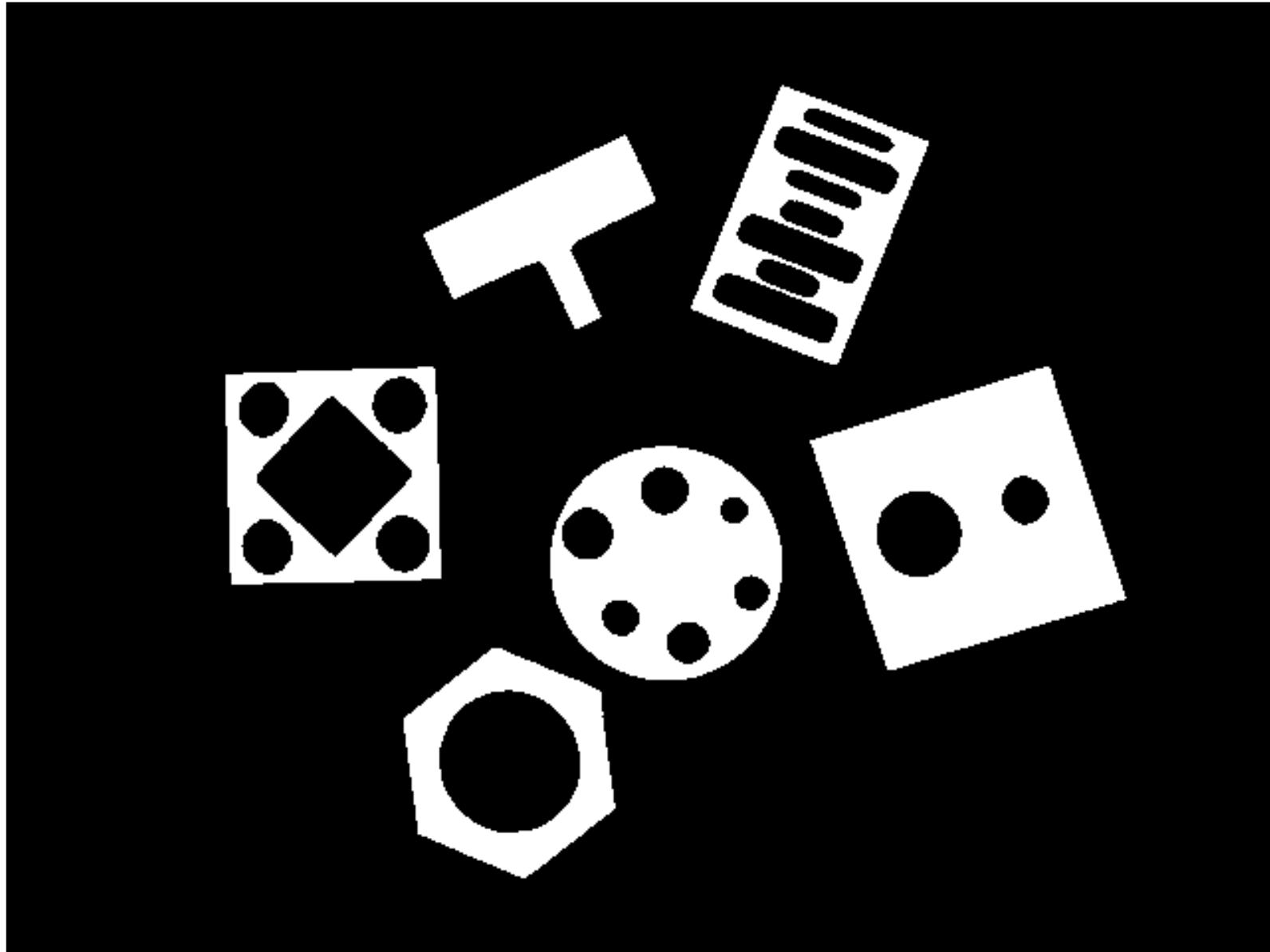
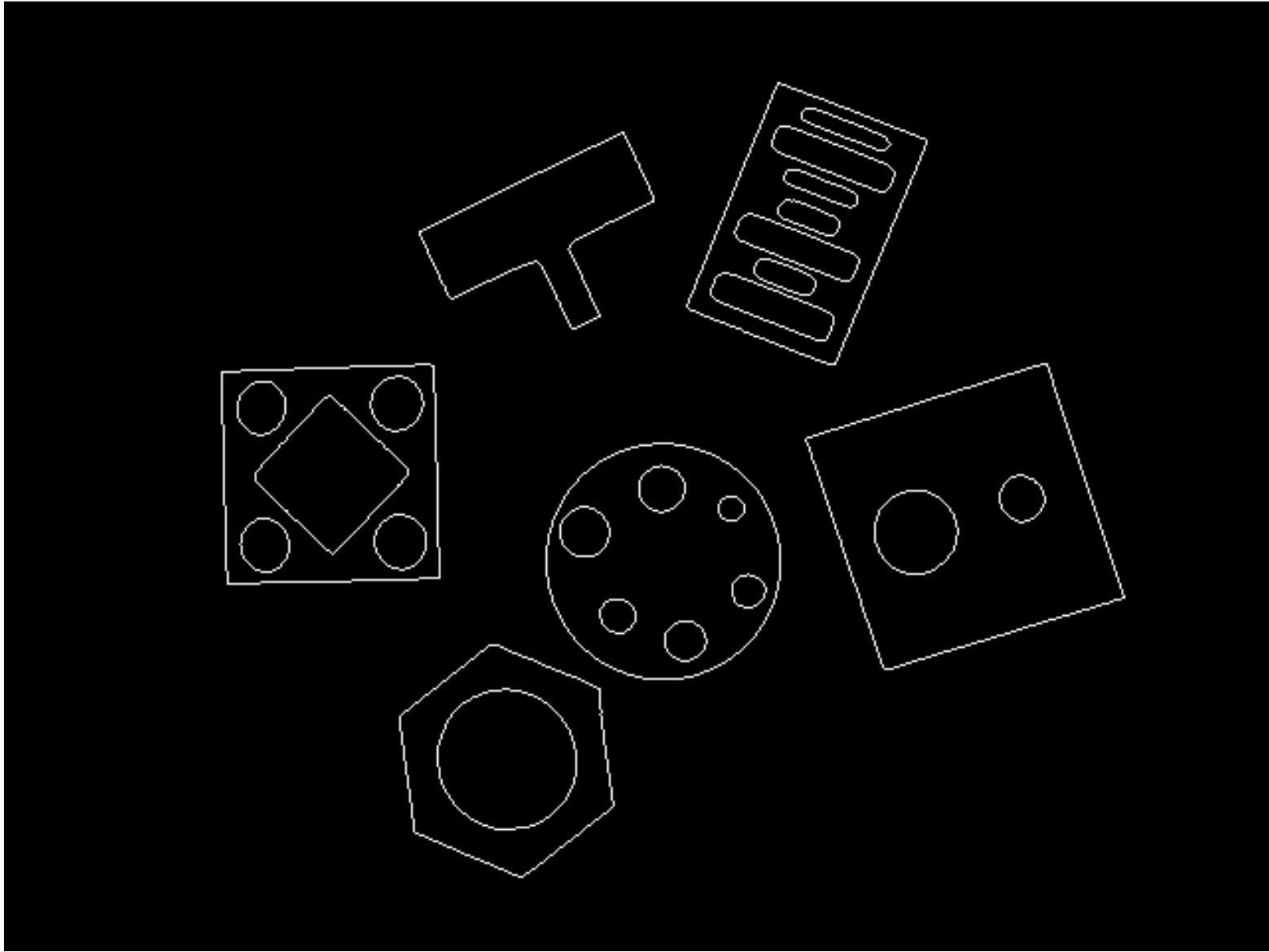


Imagen erosionada

% resta de imagen menos imagen erosionada:

contornos = g - ge;

```
figure(4)
imshow(contornos)
title('Contornos')
```



Contornos

# MATHEMATICAL MORPHOLOGY IN IMAGE PROCESSING



EDITED BY  
EDWARD R. DOUGHERTY

IMAGE ANALYSIS AND



MATHEMATICAL  
MORPHOLOGY  
VOLUME 1

JEAN SERRA

IMAGE PROCESSING  
and MATHEMATICAL  
MORPHOLOGY  
Fundamentals and Applications

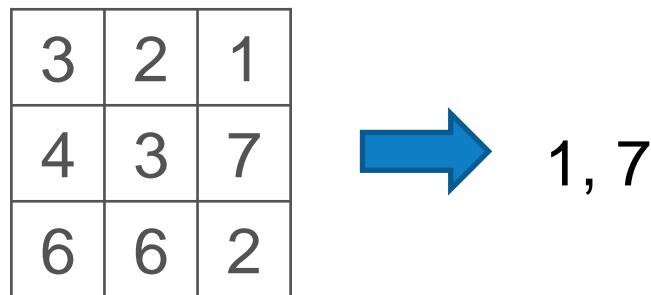
Frank Y. Shih

EBC Press  
Society & Technology

**Aplicación a  
imágenes con  
niveles de gris  
(no binarias):**

Aunque originariamente las operaciones de morfología matemática se crearon para imágenes binarias, en seguida se extendieron a imágenes con niveles de gris.

Para ello, tan sólo hay que tener en cuenta que la erosión **equivale a la aplicación de un filtro de mínimo** en imágenes con niveles de gris, mientras que la dilatación equivale a la aplicación de un **filtro de máximo**, siempre considerando el mismo vecindario que el definido por el elemento estructural.



Así por ejemplo, si queremos aplicar una operación de erosión con un elemento estructural de los 8 vecinos más próximos (figura b) a una imagen en niveles de gris, tendremos que asignar al píxel central el valor mínimo de los 9 píxeles vecinos de la imagen original.

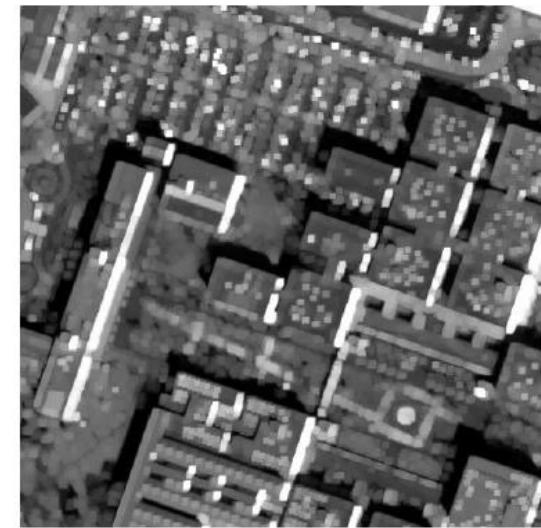
**Ejemplo:** En la figura se muestra un ejemplo de la aplicación de un filtro de **erosión**  $5 \times 5$  (figura b) y otro de **dilatación**  $5 \times 5$  (figura c) sobre un fragmento de una imagen pancromática del satélite QuickBird.



(a)



(b)



(c)

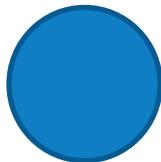
La erosión produce un incremento de las formas y espacios oscuros (bajos niveles de gris) en detrimento de los claros (altos niveles de gris), la dilatación tiene el efecto opuesto.

Así por ejemplo, en la imagen erosionada aumenta el tamaño de las sombras, los coches oscuros y los árboles, mientras que en la imagen dilatada aumenta el tamaño de los bordes de fachadas y de las rotondas con elevada reflectancia, así como de los coches claros.

## Extracción de objetos en la imagen:

La extracción automática de objetos utilizando filtros morfológicos es indicada cuando los objetos que queremos identificar tienen una forma y tamaño constantes en toda la imagen.

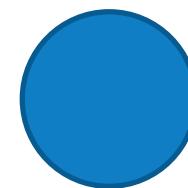
Para ello, es necesario definir dos tipos de elementos estructurales, ambos de la misma forma que los objetos, uno de ellos de tamaño ligeramente más pequeño y el otro ligeramente más grande.



Objeto de interés



EE más pequeño



EE más grande

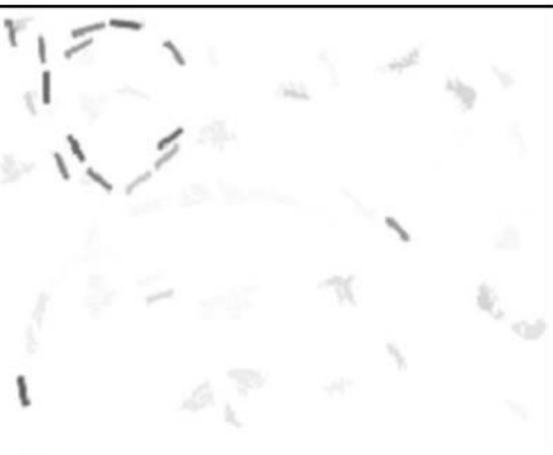
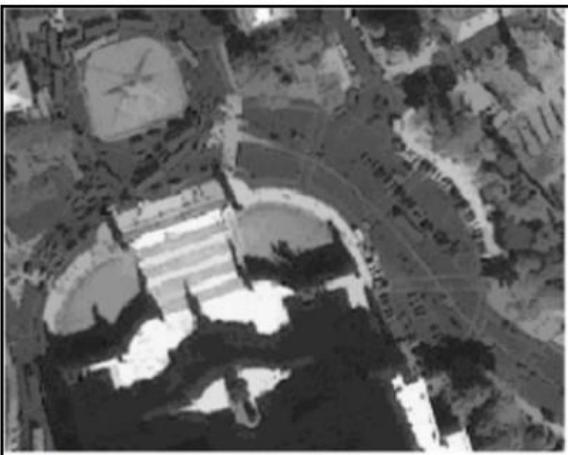
Veamos un ejemplo propuesto por Soille y Pesaresi (2002) para la extracción de autobuses en una zona urbana de Roma en imágenes pancromáticas del satélite Ikonos (figura).



Soille, P., and Pesaresi, M., 2002. Advances in mathematical morphology applied to geoscience and remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 40 (9), pp. 2042-2055.

Para ello, crearon dos baterías de elementos estructurales, uno de ellos (A) con elementos de tamaño algo inferior al de los autobuses y el otro (B) de tamaño algo mayor:





- a) Imagen IKONOS (Pan) original
- b) Apertura con EE ligeramente más pequeño que los autobuses
- c) Apertura con EE ligeramente mayor que los autobuses
- d) Diferencia entre (b) y (c)
- e) Umbralización de (d)



Extracción automática de autobuses en una imagen pancromática de alta resolución, con indicación de los pasos seguidos, desde la imagen original (a) hasta los autobuses detectados (e).

# Digital Image Processing

4

FOURTH EDITION



Rafael C. Gonzalez  
Richard E. Woods

P Pearson

Soille, P., and Pesaresi, M., 2002. Advances in mathematical morphology applied to geoscience and remote sensing. IEEE Transactions on Geoscience and Remote Sensing, 40 (9), pp. 2042-2055.

UNIVERSIDAD  
INTERNACIONAL  
DE LA RIOJA

unir

[www.unir.net](http://www.unir.net)