

# Evaluación de la segmentación

**Juan Humberto Sossa Azuela**

**“GRABAR”**

# Objetivo:

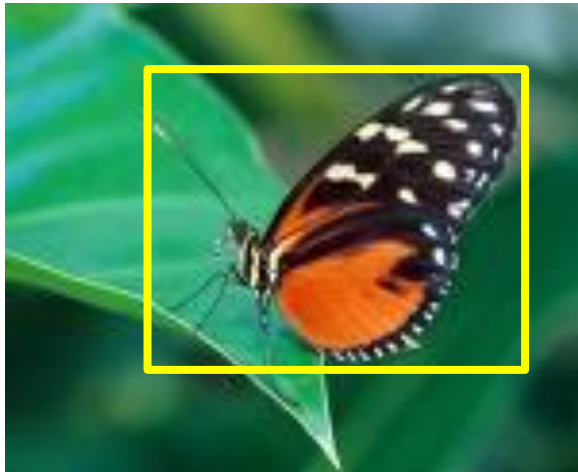
**Aprender a construir y evaluar el rendimiento de uno o más segmentadores de imágenes.**

**Esta actividad permitirá consolidar los conceptos sobre segmentación de imágenes aprendidos.**

# Descripción:

La segmentación, como se ha visto, consiste en detectar regiones homogéneas y aislar / detectar objetos dentro de una imagen.

Estas regiones habitualmente corresponden a los objetos que se están queriendo identificar.



# Descripción:

Existen muchas maneras de enfocar este problema y se puede hacer uso de las técnicas de segmentación que se considere para resolverlo.

Una vez elegidas estas técnicas, se debe evaluar su rendimiento frente a imágenes de *ground truth*.



# Descripción:

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

**Se debes mostrar en pantalla los resultados de los principales pasos.**

# Descripción:

**Escoger una o dos imágenes que considere representativas de un determinado problema y aplique varios segmentadores sobre ellas para evaluar cuál de ellos ofrece el mejor resultado.**

# Forma de entrega:

**Debes adjuntar la memoria y el código fuente usado.**

**La extensión máxima del informe ha de ser 6/8 páginas**

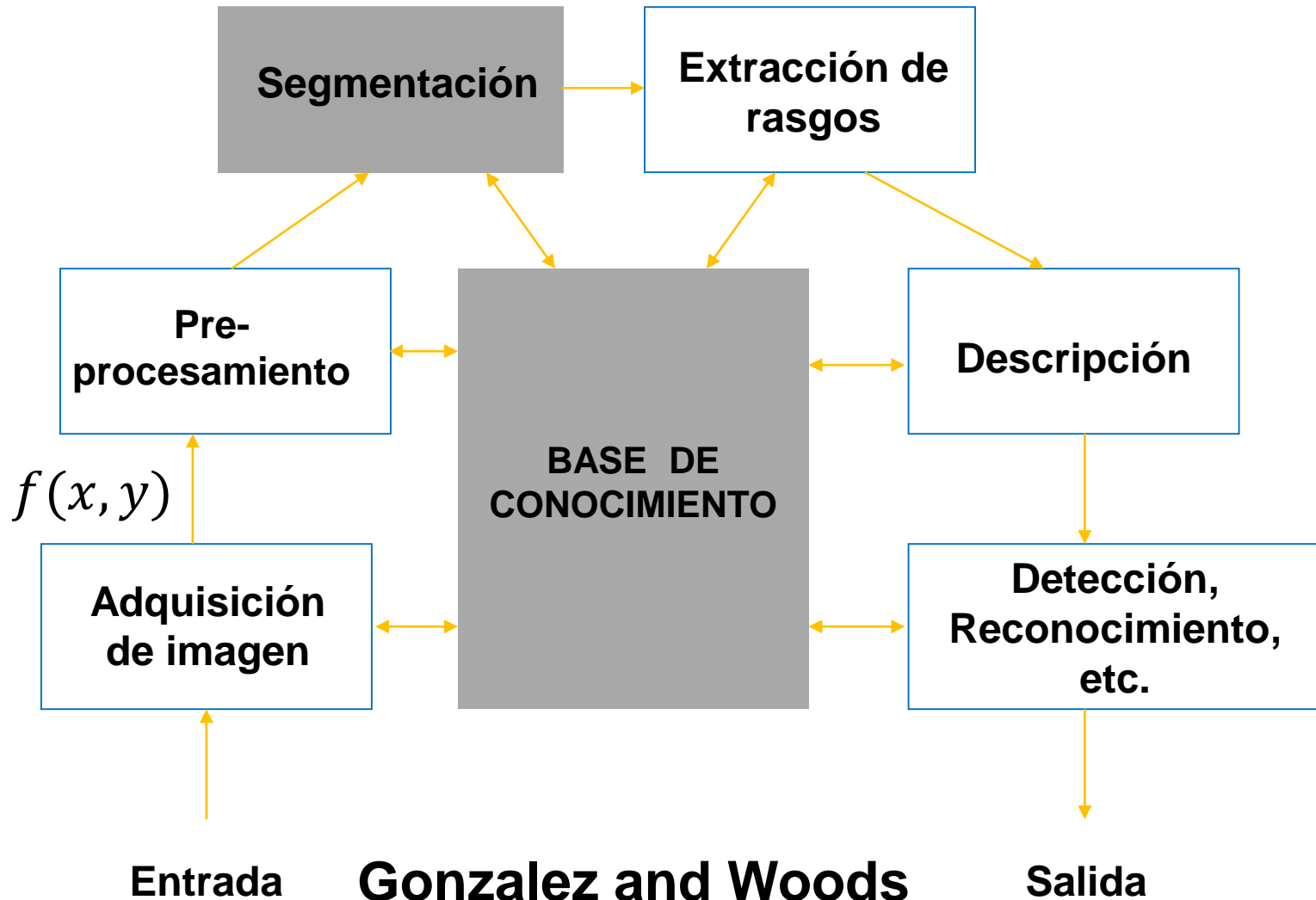
# Rúbrica:

Criterio	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	El segmentador funciona correctamente	2	20%
Criterio 2	Existe una implementación propia y rigurosa de al menos la parte principal donde el estudiante evalúa sus segmentaciones	3	30%
Criterio 3	El estudiante ha comparado más de un segmentador con varias imágenes y <i>ground truths</i>	2	20%
Criterio 4	Código claro, comprensible y muestra los principales pasos	2	20%
Criterio 5	Memoria clara y sin carencias	1	10%
Criterio 6	Existe plagio no debidamente referenciado o el trabajo es similar y con los mismos puntos de fallo que el de otro estudiante	-10	-100%
		10	100 %

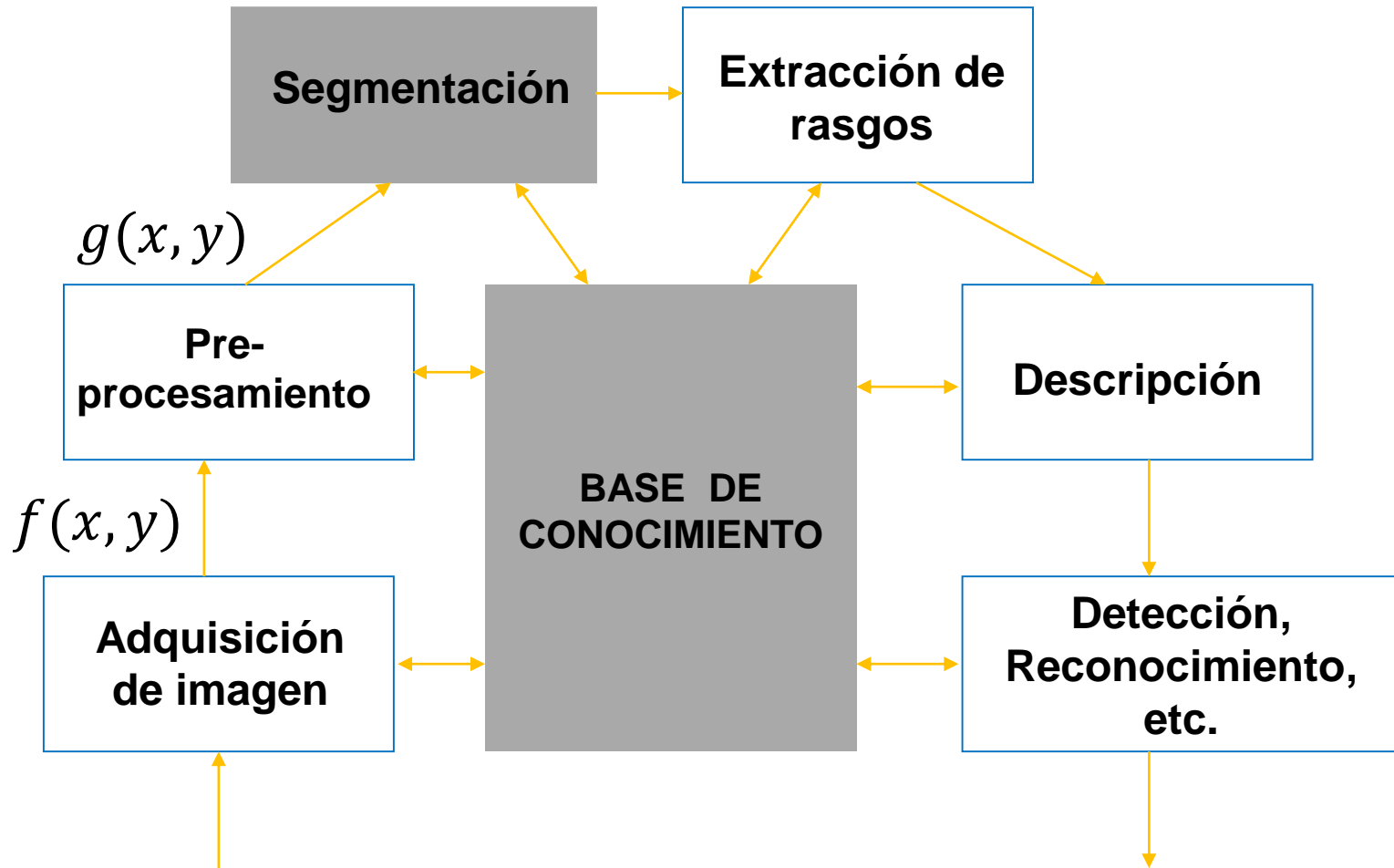


# Segmentación de imágenes:

# UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



# UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS

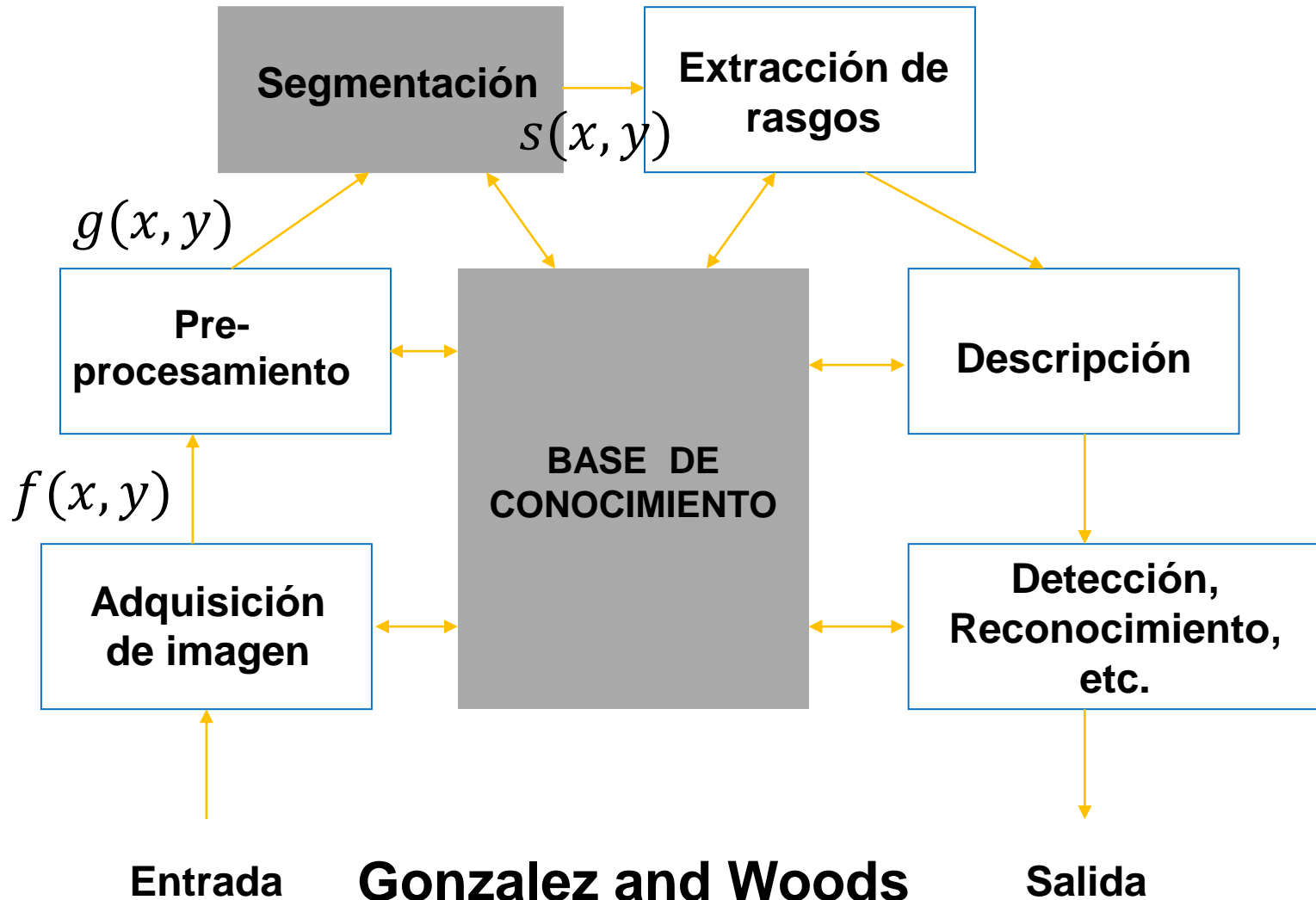


Entrada

**Gonzalez and Woods**

Salida

# UN PARADIGMA PARA EL RECONOCIMIENTO DE OBJETOS



# Contenido:

Introducción.

Umbralado manual.

Umbralado automático.

Extensión al caso de umbralado múltiple.

Etiquetado de componentes conectadas.

El problema del umbralado de imágenes como un problema de clasificación.

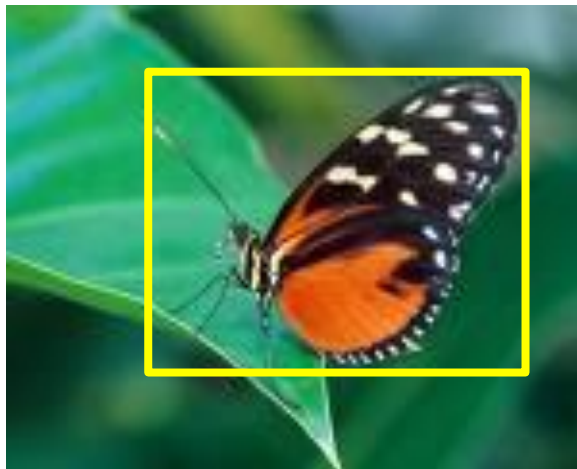
Segmentación de imágenes por crecimiento de regiones.

Segmentación por ajuste de cerco cóncavo-convexo.

# Introducción:







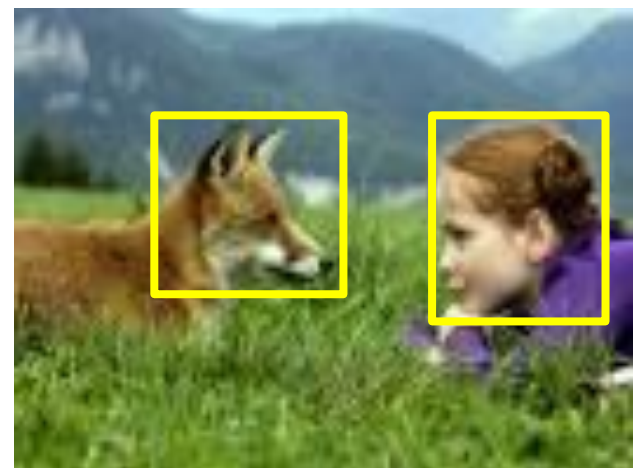




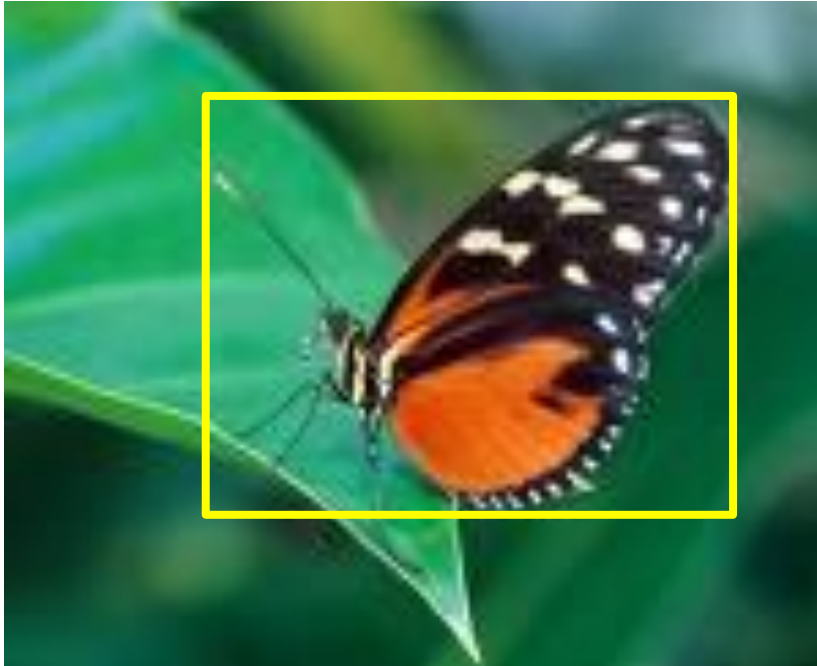








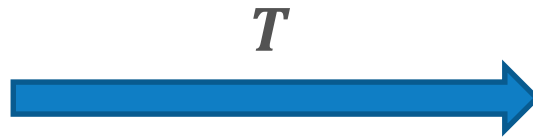
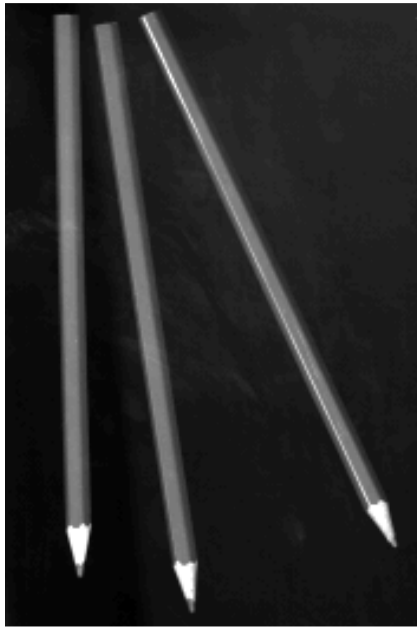
## Dos segmentaciones del mismo objeto:



¿Cuál es mayor?



# Ejemplos:



# Ejemplos:



# OBJETIVO DE LA SEGMENTACIÓN:

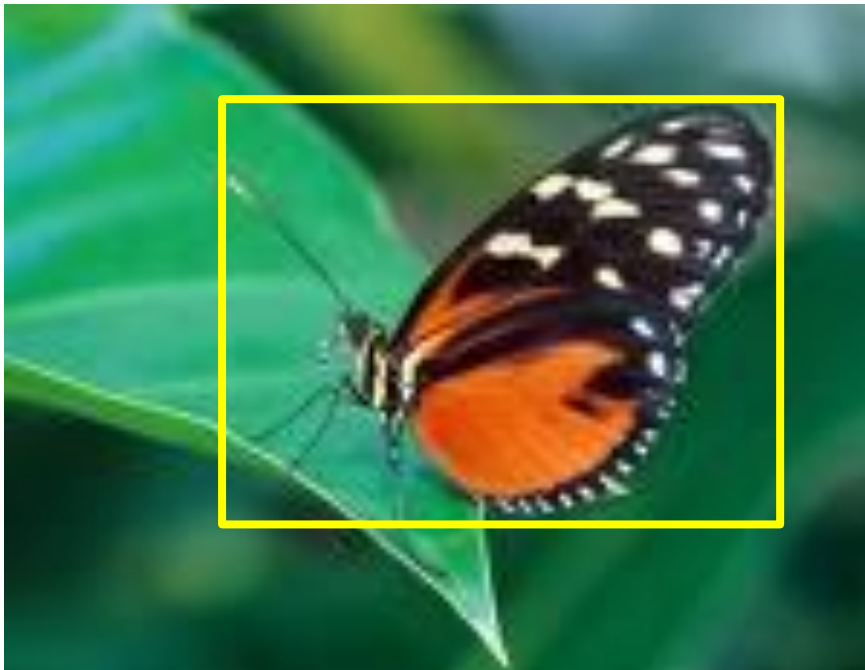
**Encontrar, de alguna manera, regiones en una imagen que representen objetos o partes significativas de objetos.**





# OBJETIVO DE LA SEGMENTACIÓN:

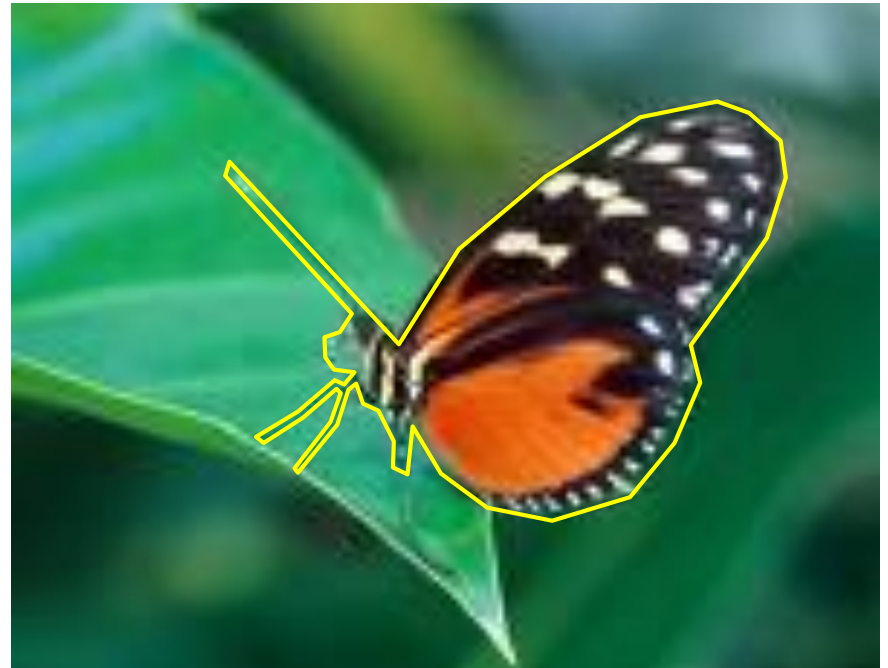
**Encontrar, de alguna manera, regiones en una imagen que representen objetos o partes significativas de objetos.**



**Segmentaciones de dos objetos.**

# OBJETIVO DE LA SEGMENTACIÓN:

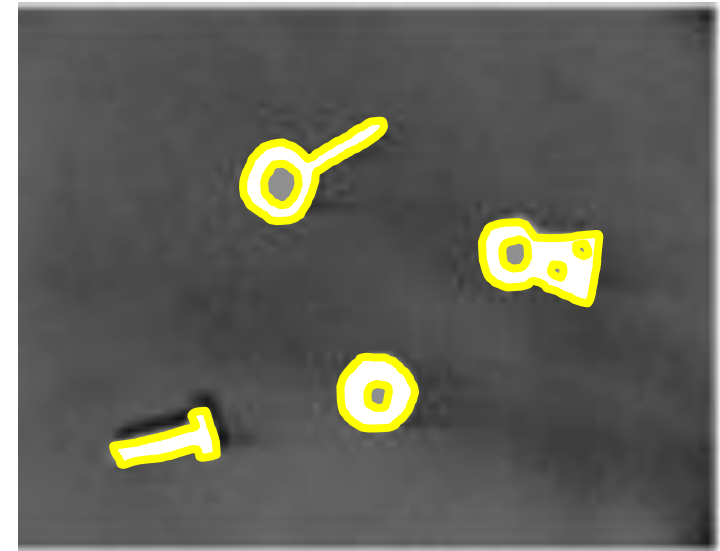
**Encontrar, de alguna manera, regiones en una imagen que representen objetos o partes significativas de objetos.**



**Otra segmentación (más precisa) de un objeto.**

# DEFINICIÓN:

La segmentación de una imagen puede definirse como el proceso que consiste en **subdividir o particionar** una imagen en sus regiones, componentes, partes u objetos.



La segmentación puede también ser definida como el encontrar, por medio de un algoritmo numérico, sus regiones homogéneas y sus bordes.

# Gonzalez y Woods

Una manera formal de definir el proceso de segmentación es la siguiente: La segmentación es el proceso de particionar una imagen  $f(x, y)$  en regiones de píxeles  $R_1, R_2, \dots, R_n$ , tal que cada una de estas sub-imágenes represente un objeto o una parte del mismo. La segmentación es, pues, el proceso que consiste en agrupar píxeles en regiones,  $R_1, R_2, \dots, R_n$ , tales que:

$$(a) \bigcup_{i=1}^n R_i \subseteq f(x, y) \quad (b) R_i, i = 1, 2, \dots, n \quad \text{es conectada.}$$

$$(c) R_i \cap R_j = \emptyset, \forall i, j, i \neq j \quad (d) P(R_i) \text{ es verdadero } i = 1, 2, \dots, n$$

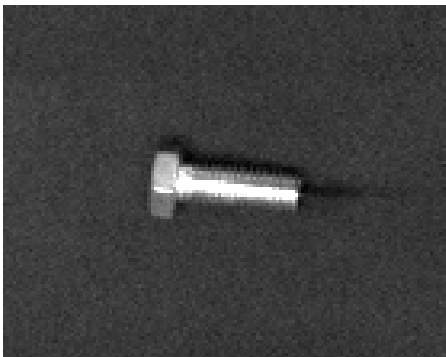
$$(e) P(R_i \cup R_j) \text{ es falso para } i \neq j$$

**¿Cómo realizar dicha partición?**

**y**

**¿Qué predicados o conjuntos de propiedades se deben tomar en cuenta para obtener la partición deseada?**

La primera tiene que ver con la selección de un método para obtener la partición, la segunda tiene que ver con las características o propiedades que deberán satisfacer los píxeles de la imagen para ser agrupados en regiones.



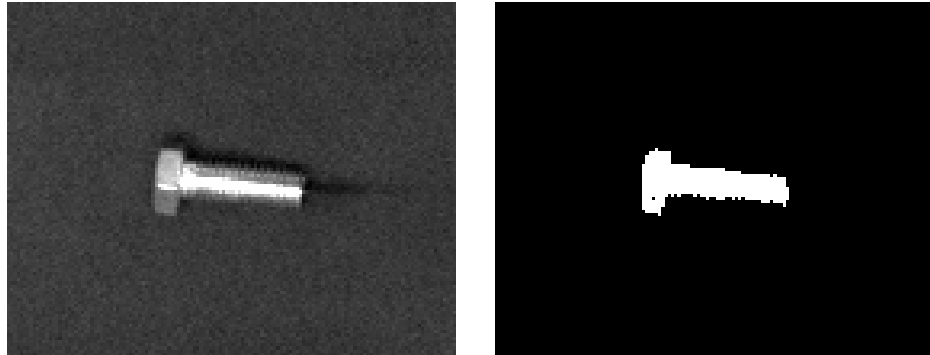
(a) Imagen.

(b) Segmentación al usar como característica el nivel de gris de los píxeles de la imagen

# Idea de la segmentación:

Agrupar píxeles con características similares en regiones que representen objetos en la imagen.

Clasificar píxeles en clases que correspondan a objetos de interés en la imagen.



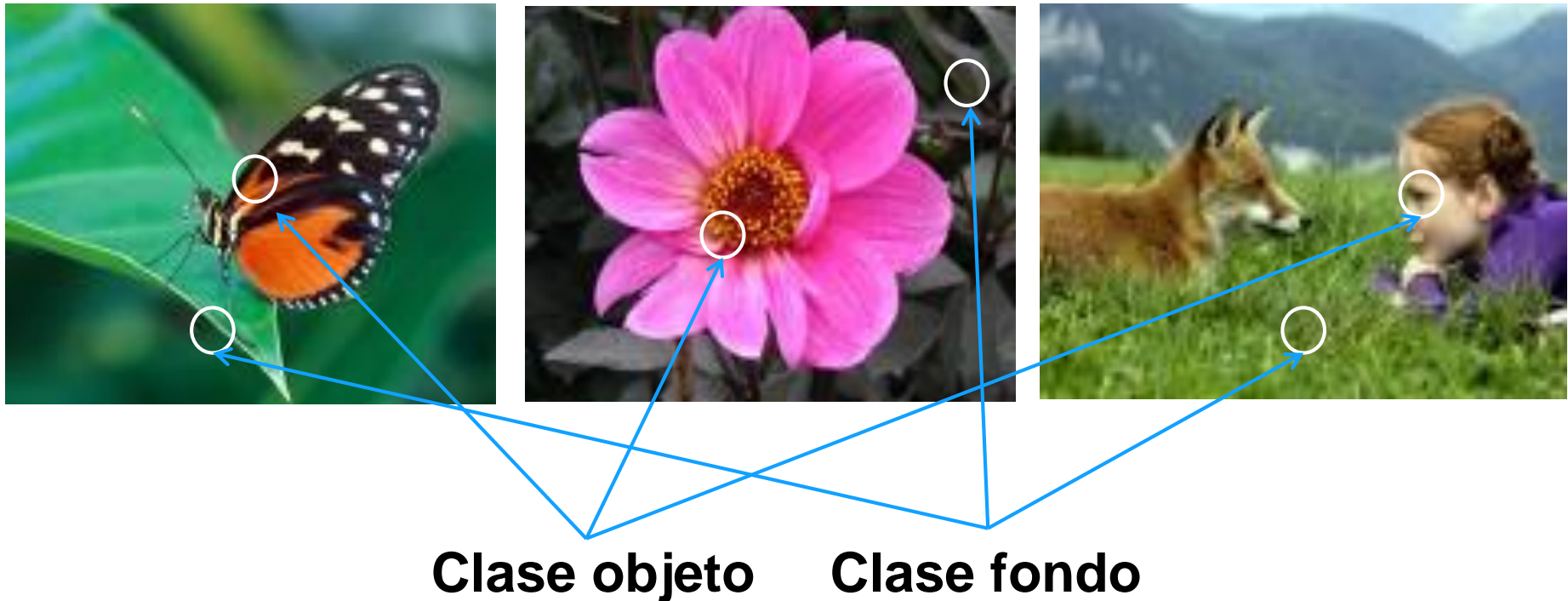
Las características de un píxel son:

- su posición  $(x, y)$  y
- su nivel de gris o color:  $[(x, y), r]$

**¡Con estas dos características se puede hacer poco...!**

## Una mejor idea consiste en:

Abrir una ventana alrededor del píxel y construir un vector descriptivo del píxel sobre la base de la información contenida dentro de la ventana y entonces procede a agrupar o clasificar píxeles.



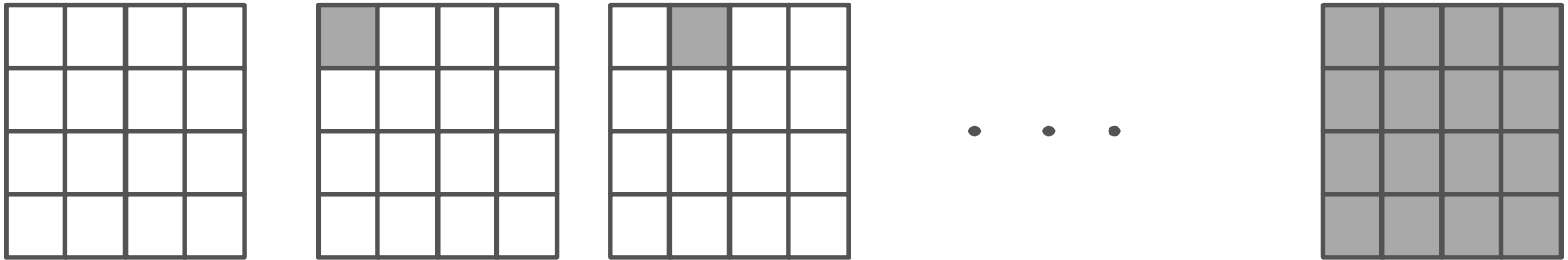






**La segmentación de imágenes, en general, es un problema muy difícil de resolver de complejidad NP.**

## Complejidad del problema de la segmentación de imágenes:



Número de particiones posibles:

$$\binom{16}{1} + \binom{16}{2} + \dots + \binom{16}{16} = 65535 = 2^{16} - 1$$

Para una imagen de  $n \times n$  píxeles:

$$2^{n \times n} - 1$$

# Número de segmentaciones de una imagen en términos de su tamaño:

$n$	Número de segmentaciones
1	1
2	15
3	511
4	65,535
5	33,554,432
6	68,719,476,736
⋮	⋮

# **HECHO:**

**Si se obtiene una buena segmentación, se obtiene una buena descripción y, en consecuencia, un buen reconocimiento; por el contrario, si se obtiene una mala segmentación, se obtendrán malas descripciones y, por tanto, malas identificaciones.**





# Paradigmas para la segmentación:

Por umbralado.

Por agrupamiento de regiones.

Por encercado de regiones.

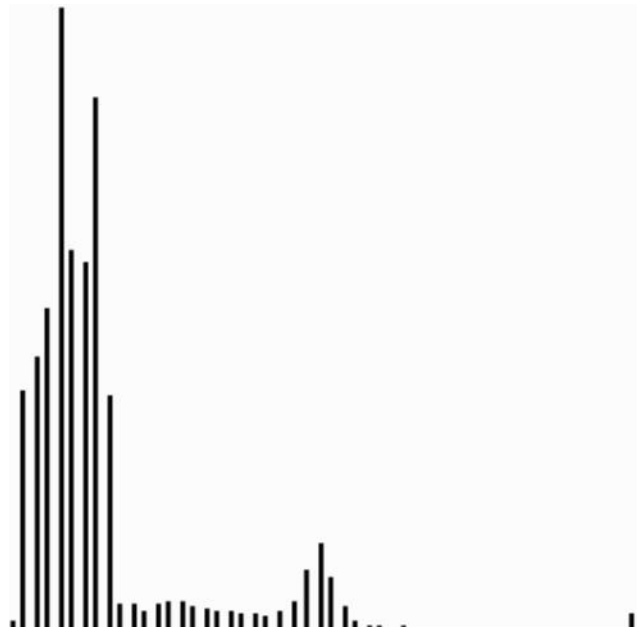
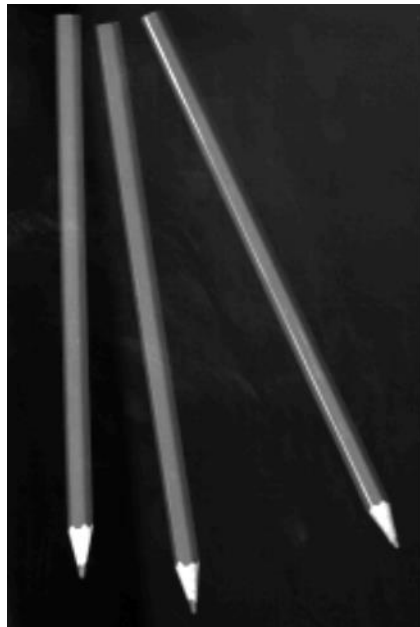
Por etiquetado de píxeles.

Por clasificación de píxeles.

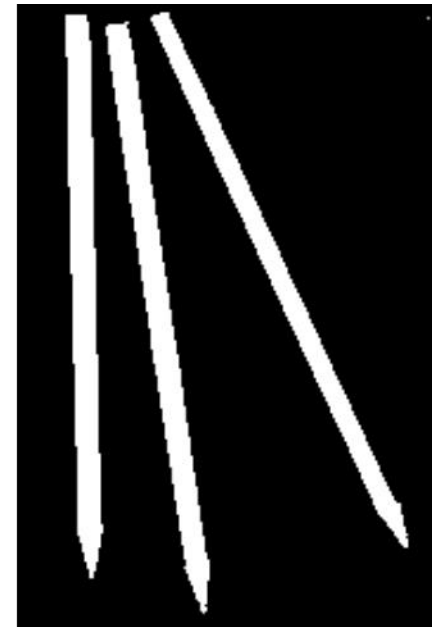
Por umbralado heurístico.

Etc.

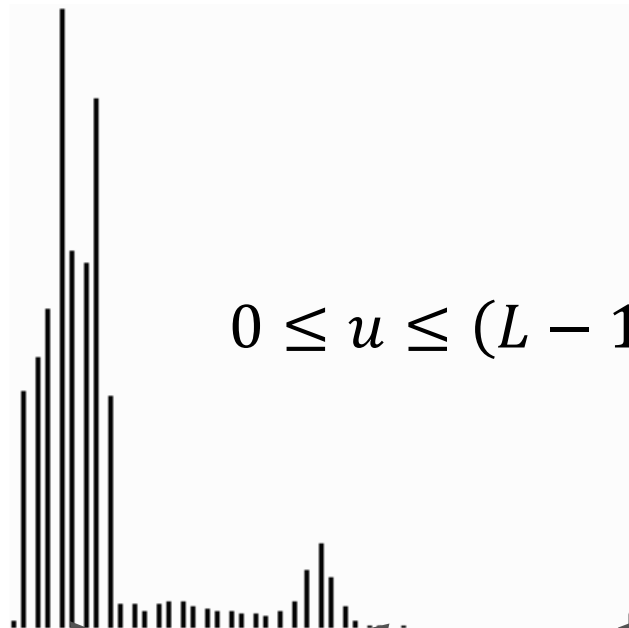
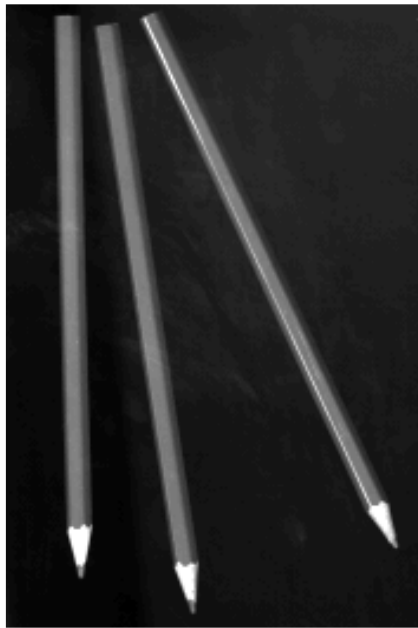
## Técnicas basadas en el umbralado:



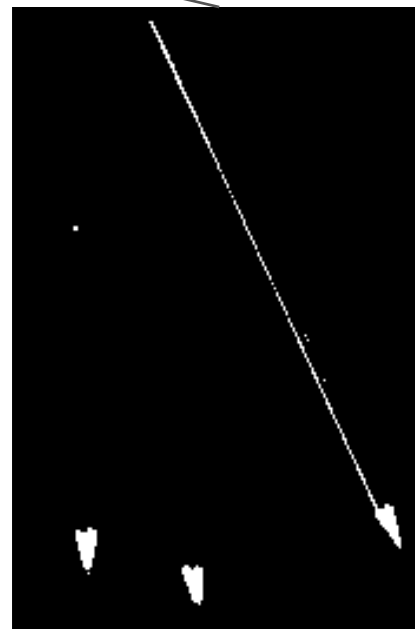
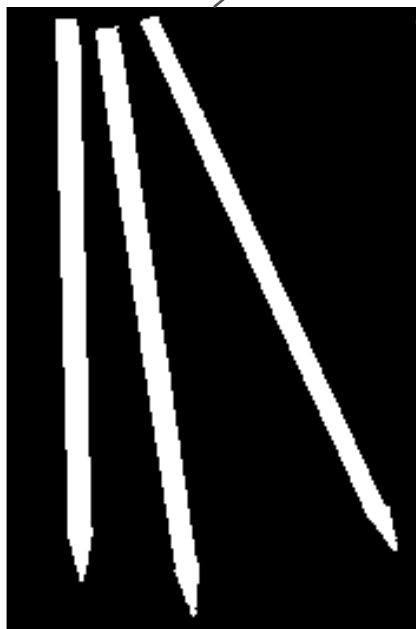
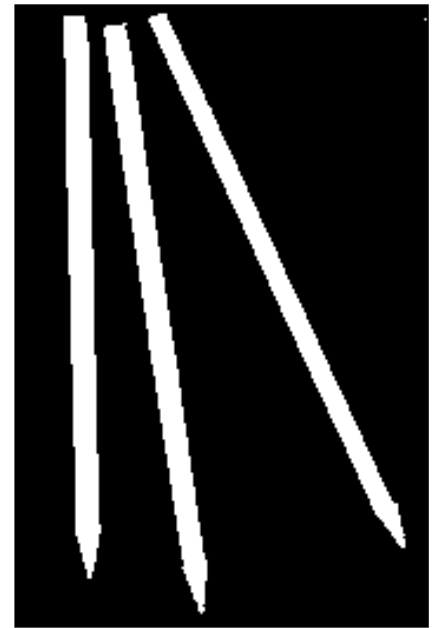
$u$





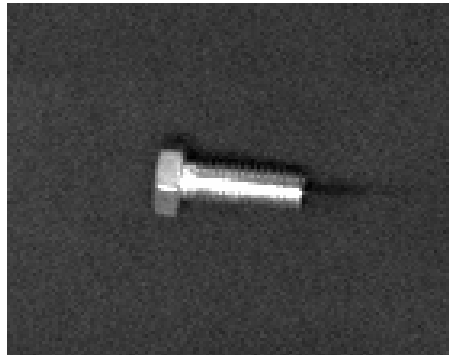


$$0 \leq u \leq (L - 1)$$



## Técnicas basadas en el umbralado:

$$f(x, y) \xrightarrow{T} b(x, y)$$



## Técnicas basadas en el umbralado

$$f(x, y) \xrightarrow{T} b(x, y)$$

Hipótesis de partida:

$$b(x, y) = \begin{cases} L-1 & \text{si } f(x, y) \geq u \\ 0 & \text{en otro caso} \end{cases}$$

Si se conoce el rango de valores del objeto:  $[u_1, u_2]$

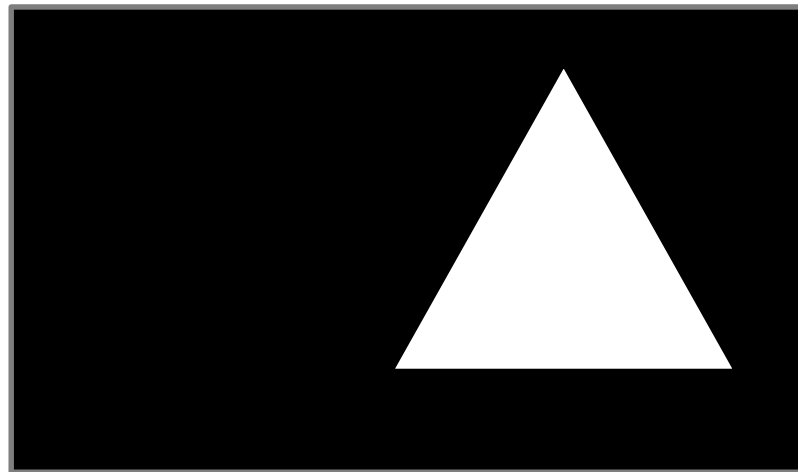
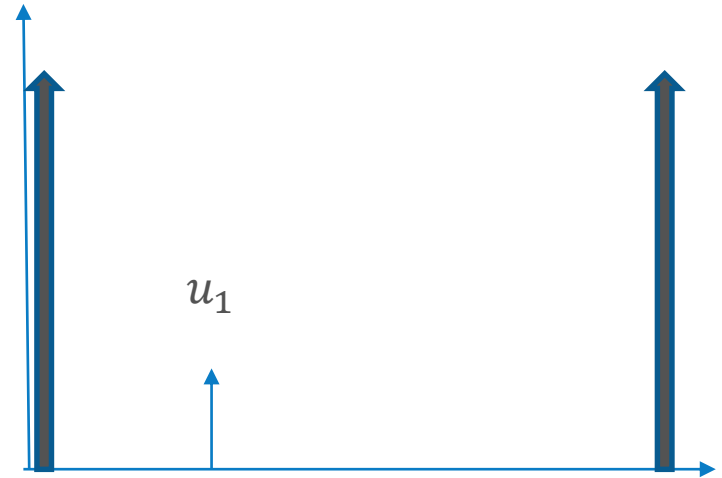
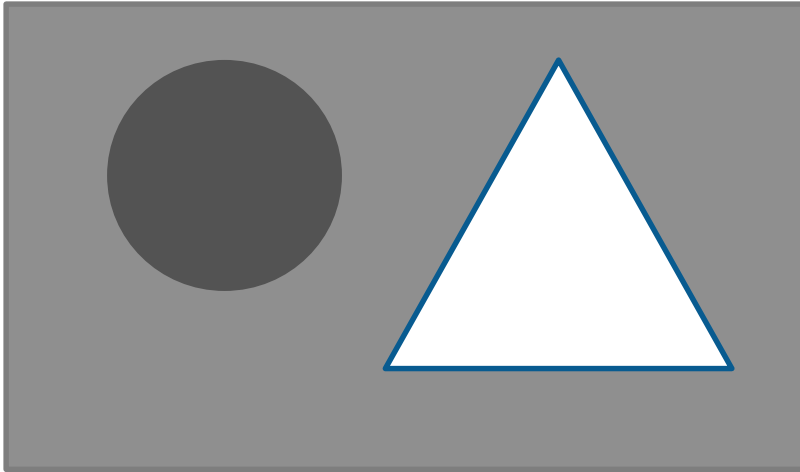
$$b(x, y) = \begin{cases} L-1 & \text{si } u_1 \leq f(x, y) \leq u_2 \\ 0 & \text{en otro caso} \end{cases}$$

Caso general de objetos con diferentes niveles de gris:

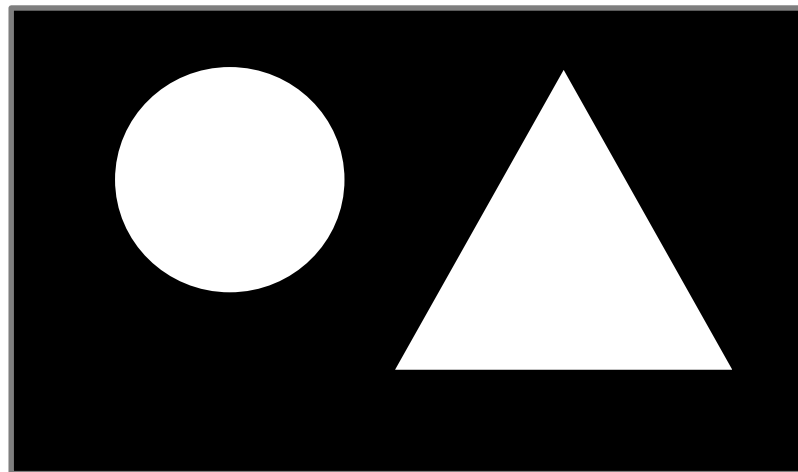
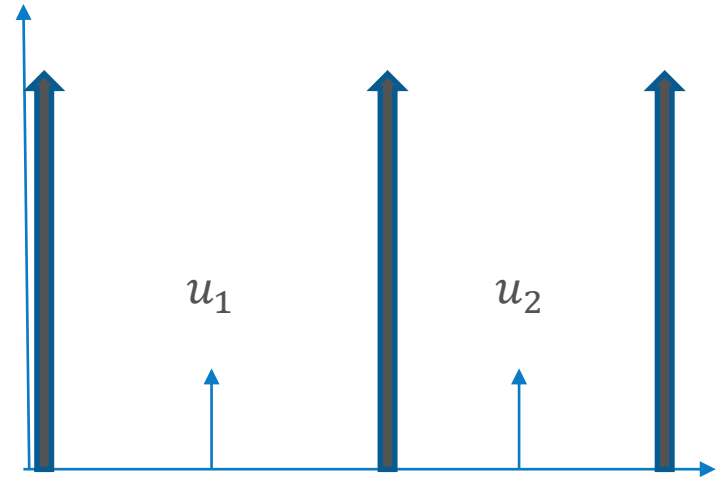
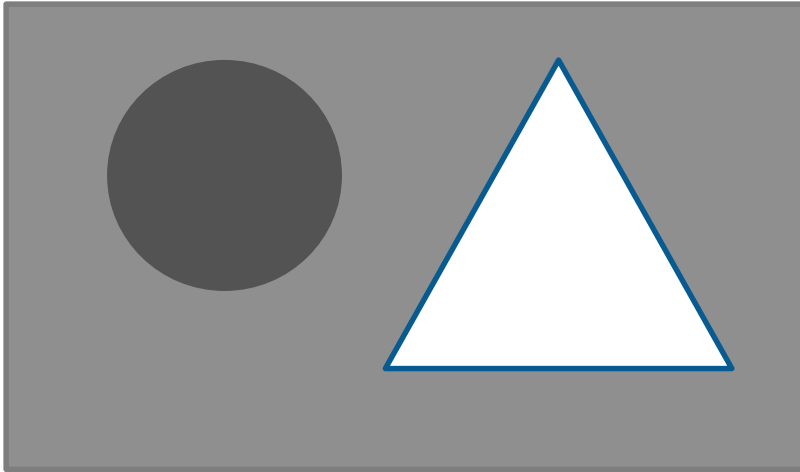
$$b(x, y) = \begin{cases} L-1 & \text{si } f(x, y) \in C \\ 0 & \text{en otro caso} \end{cases}$$

donde  $C$  es el conjunto de valores de intensidad de la superficie de cada objeto.

Ejemplo que no se puede resolver con un solo umbral:



Ejemplo que no se puede resolver con un solo umbral:



## Notación:

$r$  el nivel de gris de un píxel con coordenadas  $(x, y)$ .

$L$  el número de niveles de gris de la imagen  $f(x, y)$  con  $0 \leq r \leq L - 1$

$p_r$  es el número de píxeles con el nivel de gris  $r$

$N_T = p_0 + p_1 + \dots + p_r + \dots + p_{L-1}$  es el número total de píxeles.

$p(r) = \frac{p_r}{N_T}$  el histograma normalizado de niveles de gris de  $f(x, y)$ , tal que

$$\sum_{r=0}^{L-1} p(r) = 1.0$$

$u$  el umbral a determinar.

El objetivo consiste en encontrar el valor óptimo de  $u$ ,  $u^*$ , al **maximizar o minimizar una función criterio.**

# Umbralado Manual:

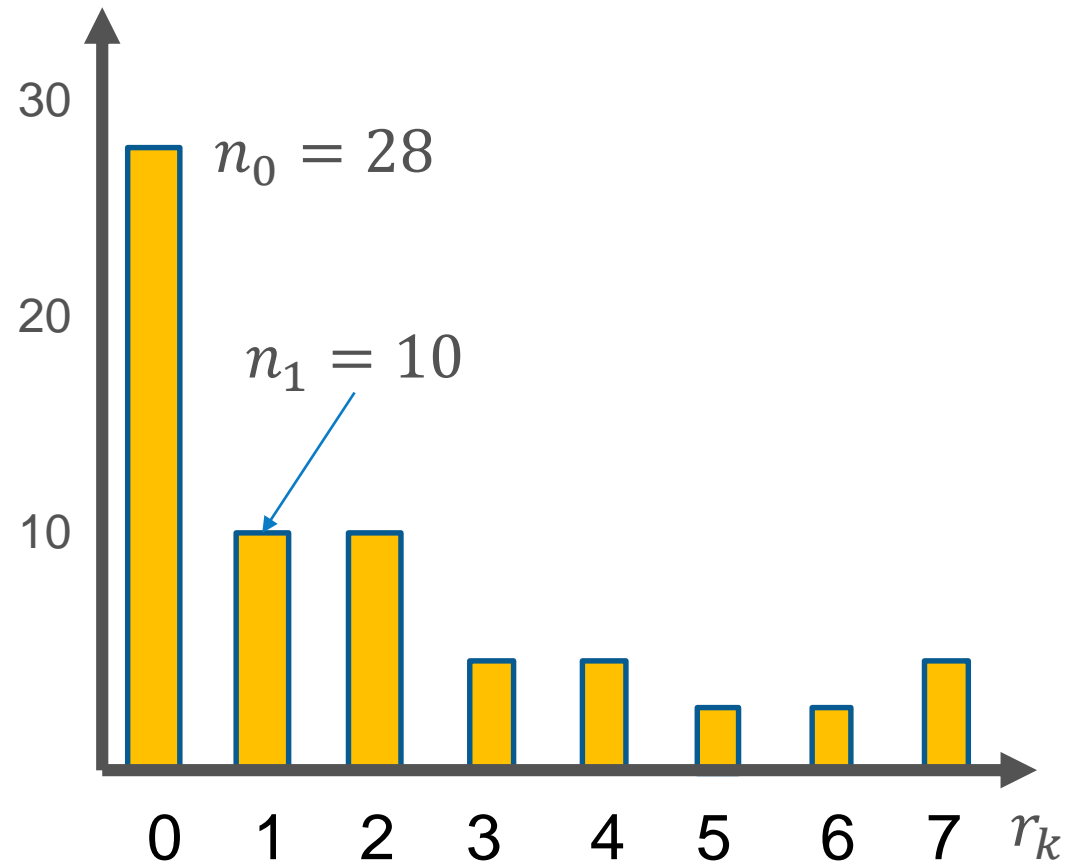


## UMBRALADO MANUAL:

El valor del umbral  $u \in [0, L - 1]$  se puede seleccionar en forma manual como sigue: Sea el siguiente ejemplo:

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

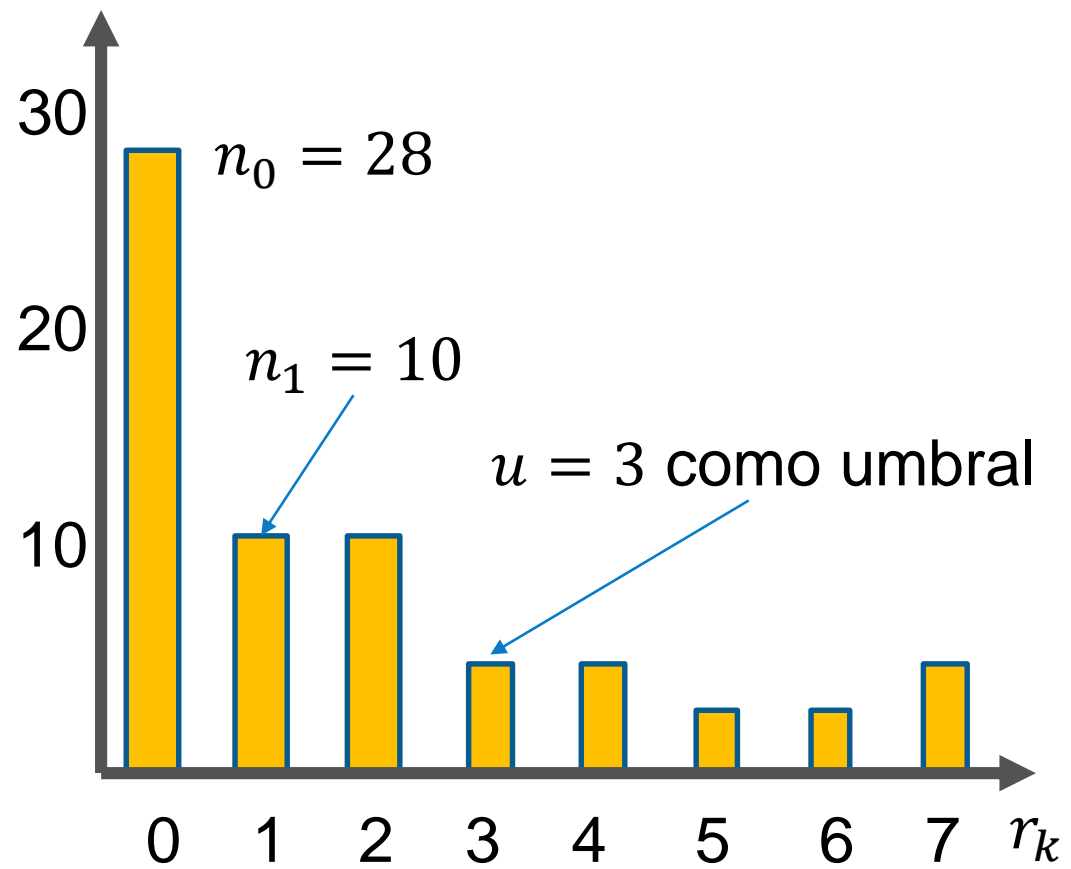
$N = 64$



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




0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



Lo que esté bajo de  $u = 3$  se pone en "0" y todo lo que esté  $u \geq 3$  se pone en  $L - 1$ .

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

  
 $u = 3$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	7	7	7	7	7	7	0
0	7	7	7	7	7	7	0
0	7	7	7	7	7	7	0
0	0	0	0	0	0	0	0

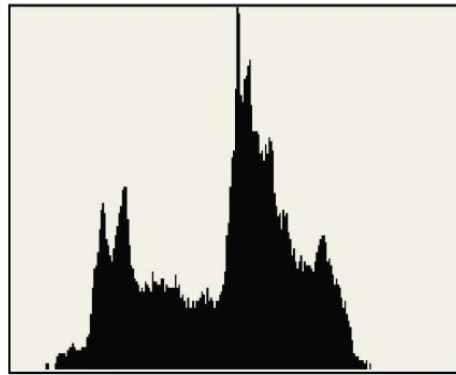
$u = 2$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$u = 4$



(a)

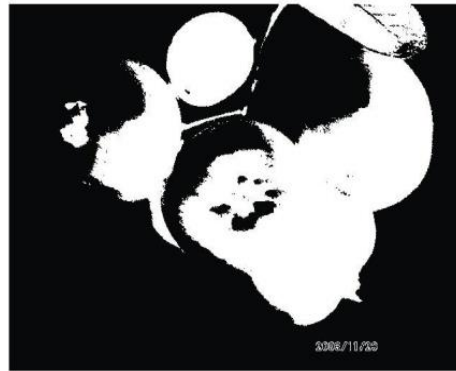


(b)

Resultados de segmentación para tres valores de  $u = 80, 100, 120$  y  $140$ .



(c)



(d)



(e)



(f)

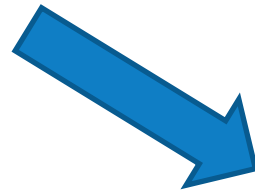
¿Porqué el resultado no es muy bueno?

R: Por que los objetos comparten el mismo color, nivel de intensidad.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	4	4	4	4	1	0
0	1	5	5	5	5	1	0
0	2	5	5	5	5	2	0
0	2	4	4	4	4	2	0
0	2	2	2	2	2	2	0
0	0	0	0	0	0	0	0

¿Qué pasa si se usan dos umbrales?

Por ejemplo, todo lo que esté entre 4 y 5:



0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	7	7	7	7	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$u_1 = 40$$

$$u_2 = 80$$



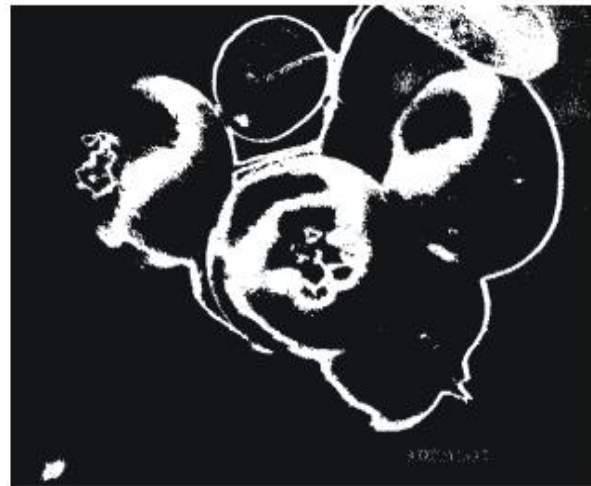
(a)



(b)

$$u_1 = 60$$

$$u_2 = 100$$



(c)

$$u_1 = 80$$

$$u_2 = 120$$

```
% Programa para umbralar manualmente una imagen.

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

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

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

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



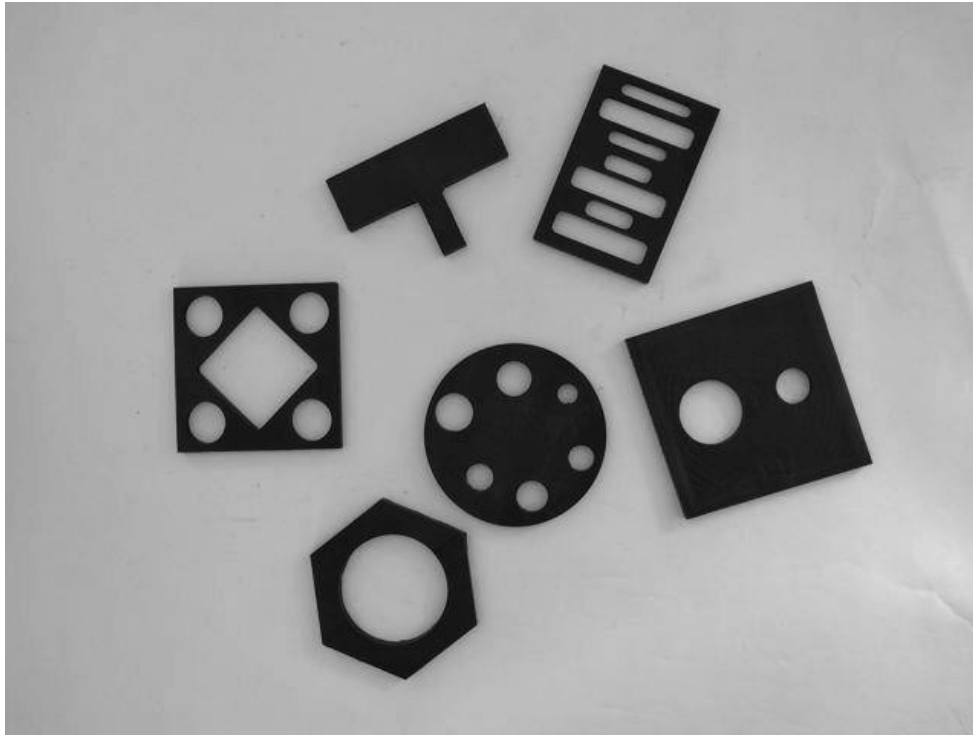
Imagen original



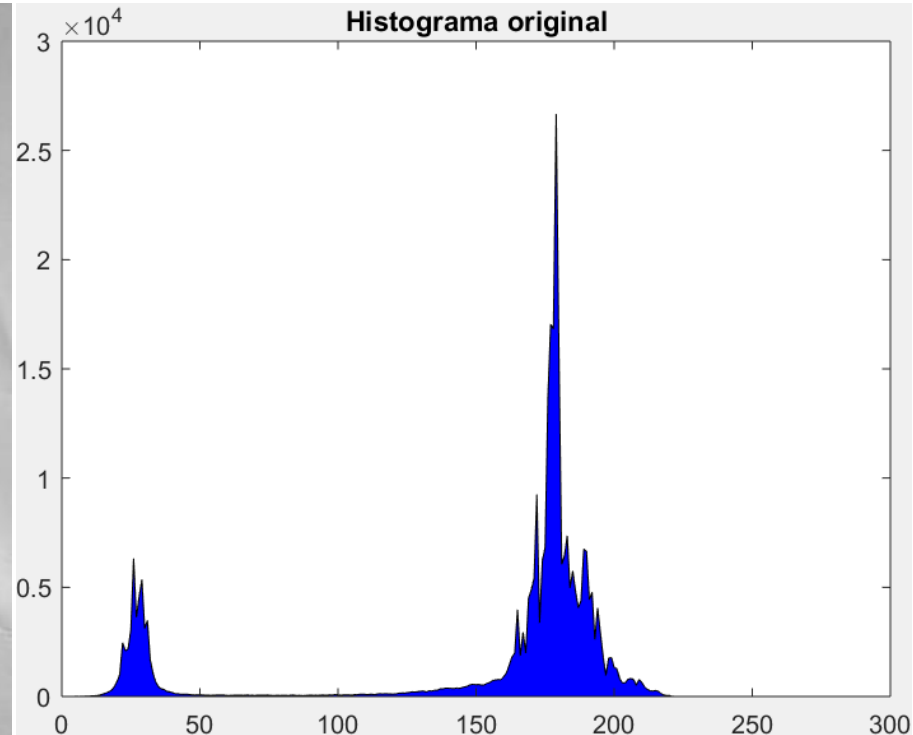
```
c=zeros(256,1); % inicializa en ceros el registro del
contador c(i).
niveles=0:255; % niveles de gris.

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

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



**Imagen original con 6 objetos**



**Histograma de la imagen**

```

th=100; % umbral manual.

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

figure(3) % se abre una ventana auxiliar.
imshow(g) % se muestra imagen f(x,y).
title('Imagen umbralada') % se asigna título de esta
imagen.

```

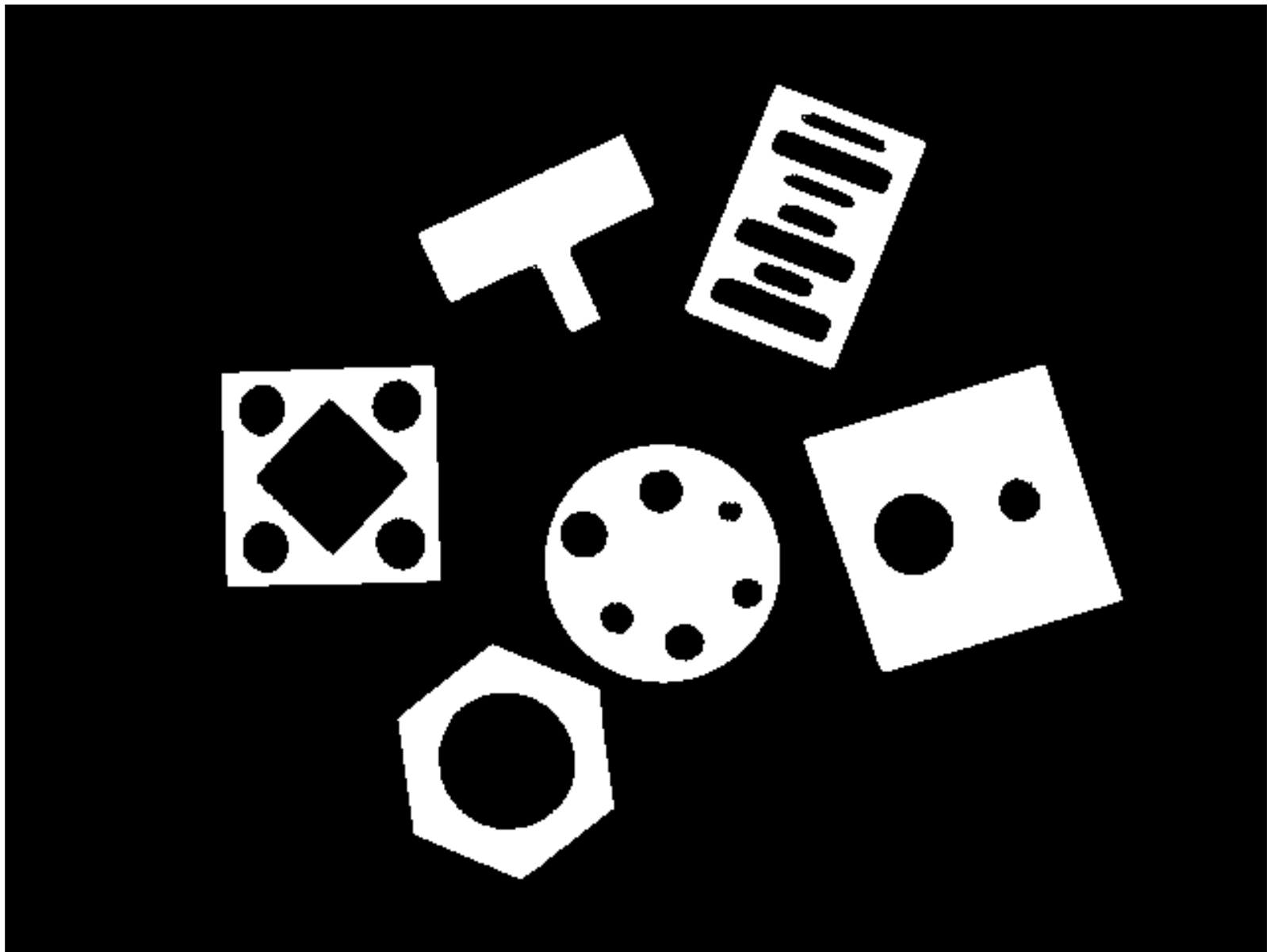


Imagen binaria con  $u = 100$

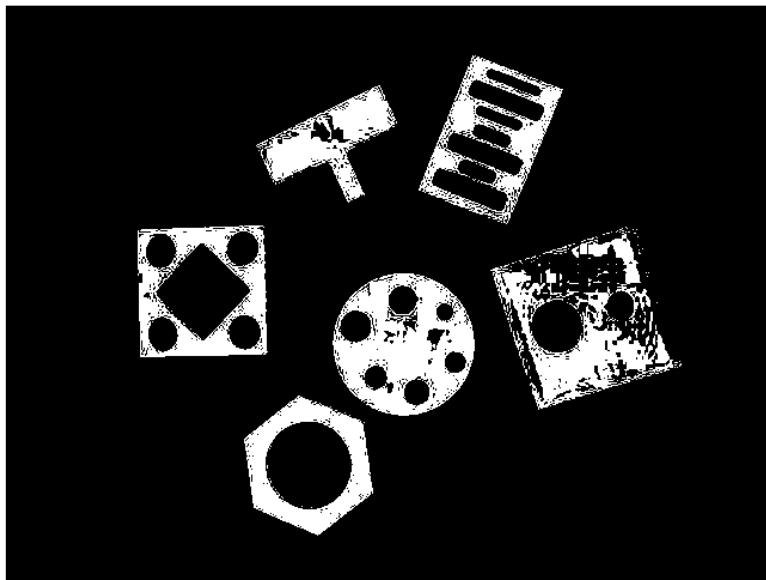


Imagen binaria con  $u = 30$

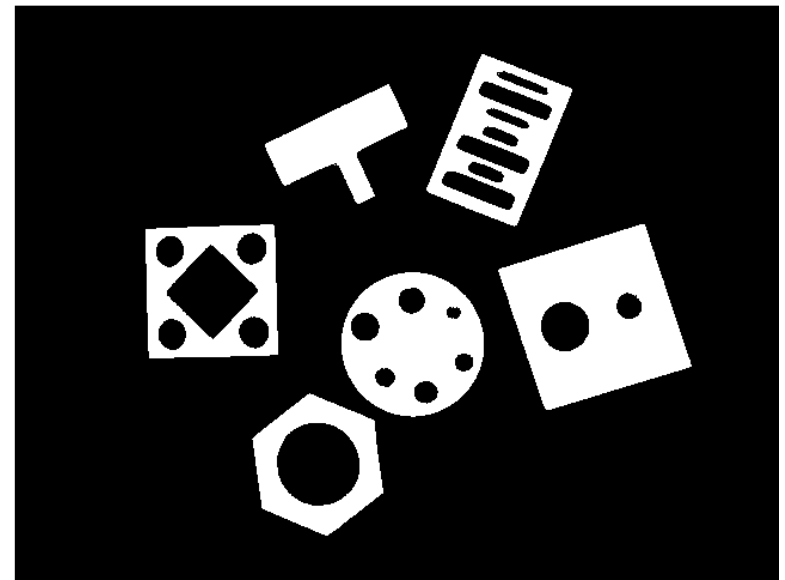


Imagen binaria con  $u = 100$

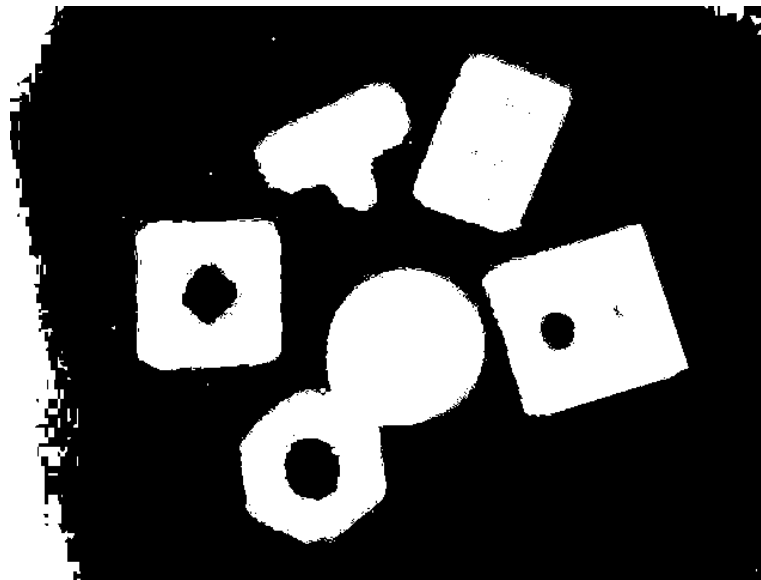


Imagen binaria con  $u = 170$

```

% Programa para umbralizar manualmente una imagen con 2 umbrales.

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

f=imread('Gato 02.JPG'); % se lee imagen de prueba f(x,y)

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

[n,m]=size(f); % dimensiones de la imagen f(x,y)
disp(n); % se muestra valor de n.
disp(m); % se muestra valor de m.
c=zeros(256,1); % inicializa en ceros el registro del contador c(i).
niveles=0:255; % niveles de gris.

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

```

```

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

th1=15 % umbral 1
th2=80; % umbral 2.

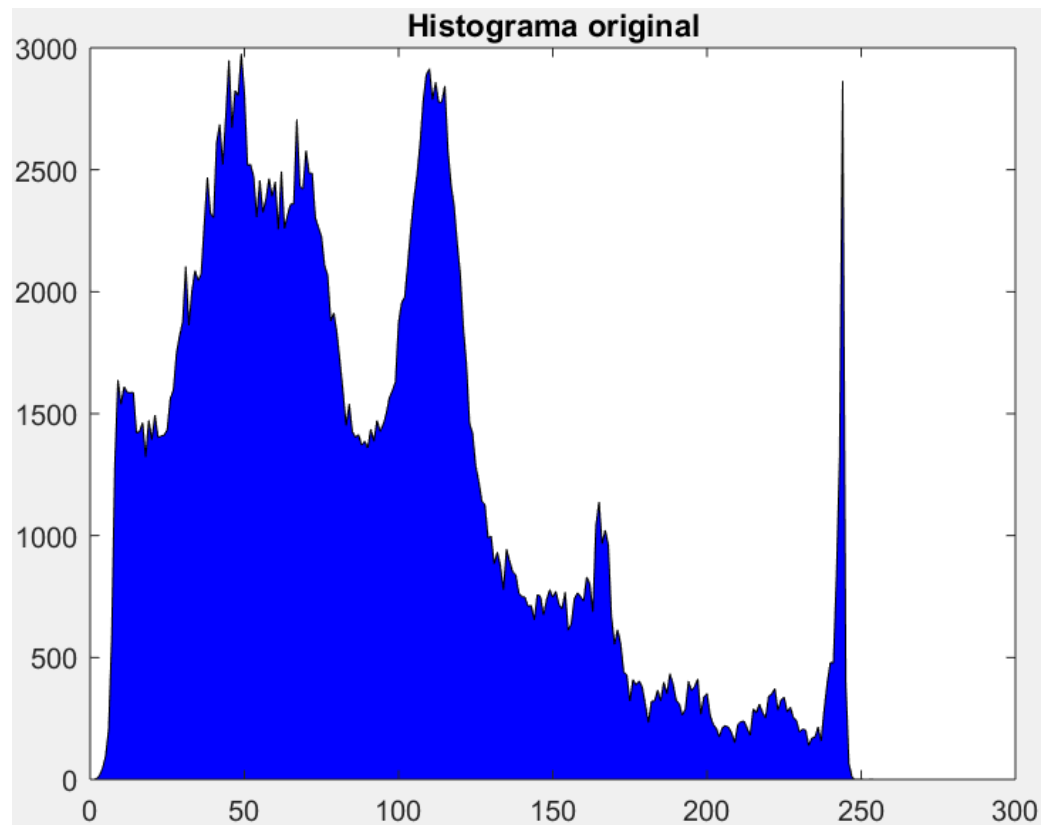
for x=1:n % se barre imagen en renglones.
    for y=1:m % se barre imagen en columnas.
        i=f(x,y); % se obtiene el nivel de gris de f(x;y).
        if i>=th1 && i<=th2
            g(x,y)=0;
        else
            g(x,y)=255;
        end
    end
end

figure(3) % se abre una ventana auxiliar.
imshow(g) % se muestra imagen f(x,y).
title('Imagen umbralada') % se asigna título de esta imagen.

```



**Imagen original de Cleo**



**Histograma de la imagen**





Imagen binaria con  $u1 = 15$  y  $u2 = 80$



**Imagen binaria con  $u1 = 15$  y  $u2 = 80$**



**Imagen binaria con  $u1 = 80$  y  $u1 = 150$**



**Imagen binaria con  $u1 = 150$  y  $u1 = 240$**

**La selección de uno o más umbrales para binarizar una imagen es un problema búsqueda que puede ser visto como un problema de optimización**

# Umbralado Automático:

**En este caso, se busca seleccionar el valor de  $u$  de manera automática.**

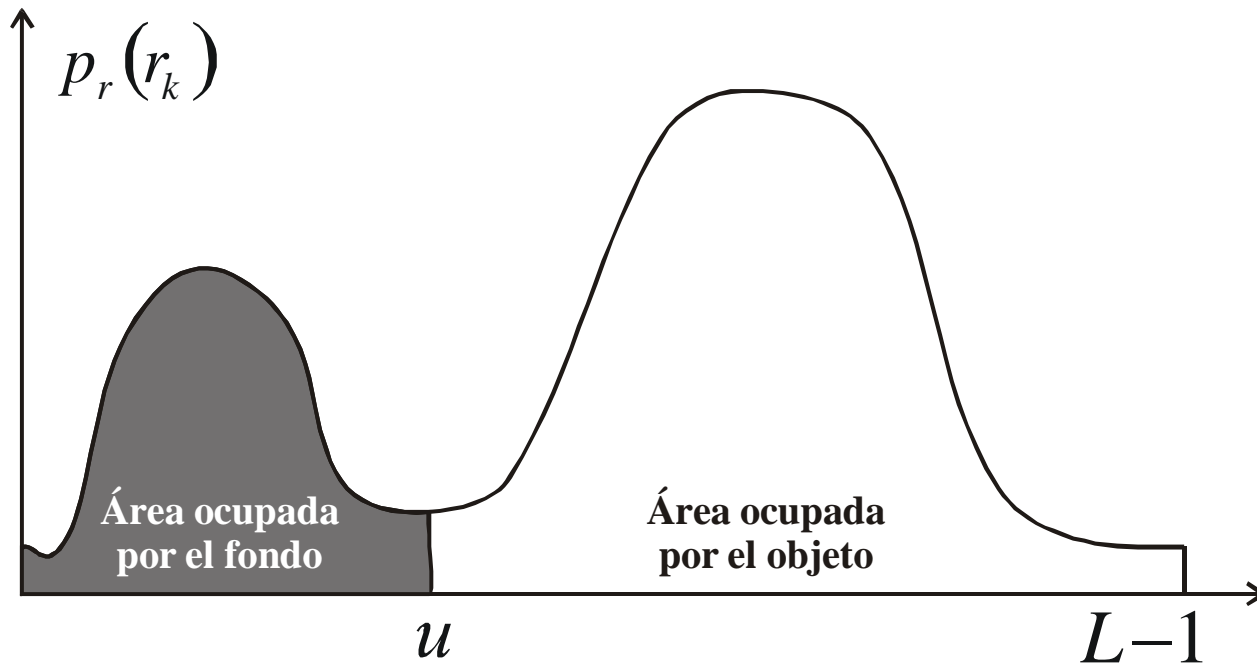
**Analicemos varios métodos:**

## Método P (percentil)-Tile:

Supone que el histograma es bimodal y que cada moda representa a cada clase.

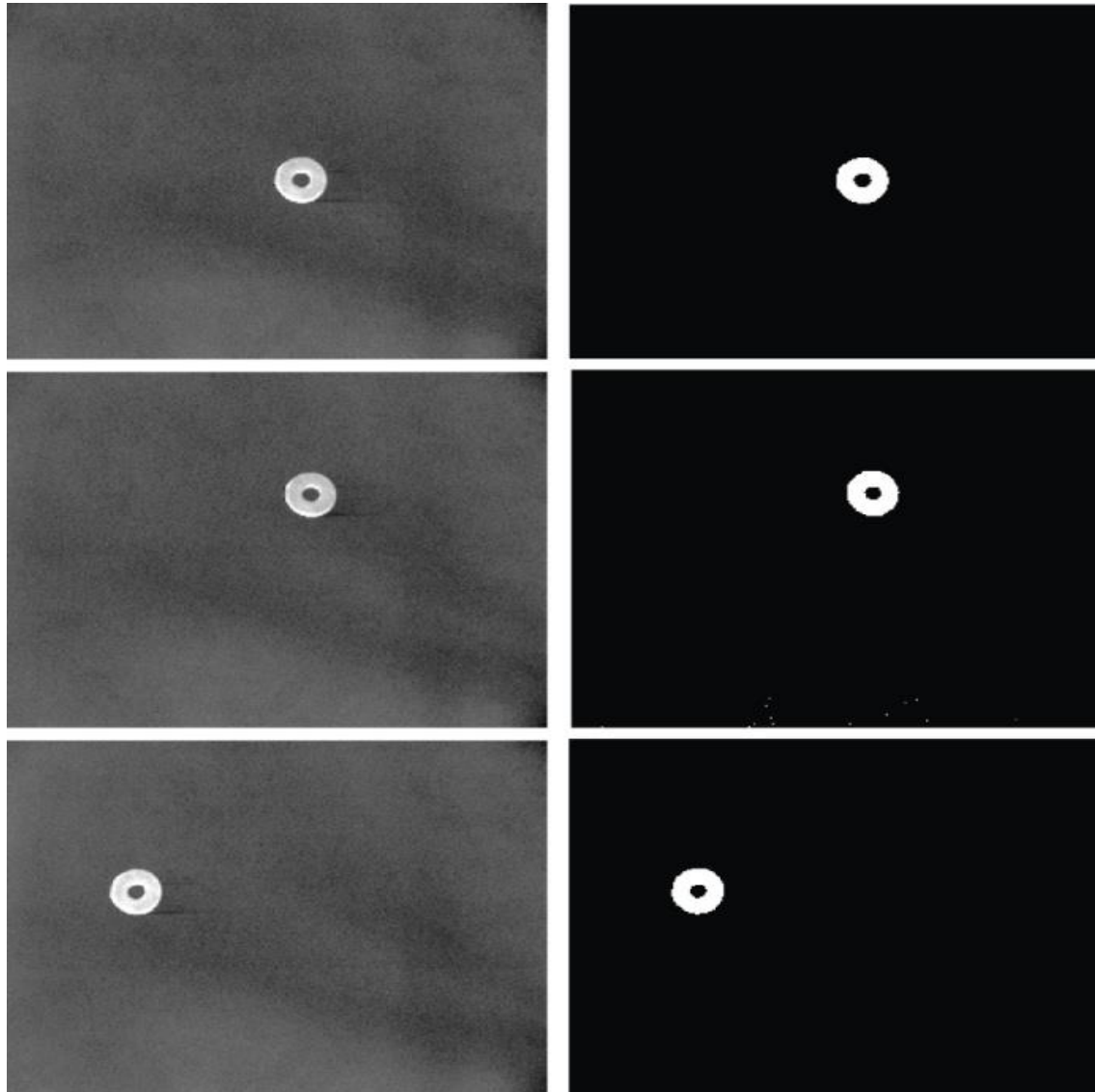
Suponga que conoce el tamaño del objeto en cuestión.

Aplicaciones: en segmentado de texto en hojas.



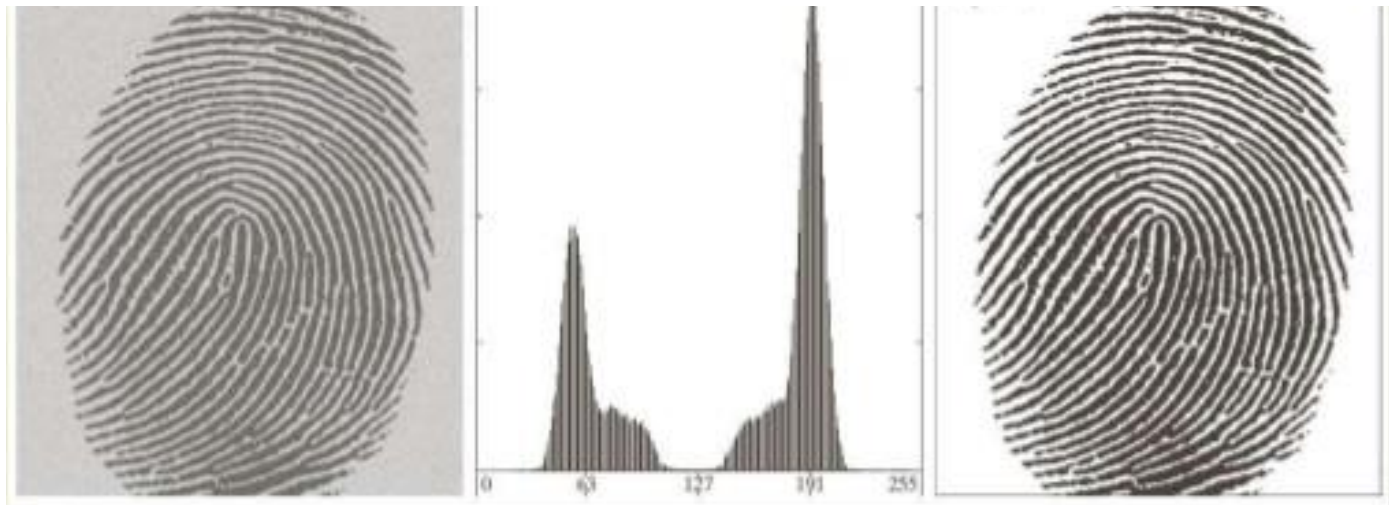
## Método P-Tile:

$u = 115$



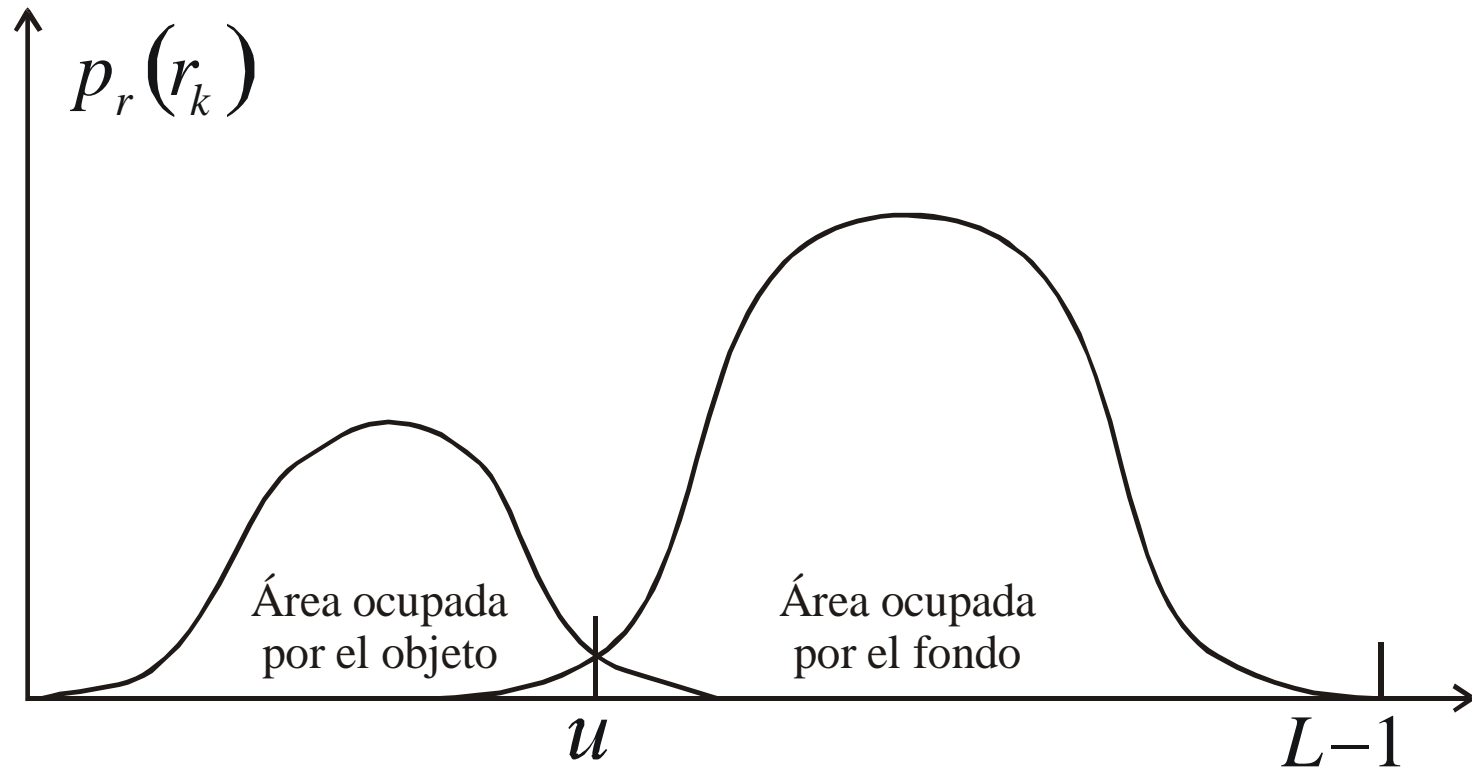


# Método P-Tile:



**FIGURE 10.38** (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border is added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

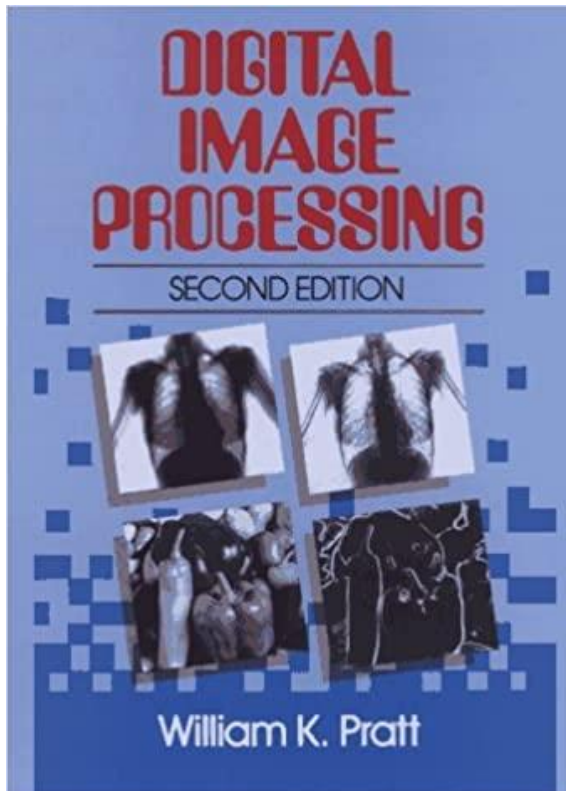
## Método basado en la moda:



# Método iterativo:

1 Seleccionar un umbral inicial, el estimado de  $u$ .

$$u^* = \frac{1}{N \times M} \sum_{x,y}^{N,M} f(x, y)$$



T. W. Ridler and S. Calvard, Picture thresholding using an iterative selection method, IEEE Trans. Syst. Man Cybern. SMC-8, 630–632, 1978.

## Método iterativo:

- 1 Seleccionar un umbral inicial, el estimado de  $u$ .
- 2 Segmentar la imagen en dos grupos de píxeles  $R1$  y  $R2$  en términos de  $u^*$ .

## Método iterativo:

- 1 Seleccionar un umbral inicial, el estimado de  $u$ .
- 2 Segmentar la imagen en dos grupos de píxeles  $R1$  y  $R2$  en términos de  $u^*$ .
- 3 Calcular los valores promedio:  $\mu_1$  y  $\mu_2$  de  $R1$  y  $R2$ .

## Método iterativo:

- 1 Seleccionar un umbral inicial, el estimado de  $u$ .
- 2 Segmentar la imagen en dos grupos de píxeles  $R1$  y  $R2$  en términos de  $u^*$ .
- 3 Calcular los valores promedio:  $\mu_1$  y  $\mu_2$  de  $R1$  y  $R2$ .
- 4 Refinar el valor de  $u^*$ ,  $u^* = \frac{\mu_1 + \mu_2}{2}$

## Método iterativo:

- 1 Seleccionar un umbral inicial, el estimado de  $u$ .
- 2 Segmentar la imagen en dos grupos de píxeles  $R1$  y  $R2$  en términos de  $u^*$ .
- 3 Calcular los valores promedio:  $\mu_1$  y  $\mu_2$  de  $R1$  y  $R2$ .
- 4 Refinar el valor de  $u^*$ ,  $u^* = \frac{\mu_1 + \mu_2}{2}$
- 5 Repetir los pasos 2 a 4 hasta que la **diferencia en  $u$  en iteraciones sucesivas sea menor que un parámetro** predefinido  $\mu_0$ .

## Ejemplo numérico:

0	0	1	0	0	1
0	5	5	6	5	0
0	6	5	7	5	0
0	0	6	5	0	1
1	0	6	7	0	0
0	0	0	0	1	0

$u=2,$



## Ejemplo numérico:

0 0 1 0 0 1		0 0 1 0 0 1
0 5 5 6 5 0		0 5 5 6 5 0
0 6 5 7 5 0		0 6 5 7 5 0
0 0 6 5 0 1	→	0 0 6 5 0 1
1 0 6 7 0 0		1 0 6 7 0 0
0 0 0 0 1 0		0 0 0 0 1 0

$u=2,$                        $u_1=5/24=0.208$

$u_2=68/12=5.66$

(paso 4):  $u^*=2.93\sim 3$

Dif:               $3-2=1$

## Ejemplo numérico:

0 0 1 0 0 1	→	0 0 1 0 0 1	→	0 0 1 0 0 1	→
0 5 5 6 5 0		0 5 5 6 5 0		0 5 5 6 5 0	
0 6 5 7 5 0		0 6 5 7 5 0		0 6 5 7 5 0	
0 0 6 5 0 1		0 0 6 5 0 1		0 0 6 5 0 1	
1 0 6 7 0 0		1 0 6 7 0 0		1 0 6 7 0 0	
0 0 0 0 1 0		0 0 0 0 1 0		0 0 0 0 1 0	
u=2,		u <sub>1</sub> =5/24=0.208		Segmentación	
		u <sub>2</sub> =68/12=5.66		u <sub>1</sub> =5/24=0.208	
(paso 4): u*=2.93~3				u <sub>2</sub> =68/12=5.66	
Dif: 3-2=1		Dif: 3-3=0		u*=2.93~3	

## Ejemplo numérico:

0 0 1 0 0 1	→	0 0 1 0 0 1	→	0 0 1 0 0 1	→	0 0 0 0 0 0
0 5 5 6 5 0		0 5 5 6 5 0		0 5 5 6 5 0		0 1 1 1 1 0
0 6 5 7 5 0		0 6 5 7 5 0		0 6 5 7 5 0		0 1 1 1 1 0
0 0 6 5 0 1		0 0 6 5 0 1		0 0 6 5 0 1		0 0 1 1 0 0
1 0 6 7 0 0		1 0 6 7 0 0		1 0 6 7 0 0		0 0 1 1 0 0
0 0 0 0 1 0		0 0 0 0 1 0		0 0 0 0 1 0		0 0 0 0 0 0
				Segmentación		Imagen

binaria

$u=2$ ,

$$u_1 = 5/24 = 0.208$$

$$u_1 = 5/24 = 0.208$$

$$u_2 = 68/12 = 5.66$$

$$u_2 = 68/12 = 5.66$$

(paso 4):  $u^* = 2.93 \sim 3$

$u^* = 2.93 \sim 3$

Dif:  $3 - 2 = 1$

Dif:  **$3 - 3 = 0$**

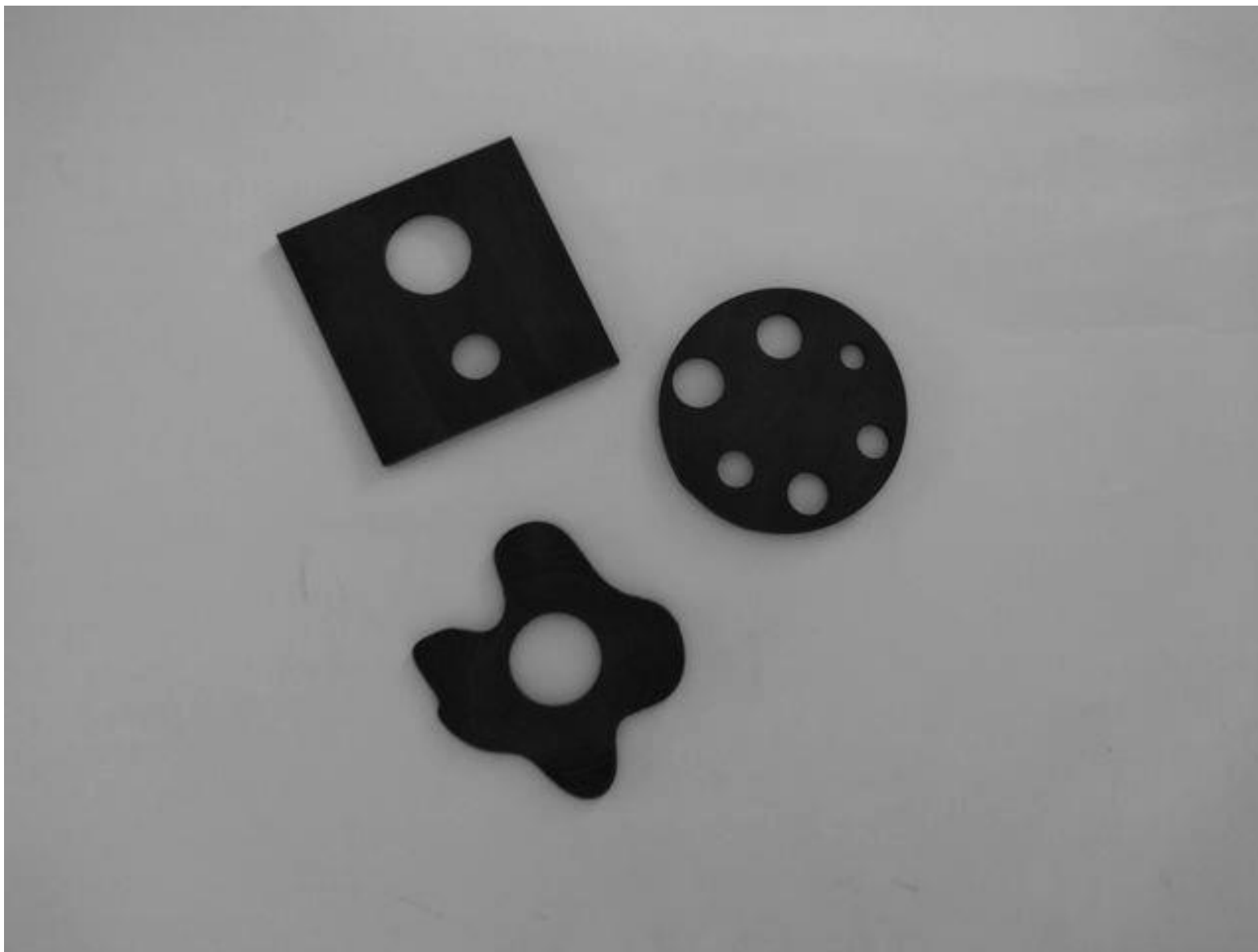
% Programa para umbralar imagen mediante **método iterado.**

```
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)  
disp(n); % se muestra valor de n.  
disp(m); % se muestra valor de m.
```



**Imagen con 3 objetos**

```

% se obtiene el nivel de gris promedio de f(x,y).

val=0;
for x=1:n
    for y=1:m
        val=val+double(f(x,y));
    end
end

th=val/(n*m)

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

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

```

```

u1=0; u2=0;
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).
        if i==0
            u1=u1+double(f(x,y));
        else
            u2=u2+double(f(x,y));
        end
    end
end

u1=u1/(n*m)
u2=u2/(n*m)
u_nuevo=(u1+u2)/2

```

```

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

figure(3) % se abre una ventana auxiliar.
imshow(g) % se muestra imagen f(x,y).
title('Imagen umbralada') % se asigna título de esta imagen.

```



# Aplicación del método de umbralado iterado hasta la segunda iteración:

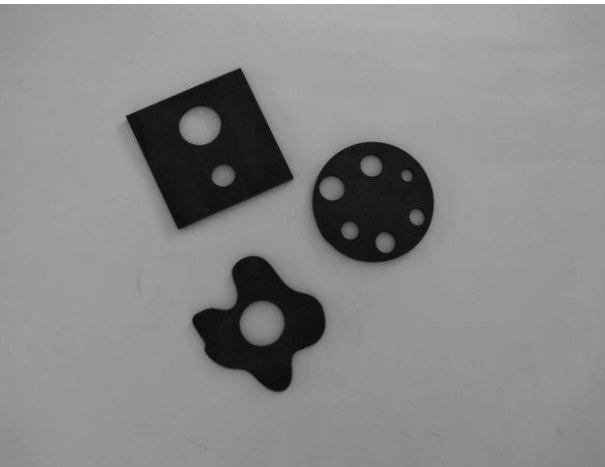
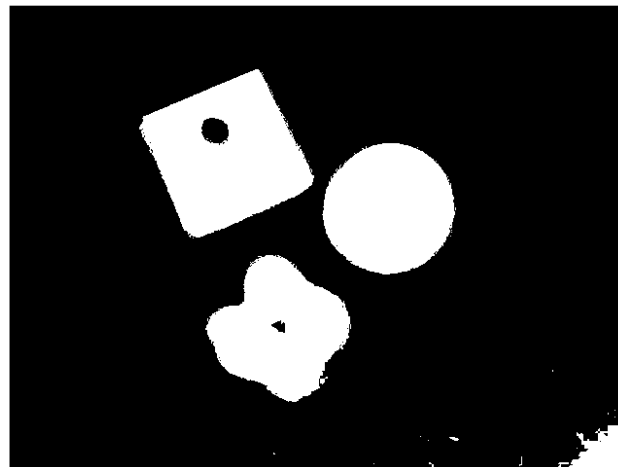
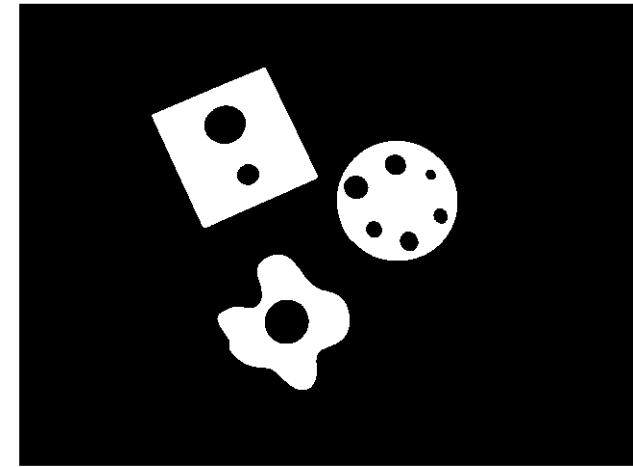


Imagen con 3 objetos



Binaria con  $u = 127.8$



Binaria con  $u = 68.04$

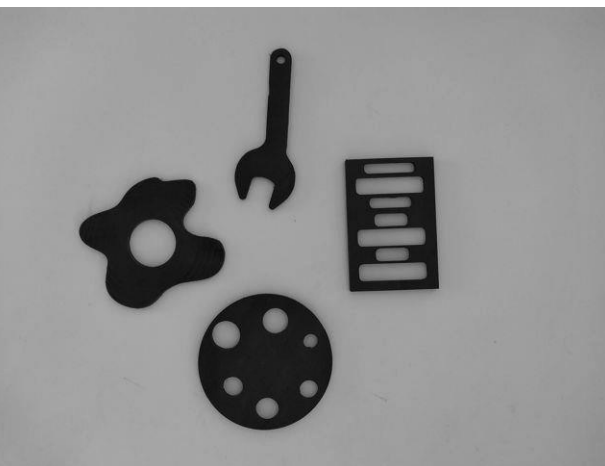
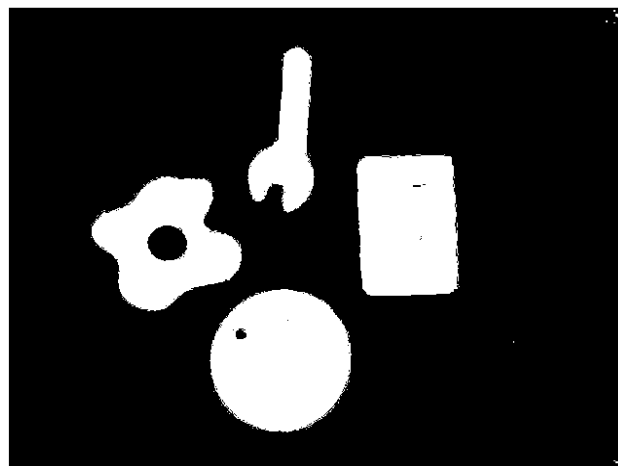
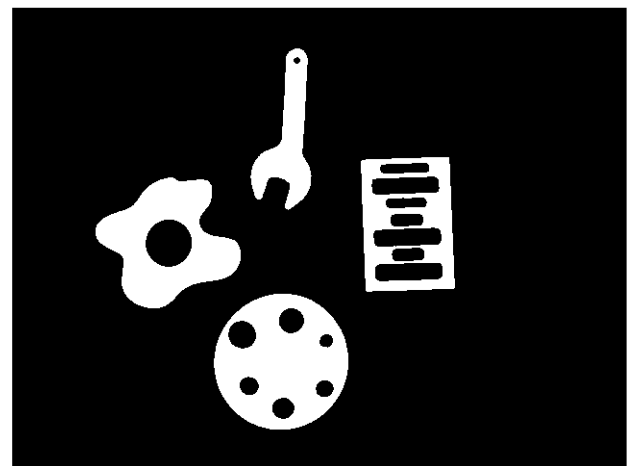


Imagen con 4 objetos



Binaria con  $u = 141.65$



Binaria con  $u = 70.83$

## Método iterativo:

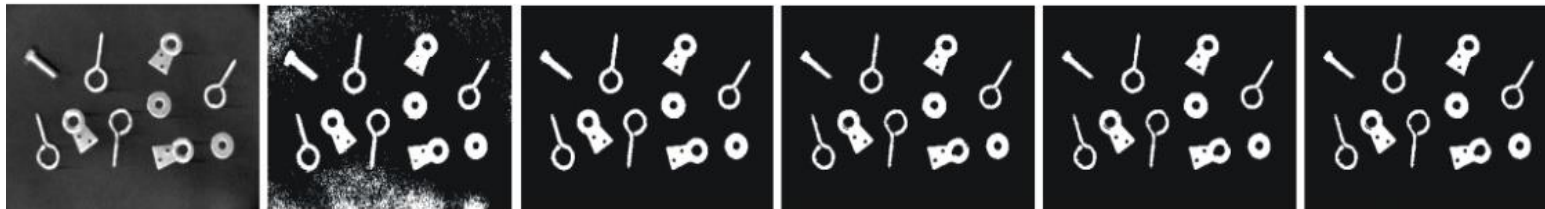
¿Bajo qué condiciones este método converge a una solución?



Primera imagen



Segunda imagen



Tercera imagen

1ra imagen:  
66, 88, 120 y  
133.

2da imagen:  
82, 122 y 131.

3ra imagen:  
65, 95, 123,  
127 y 128.

# Método de Otsu (1979):

Separar los píxeles en dos clases:  $C_1$  y  $C_2$ .

$$J_1(u) = \frac{P_1(u)P_2(u)[\mu_1(u) - \mu_2(u)]^2}{P_1(u)\sigma_1^2(u) + P_2(u)\sigma_2^2(u)}$$

$$P_1(u) = P_r(C_1) = \sum_{r=0}^u p(r)$$

$$P_2(u) = P_r(C_2) = \sum_{r=u+1}^{L-1} p(r) = 1 - P_1(u)$$

$$\mu_1(u) = \sum_{r=0}^u rP_r(r | C_1) = \frac{1}{P_1(u)} \sum_{r=0}^u rp(r)$$

$$\begin{aligned}\sigma_1^2(u) &= \sum_{r=0}^u (r - \mu_1(u))^2 P_r(r | C_1) \\ &= \frac{1}{P_1(u)} \sum_{r=0}^u (r - \mu_1(u))^2 p(r)\end{aligned}$$

$$\mu_2(u) = \sum_{r=u+1}^{L-1} rP_r(r | C_2) = \frac{1}{P_2(u)} \sum_{r=u+1}^{L-1} rp(r)$$

$$\begin{aligned}\sigma_2^2(u) &= \sum_{r=u+1}^{L-1} (r - \mu_2(u))^2 P_r(r | C_2) \\ &= \frac{1}{P_2(u)} \sum_{r=u+1}^{L-1} (r - \mu_2(u))^2 p(r)\end{aligned}$$

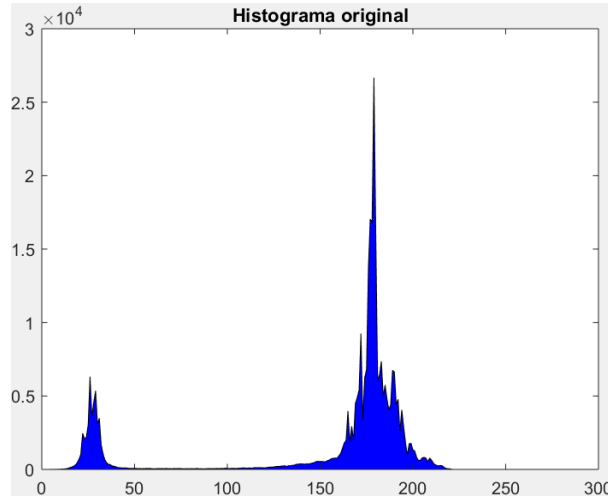
# Procedimiento general para umbralar una imagen:

1. Obtener histograma de la imagen.
2. Recorrer el histograma y calcular las probabilidades y desviaciones estándar de cada clase y aplican la ecuación  $J_{otsu}(u)$  para formar un vector.
3. Buscan el máximo o el mínimo y ahí donde se encuentre este máximo o mínimo recuperar el índice del vector. Este índice es el  $u$  deseado.

N. Otsu, A threshold selection method from gray level histograms, IEEE Trans. Syst. Man Cybern. SMC-9,62–66, 1979.

# Método de Otsu:

Para poder maximizar el criterio dado por la ecuación (3.4), las medias de las dos clases deberían estar bastante bien separadas y las varianzas deberían ser lo más pequeñas posibles.



Si esto no sucede, el valor del umbral obtenido simplemente no producirá el resultado deseado.

Una imagen con un fondo muy grande comparado con el objeto u objetos en la imagen puede también dar lugar a valores de umbral que produzcan resultados indeseados.

## Método de Otsu:

El valor óptimo de  $u^*$ , en el rango  $[0, L - 1]$ :  $u^* = \arg \max_{0 \leq u \leq L-1} J_1(r)$

Otras funcionales:

$$J_2(u) = \frac{\sigma_1}{P_1(u)\sigma_1^2(u) + P_2(u)\sigma_2^2(u)}$$

$$J_3(u) = \frac{P_1(u)P_2(u)[\mu_1(u) - \mu_2(u)]^2}{\sigma^2}$$

$$\sigma^2 = \sum_{r=0}^{l-1} (r - \mu)^2 p(r)$$

$$\mu = \sum_{r=0}^{L-1} rp(r) = P_1(r)\mu_1(r) + P_2(r)\mu_2(r)$$

## Método de Otsu:

Vamos a usar la siguiente funcional:

$$J_B(u) = P_1(u)(\mu_1(u) - \mu)^2 + P_2(u)(\mu_2(u) - \mu)^2$$

$$u^* = \arg \max \{J_B(u)\}$$

```
% Programa para umbralar imagen mediante método de Otsu.

clc; % limpia pantalla
clearvars; % remueve todas las variables del espacio actual de
trabajo
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)
disp(n); % se muestra valor de n.
disp(m); % se muestra valor de m.
```



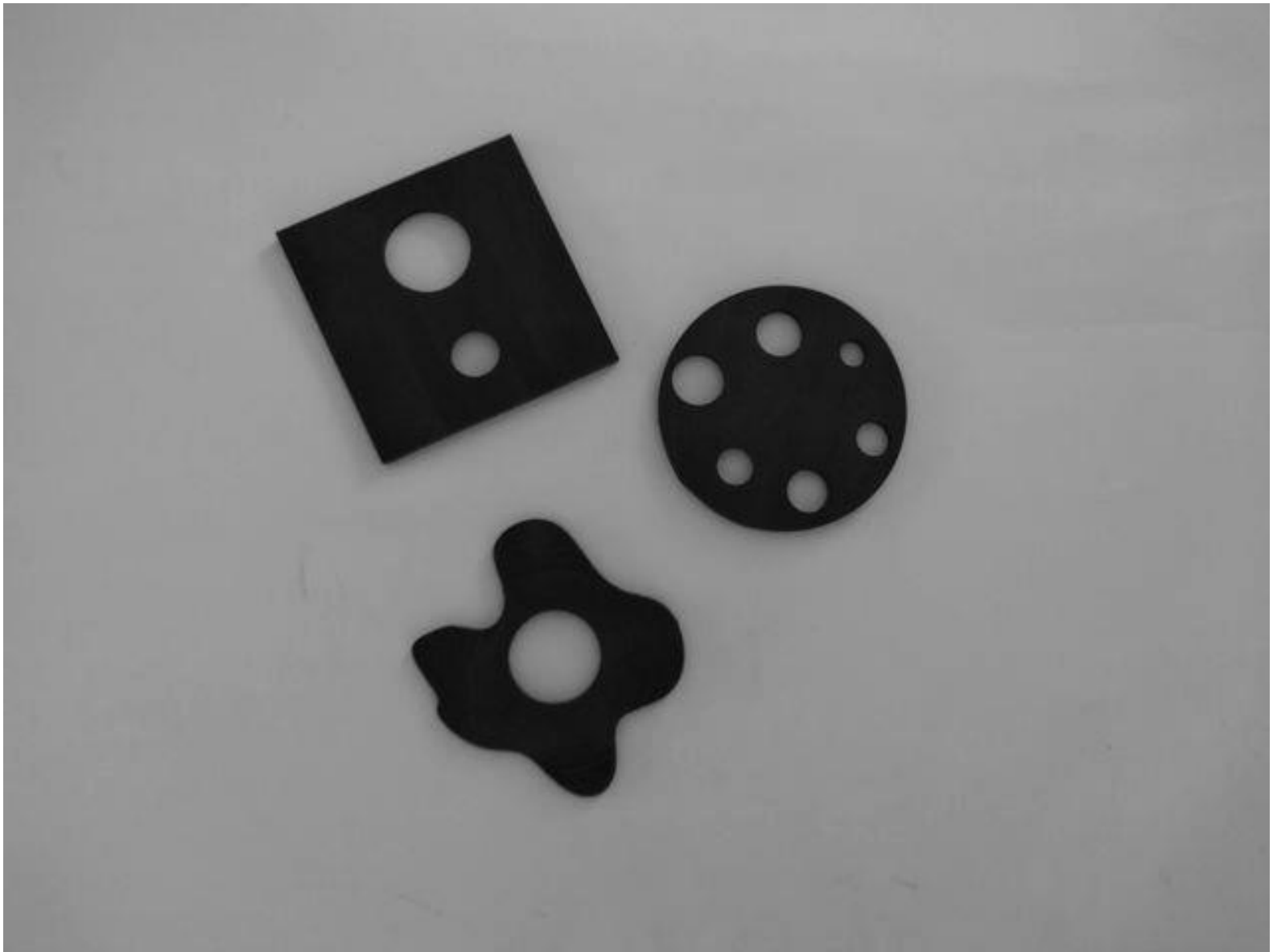


Imagen con 3 objetos

```

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

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

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

c_norm=c/(n*m); % histograma normalizado.

```

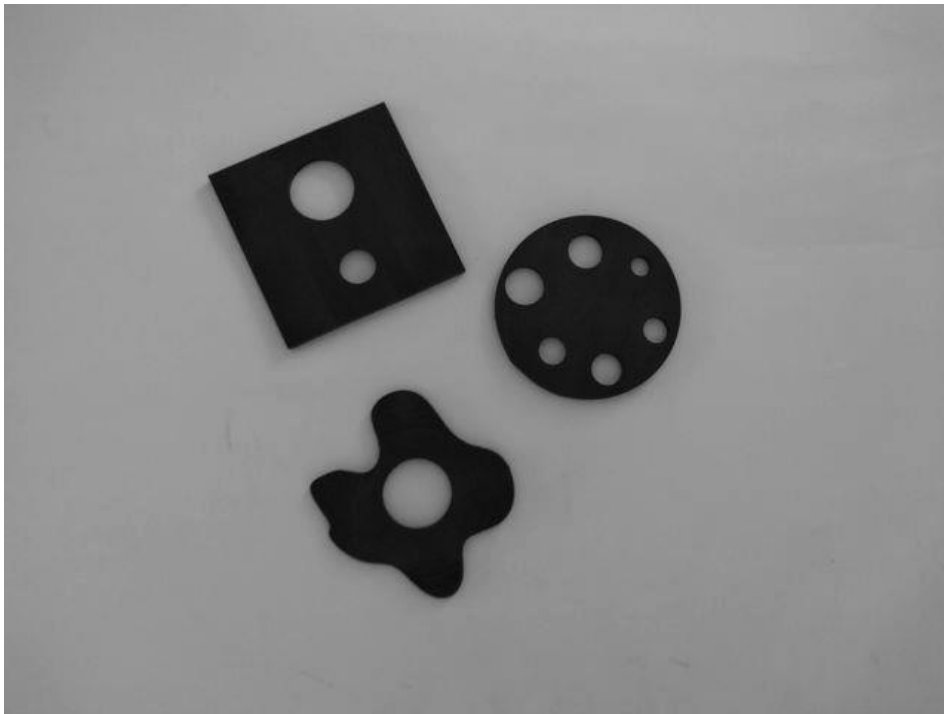
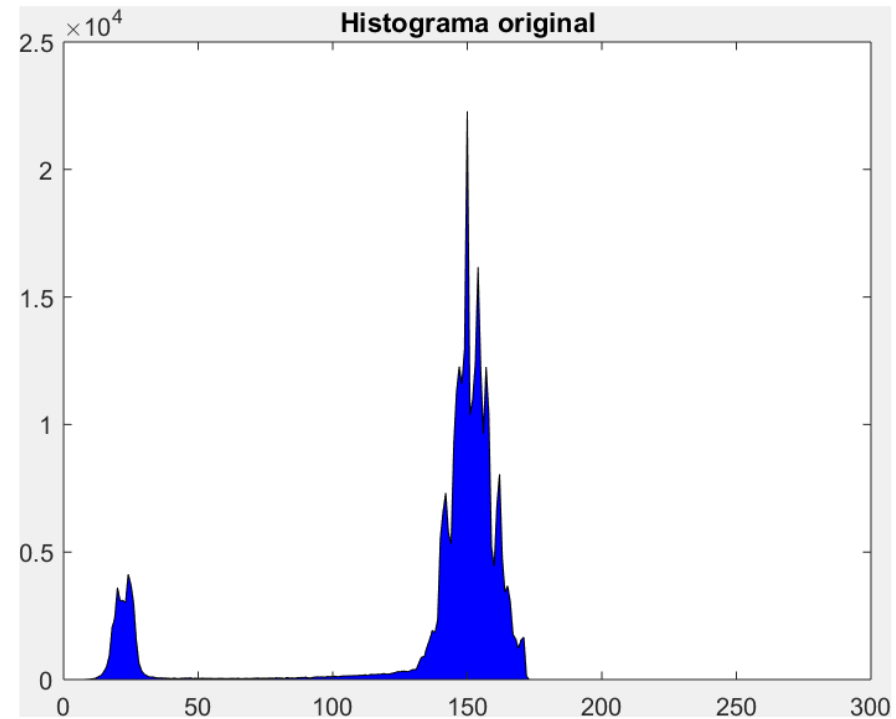


Imagen con 3 objetos



su histograma

```
p1=0; p2=0; u1=0; u2=0;
```

```
for i=1:256
    for k=1:i
        p1=p1+c_norm(k);
    end
    for k=1:i
        u1=u1+(1/p1)*k*c_norm(k);
    end
    p2=1-p1;
    for j=i+1:256
        u2=u2+(1/p2)*j*c_norm(j);
    end
    u_total=p1*u1+p2*u2;
    sigma(i)=p1*(u1-u_total)*(u1-u_total)+p2*(u2-u_total)*(u2-
u_total);
    p1=0;
    p2=0;
    u1=0;
    u2=0;
end
```

```

[M,th]=max(sigma) % M contiene el valor máximo y th el umbral.

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

figure(3) % se abre una ventana auxiliar.
imshow(g) % se muestra imagen f(x,y).
title('Imagen umbralada') % se asigna título de esta imagen.

```

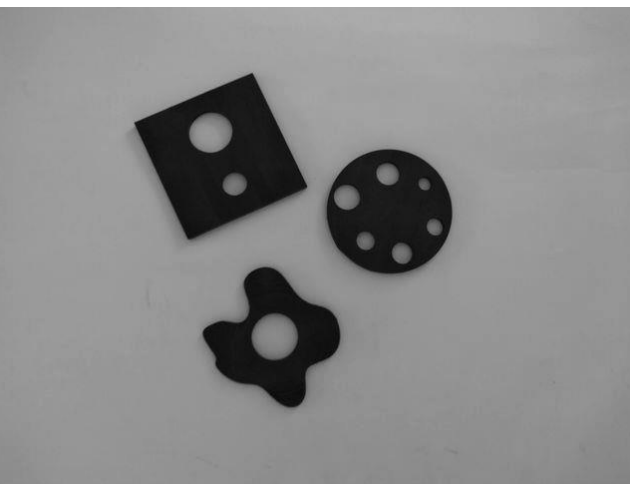
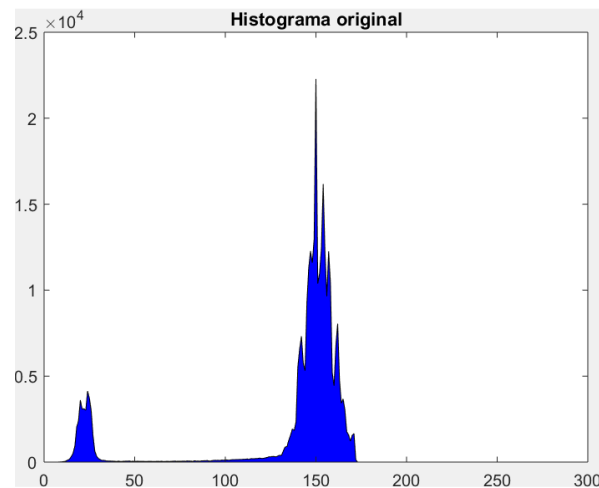
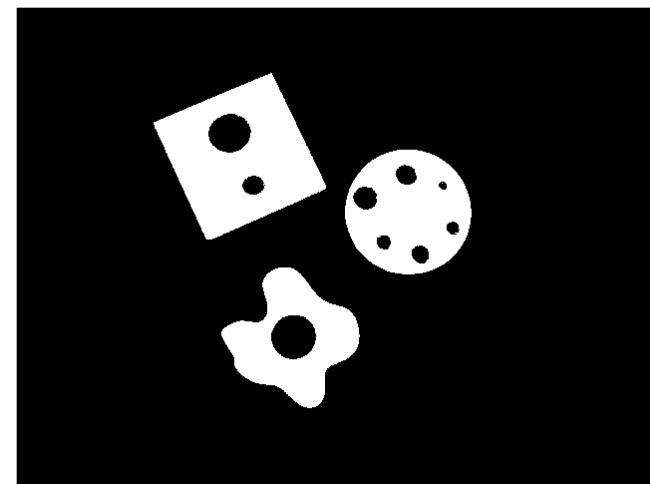


Imagen con 3 objetos



su histograma



umbralada con  $u = 89$

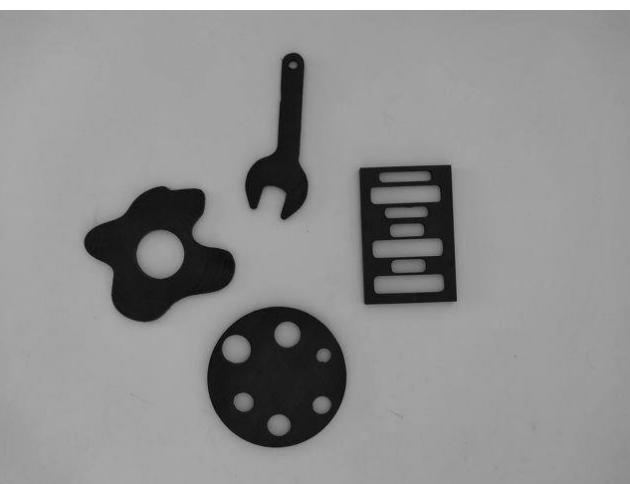
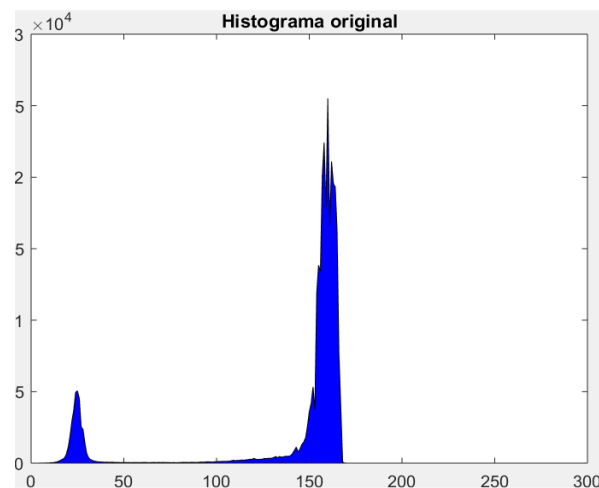
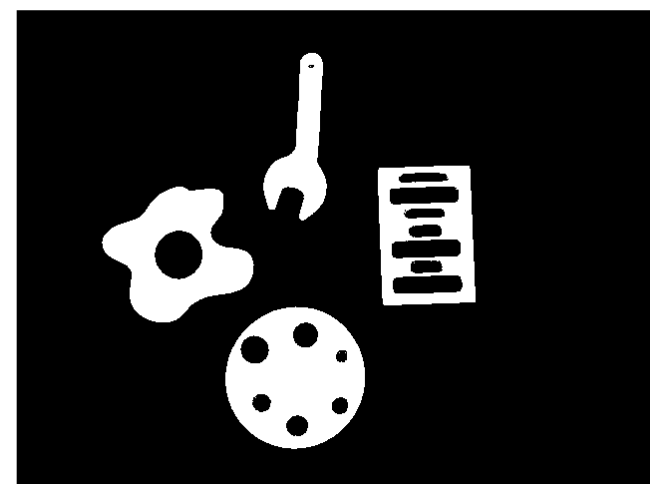


Imagen con 4 objetos

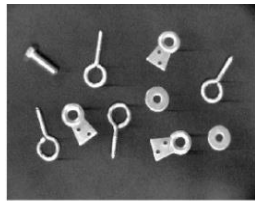


su histograma

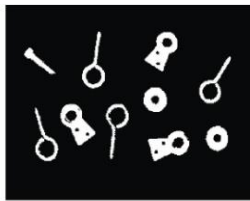


umbralada con  $u = 93$

# Método de Otsu:

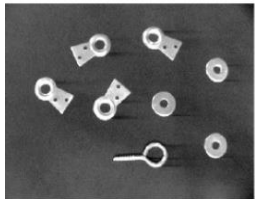


(a)



(f)

128

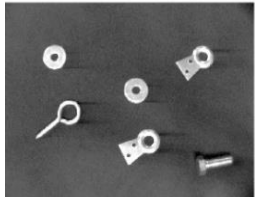


(b)



(g)

126

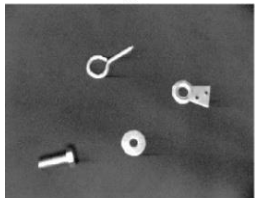


(c)



(h)

137

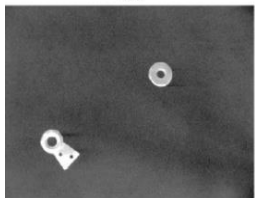


(d)

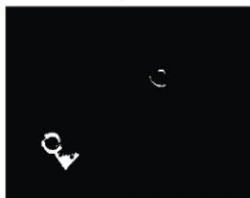


(i)

138



(e)



(j)

254

¿Por qué aquí falló?

## Método de mínimo error de Kittler e Illinworth (1986):

$$J_{KI}(u) = \sum_{r=0}^{L-1} p(r)c_{KI}(r,u) = \sum_{r=0}^u p(r)c_{KI}^1(r,u) + \sum_{r=u+1}^{L-1} p(r)c_{KI}^2(r,u)$$

Donde  $c_{KI}^1(r,u)$  y  $c_{KI}^2(r,u)$  pueden ser consideradas como dos partes de la función de costo definida como:

$$c_{KI}(r,u) = \begin{cases} c_{KI}^1(r,u) & \text{si } r \leq u \\ c_{KI}^2(r,u) & \text{si } r > u \end{cases}$$

J. Kittler and J. Illingworth, Minimum error thresholding, Pattern Recognition 19:41–47, 1986.



## Método de mínimo error de Kittler e Illinworth:

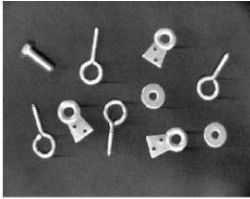
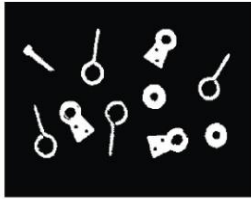
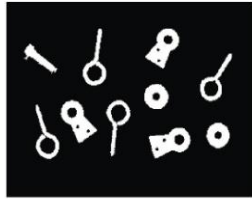
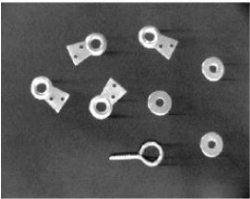
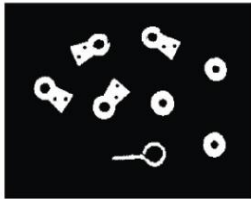
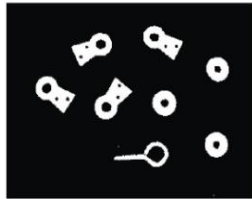
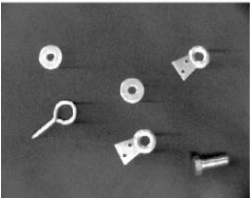


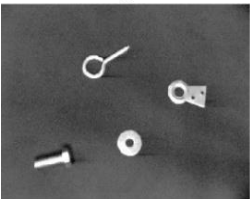


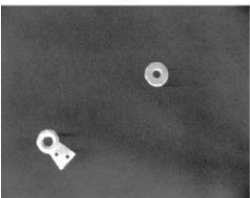
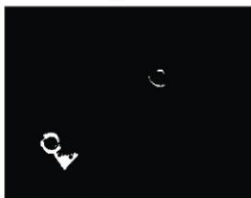

$$c_{KI}(r, u) = \begin{cases} c_{KI}^1(r, u) = \frac{(r - \mu_1(u))^2}{\sigma_1^2(u)} + 2 \ln \sigma_1(u) - 2 \ln P_1(u) & \text{si } r \leq u \\ c_{KI}^2(r, u) = \frac{(r - \mu_2(u))^2}{\sigma_2^2(u)} + 2 \ln \sigma_2(u) - 2 \ln P_2(u) & \text{si } r > u \end{cases}$$

$$J_{KI}(u) = 1 + 2[P_1(u) \ln \sigma_1(u) + P_2(u) \ln \sigma_2(u)] \\ - 2[P_1(u) \ln P_1(u) + P_2(u) \ln P_2(u)]$$

$$u^* = \arg \min_{0 \leq u \leq L-1} J_{KI}(r)$$

**Cuidado con los logaritmos, ya que si los argumentos son “0”, los logaritmos se indeterminan**

# Método de mínimo error de Kittler e Illinworth:

			128	86
(a)	(f)	(k)		
			126	87
(b)	(g)	(l)		
			137	97
(c)	(h)	(m)		
			138	98
(d)	(i)	(n)		
			254	101
(e)	(j)	(o)		

```
% Programa para umbralar imagen mediante método de Kitler.

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

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

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

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

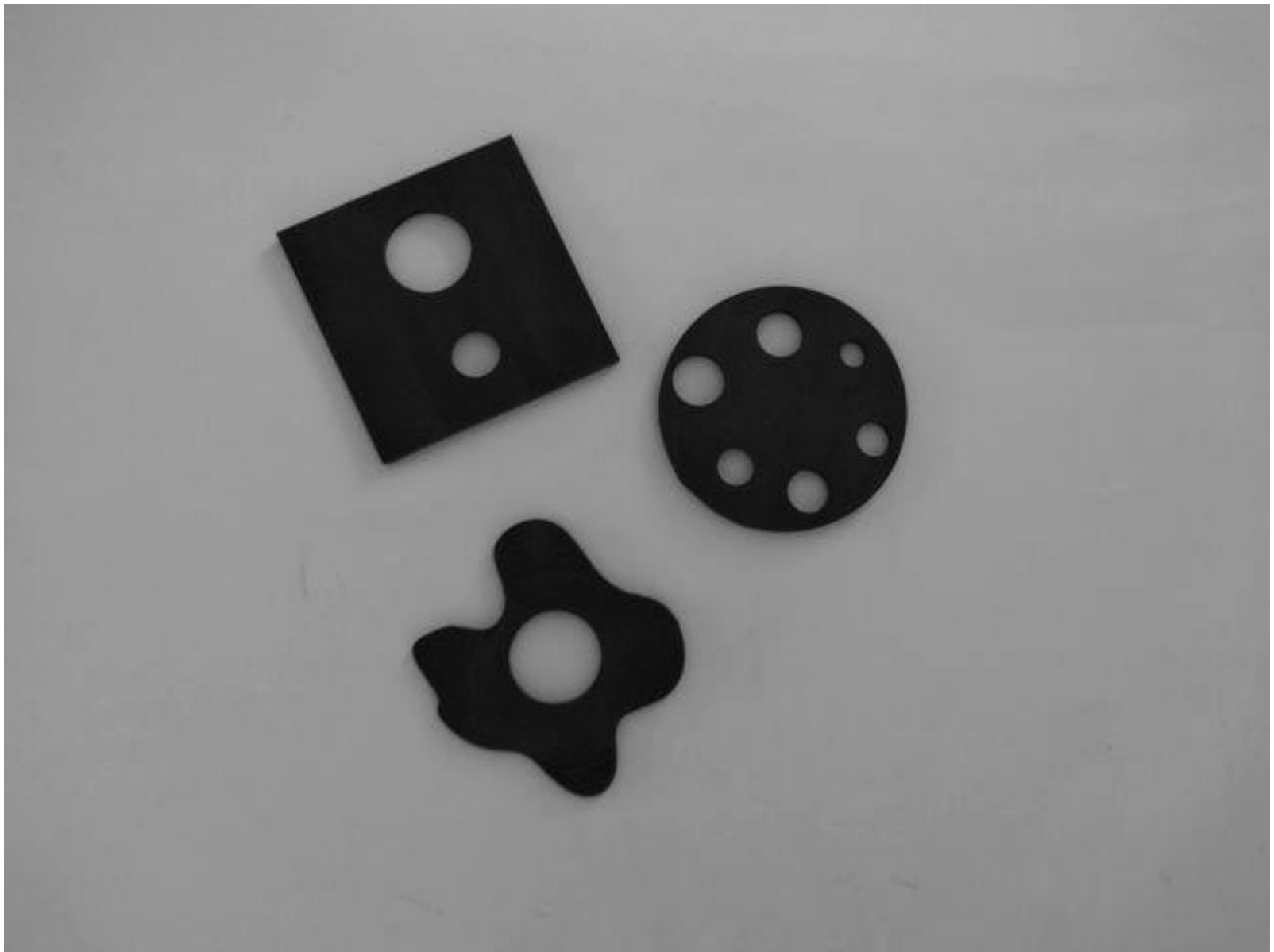


Imagen con 3 objetos

```

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

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

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

c_norm=c/(n*m); % histograma normalizado.

```

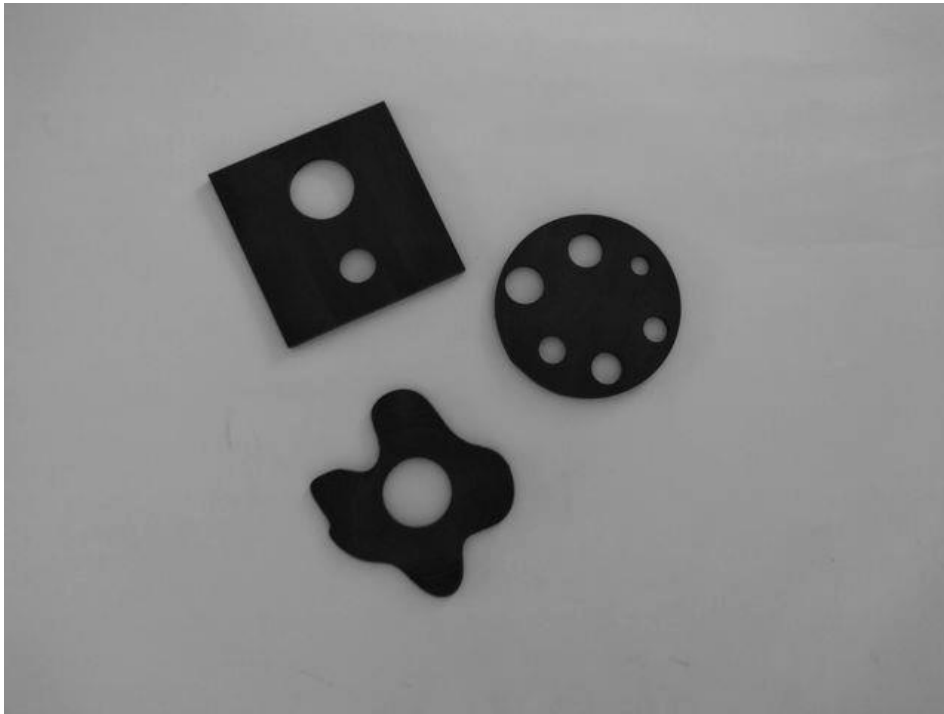
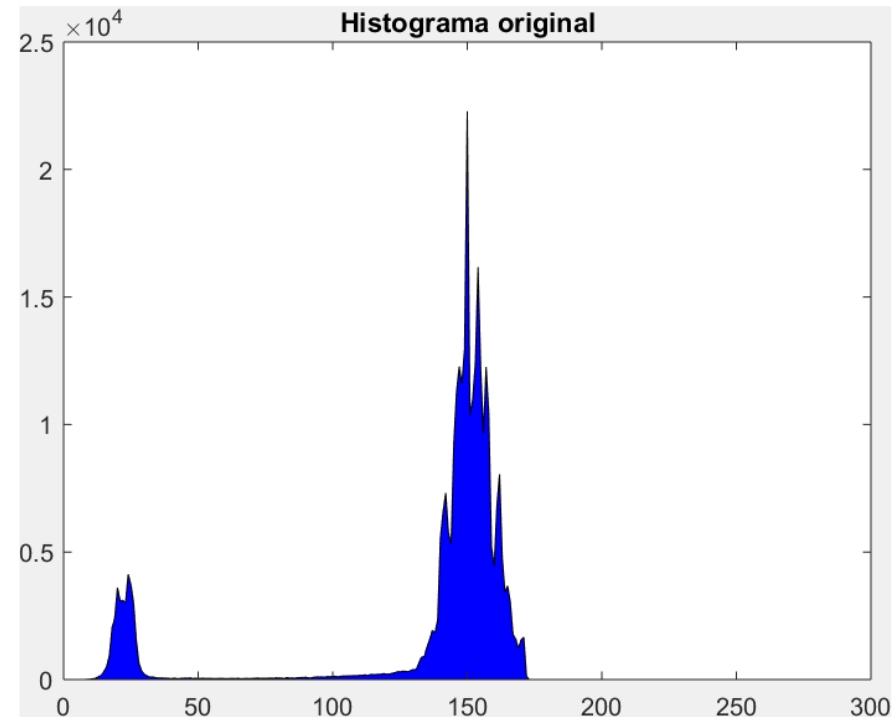


Imagen con 3 objetos



su histograma

```

p1=0; p2=0; u1=0; u2=0; sigma1=0; sigma2=0;
for i=1:256
    for k=1:i
        p1=p1+c_norm(k);
    end
    if p1==0
        p1=0.1;
    end
    for k=1:i
        u1=u1+(1/p1)*k*c_norm(k);
    end
    for k=1:i
        sigma1=sigma1+(1/p1)*(k-u1)*(k-u1)*c_norm(k);
    end
    if sigma1==0
        sigma1=0.1;
    end
    p2=1-p1;
    if p2==0
        p2=0.1;
    end
    for j=i+1:256
        u2=u2+(1/p2)*j*c_norm(j);
    end
    for j=1+1:256
        sigma2=sigma2+(1/p2)*(j-u1)*(j-u2)*c_norm(j);
    end
    if sigma2==0
        sigma2=0.1;
    end
    J_KI(i)=1+2*(p1*log(sigma1)+p2*log(sigma2))-2*(p1*log(p1)+p2*log(p2));
    p1=0; p2=0; u1=0; u2=0; sigma1=0; sigma2=0;
end

```

```

[M,th]=min(J_KI) % M contiene el valor máximo y th el umbral.

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

figure(3) % se abre una ventana auxiliar.
imshow(g) % se muestra imagen f(x,y).
title('Imagen umbralada') % se asigna título de esta imagen.

```



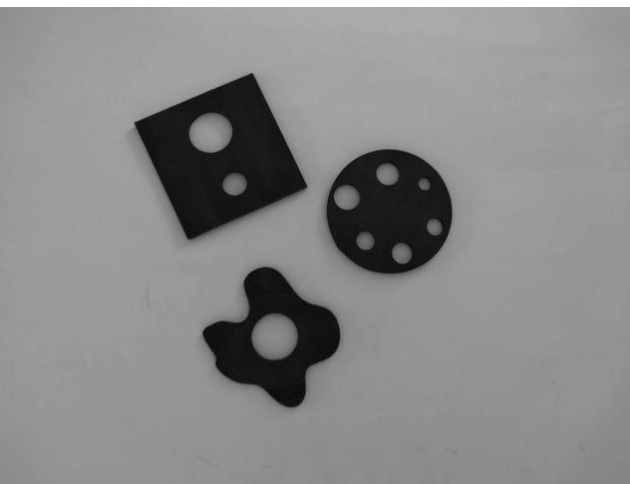
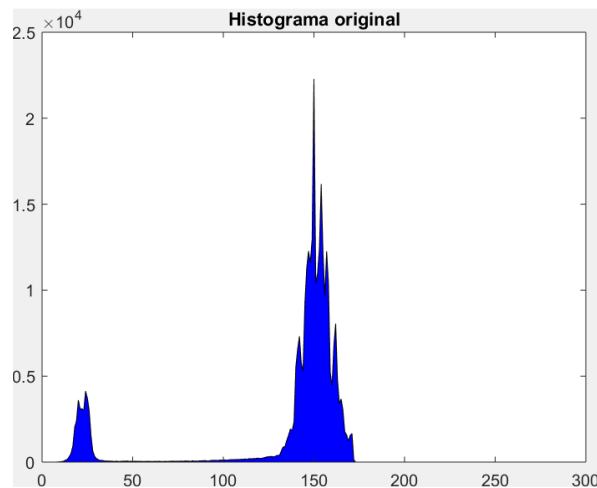
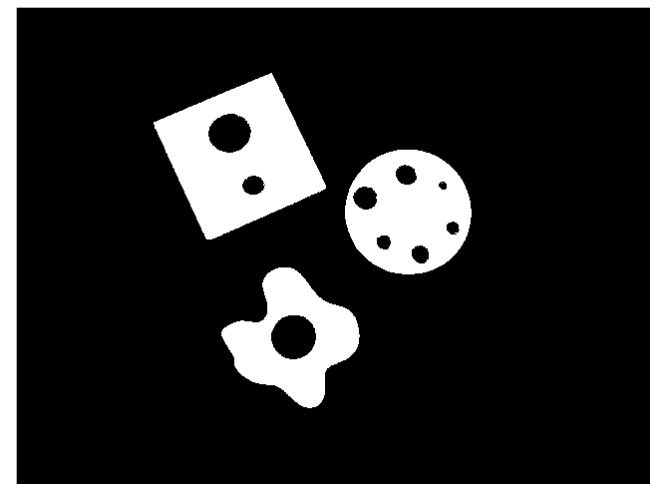


Imagen con 3 objetos



su histograma



umbralada con  $u = 66$

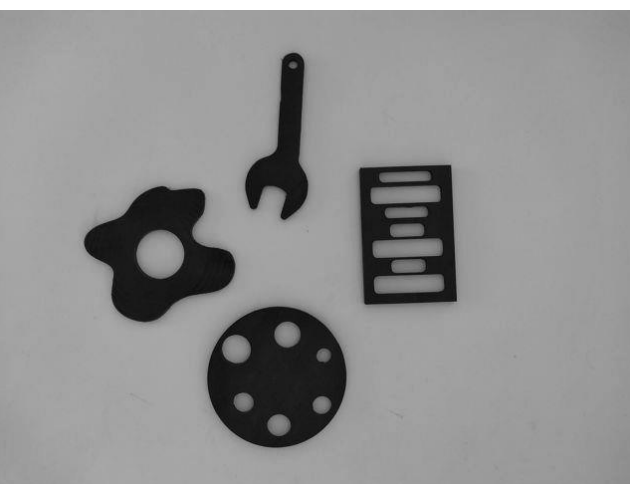
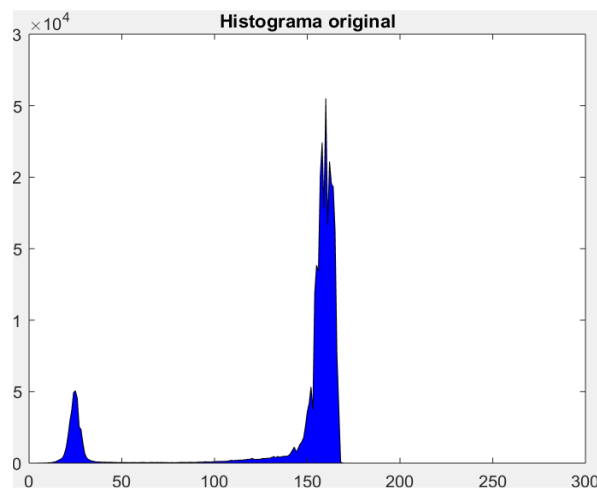
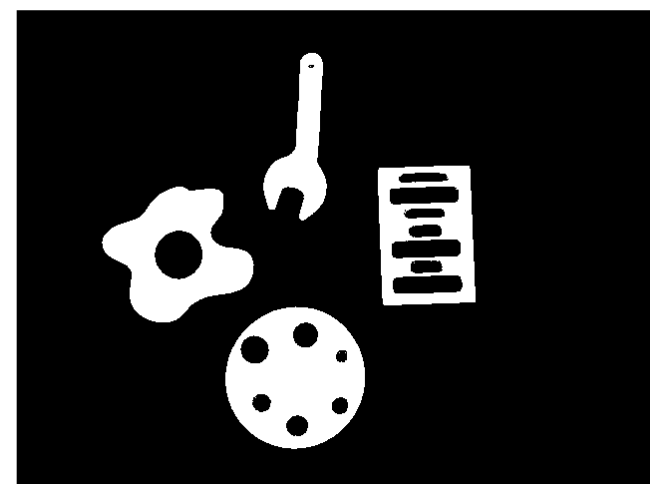


Imagen con 4 objetos



su histograma



umbralada con  $u = 76$

# Método de Kapur et al. basado en el cálculo de la máxima entropía (1985):

La maximización de la medida de información, de nuevo, entre dos clases:  $C_1$  y  $C_2$ :

Entropía total de la imagen:

$$H_T = - \sum_{r=0}^{L-1} p_r \ln(p_r)$$

J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Graph. Models Image Process. 29:273–285, 1985.

# Método de Kapur et al. basado en el cálculo de la máxima entropía:

Las entropías a priori de las dos clases se definen como:

$$\begin{aligned} H_{C_1}(u) &= - \sum_{r=0}^u \frac{p_r}{p(C_1)} \ln \left( \frac{p_r}{p(C_1)} \right) \\ &= - \frac{1}{p(C_1)} \left[ \sum_{r=0}^u p_r \ln p_r + p(C_1) \ln p(C_1) \right] \\ &= \ln p(C_1) + \frac{H_u}{p(C_1)} \end{aligned}$$
$$H(u) = - \sum_{r=0}^u p_r \ln(p_r)$$
$$\begin{aligned} H_{C_2}(u) &= - \sum_{r=u+1}^{L-1} \frac{p_r}{1-p(C_1)} \ln \left( \frac{p_r}{1-p(C_1)} \right) \\ &= - \frac{1}{1-p(C_1)} \left[ \sum_{r=u+1}^{L-1} p_r \ln p_r + (1-p(C_1)) \ln(1-p(C_1)) \right] \\ &= \ln(1-p(C_1)) + \frac{H_T - H_u}{1-p(C_1)} \end{aligned}$$

# Método de Kapur et al. basado en el cálculo de la máxima entropía:

La información de las dos clases:  $I(C_1, C_2) = H_{C_1}(u) + H_{C_2}(u)$

la cual puede ser rescrita como:

$$J_K(u) = \ln[p(C_1)(1 - p(C_1))] + \frac{H(u)}{p(C_1)} + \frac{H_T - H(u)}{1 - p(C_1)}$$

$$u^* = \arg \max_{0 \leq u \leq L-1} J_K(r)$$

problemas al momento de su aplicación:  $p_r \ln(p_r)$

**¿Cómo quedaría el  
código para la  
implementación del  
método de Kapur?**

**¡Dos puntos extra!**

## **Método basado en correlaciones de Cheng et al (1995):**

Jui-Cheng Yen, Fu-Juay Chang and Shyang Chang, A new criterion for automatic multilevel thresholding, in IEEE Transactions on Image Processing, vol. 4, no. 3, pp. 370-378, March 1995, doi: 10.1109/83.366472.

# Método basado en correlaciones de Cheng et al (1995):

Suponen a los píxeles como variables aleatorias:

$$C(X) = -\ln \sum_{r \geq 1} p_r^2$$

la cantidad total de correlación dada por las distribuciones de probabilidad:  $C_1$  y  $C_2$ :

$$\begin{aligned} \text{Con: } G(u) &= \sum_{r=1}^u p_r & CT(u) &= C_{C_1}(u) + C_{C_2}(u) \\ & & &= -\ln \sum_{r=1}^u \left( \frac{p_r}{P_1(u)} \right)^2 - \ln \sum_{r=u+1}^{L-1} \left( \frac{p_r}{1 - P_1(u)} \right)^2 \\ G'(u) &= \sum_{r=u+1}^{L-1} p_r & &= -\ln \left\{ \frac{G(u) \times G'(u)}{P_1^2(u) \times (1 - P_1^2(u))} \right\} \\ & & &= -\ln [G(u) \times G'(u)] + 2 \ln [P_1(u) \times (1 - P_1(u))] \end{aligned}$$

## Método basado en correlaciones de Cheng et al (1995):

A fin de obtener la correlación máxima entre el objeto de interés y el fondo en la imagen, hay que maximizar:

$$u^* = \arg \max_{0 \leq u \leq L-1} CT(u)$$

Problemas:  $\ln[G(u) \times G'(u)] \quad \ln[P_1(u) \times (1 - P_1(u))]$

$$\begin{aligned} CT(u) &= C_{C_1}(u) + C_{C_2}(u) \\ &= -\ln \sum_{r=1}^u \left( \frac{p_r}{P_1(u)} \right)^2 - \ln \sum_{r=u+1}^{L-1} \left( \frac{p_r}{1 - P_1(u)} \right)^2 \\ &= -\ln \left\{ \frac{G(u) \times G'(u)}{P_1^2(u) \times (1 - P_1^2(u))} \right\} \\ &= -\ln[G(u) \times G'(u)] + 2\ln[P_1(u) \times (1 - P_1(u))] \end{aligned}$$



## Método basado en correlaciones de Cheng et al:

Una manera de reducir el problema al momento de calcular los logaritmos, consiste en alisar la información dada por el histograma de la imagen antes de los cálculos.

Esto se puede lograr al usar el siguiente filtro:

$$p'_1 = \frac{p_{r-2} + p_{r-1} + p_r + p_{r+1} + p_{r+2}}{A} \quad 0 < A \leq 5$$

$$\begin{array}{ccccccccccccccccc} 3 & 5 & 2 & 0 & 1 & 3 & \mathbf{0} & 4 & 6 & 2 & 0 & 2 & 6 & \dots \\ & & & & & & 1 + 3 + 0 + 4 + 6 & & & & & & & \\ & & & & & & \hline & & & & & & 2.5 & & & & & & & \\ 3 & 5 & 2 & 0 & 1 & 3 & \mathbf{5.6} & 4 & 6 & 2 & 0 & 2 & 6 & \dots \\ 3 & 5 & 2 & 0 & 1 & 3 & \mathbf{5} & 4 & 6 & 2 & 0 & 2 & 6 & \dots \end{array}$$

**El tamaño  
del filtro  
puede ser  
cualquiera**

**¿Cómo quedaría el  
código para la  
implementación del  
método de Cheng et al?**

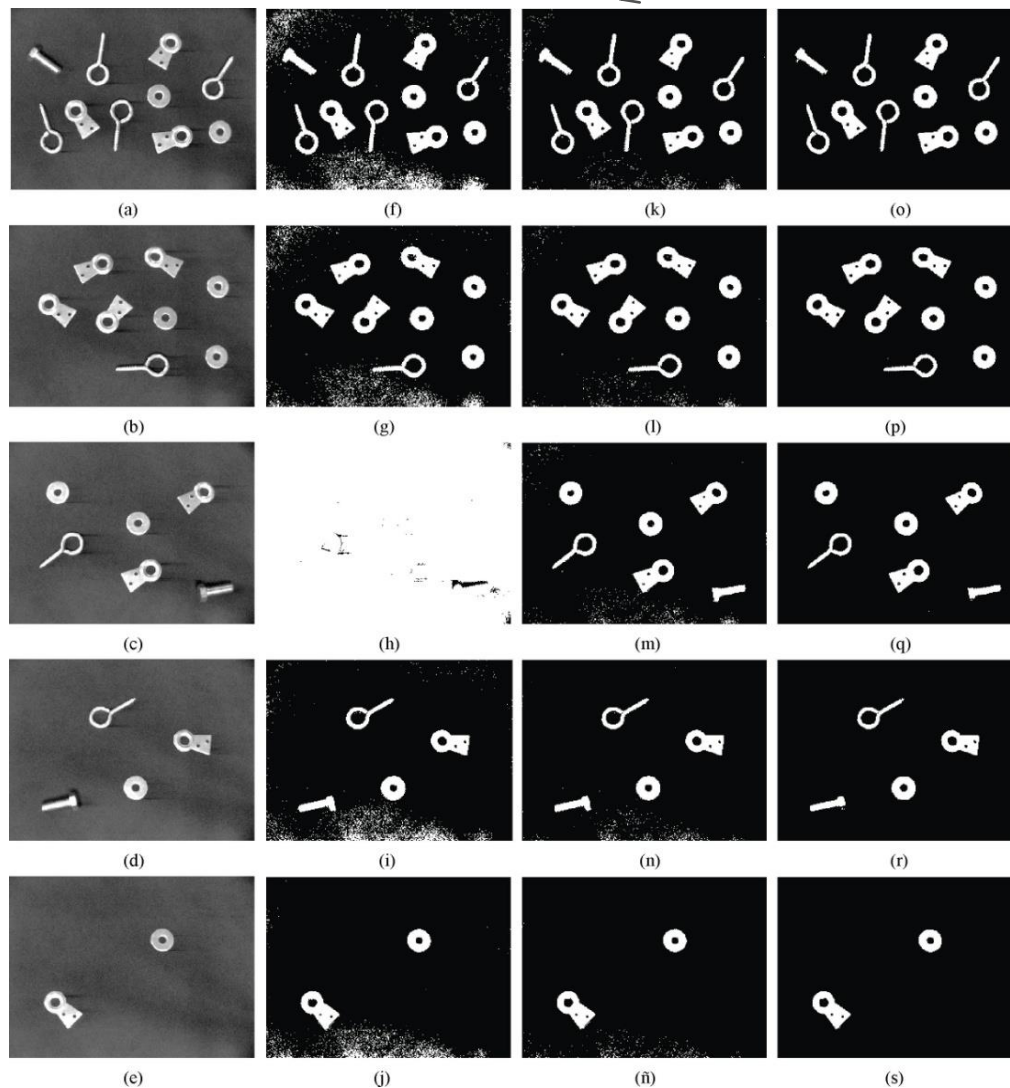
**¡Dos puntos extra!**

# Método basado en correlaciones de Cheng et al:

66, 69, 28, 79 y 81,  
sin aplicar el filtro.

73, 74, 81, 85 y 87,  
con filtro y  $A=5$ .

87, 89, 97, 103 y 104,  
con filtro y  $A=2.5$ .



# Método de Sahoo et al. basado en el cálculo de entropía de Renyi:

$$H_T^\alpha = \frac{1}{1-\alpha} \ln \sum_{r=0}^{L-1} (p_r)^\alpha$$

$\alpha \neq 1$  real y positivo.

La entropía de Renyi es una generalización en un parámetro de la entropía clásica de Shanon  $H_T$ , ya que:

$$\lim_{\alpha \rightarrow 1} H_T^\alpha = H_T$$

P. Sahoo, C. Wilkins, and J. Yeager, "Threshold selection using Re-nyi's entropy," Pattern Recogn.30,71–84-1997.

# Método de Sahoo et al. basado en el cálculo de entropía de Renyi:

Entropías del objeto y del fondo:

$$H_{C_1}^{\alpha}(u) = \frac{1}{1-\alpha} \ln \sum_{r=1}^u \left( \frac{p_r}{P_1(u)} \right)^{\alpha} \quad H_{C_2}^{\alpha}(u) = \frac{1}{1-\alpha} \ln \sum_{r=u+1}^{L-1} \left( \frac{p_r}{P_2(u)} \right)^{\alpha}$$

El valor de umbral  $u^*$  se escoge como:

$$u^* = \arg \max_{0 \leq u \leq L-1} (H_{C_1}^{\alpha} + H_{C_2}^{\alpha})$$

```
% Programa para umbralar imagen mediante método de Sahoo.

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

f=imread('4 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.

[m,n]=size(f); % dimensiones de la imagen f(x,y)
disp(n); % se muestra valor de n.
disp(m); % se muestra valor de m.
if ~islogical(f)
    f = im2uint8(f);
end;
```

```
c=zeros(256,3); % define una matriz con 3 filas de 256 elementos
```

```
for i=1:256
```

```
    c(i,1)=i-1;
```

```
end;
```

```
c(:,2) = imhist(f);
```

```
c (c(:,2)==0,:) = [];
```

```
c(:,3) = c(:,2) ./ numel(f);
```

```
alpha=-0.5;
```

```
ArgMax=0;
```

```

for t=1 : size(c)
    PA=sum(c(1:t,3));
    PB=1-PA ;
    p1=c(1:t,3)/PA;
    p2=c(t+1: size(c),3)/PB;
    Ha=log2( sum ((p1.^alpha)) )/(1-alpha);
    Hb=log2( sum ((p2.^alpha)) )/(1-alpha);
    Hab = Ha + Hb;
    if ( Hab > ArgMax )
        t_opt=c(t,1);
        ArgMax=Hab;
    end;
end;

disp(t_opt);

```

$$H_{C_1}^{\alpha}(u) = \frac{1}{1-\alpha} \ln \sum_{r=1}^u \left( \frac{p_r}{P_1(u)} \right)^{\alpha}$$

$$H_{C_2}^{\alpha}(u) = \frac{1}{1-\alpha} \ln \sum_{r=u+1}^{L-1} \left( \frac{p_r}{P_2(u)} \right)^{\alpha}$$



```
g=double(f);

for i=1:m;
    for j=1:n;
        g(i,j)=1;
        if (f(i,j) >= t_opt)
            g(i,j)=0;
        end;
    end;
end;

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

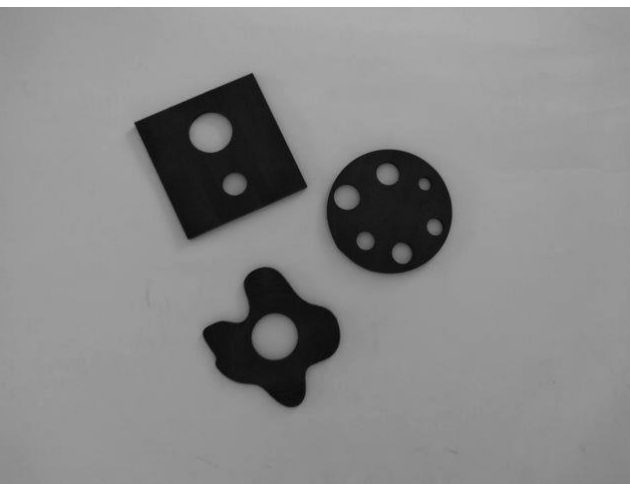
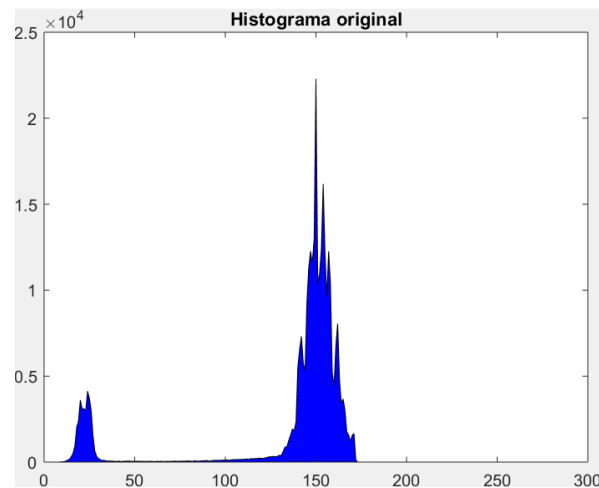
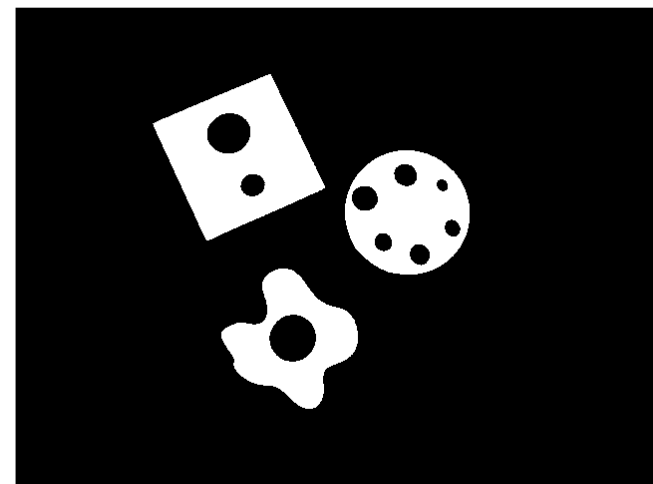


Imagen con 3 objetos



su histograma



$$\alpha = -0.5 \quad u = 56$$

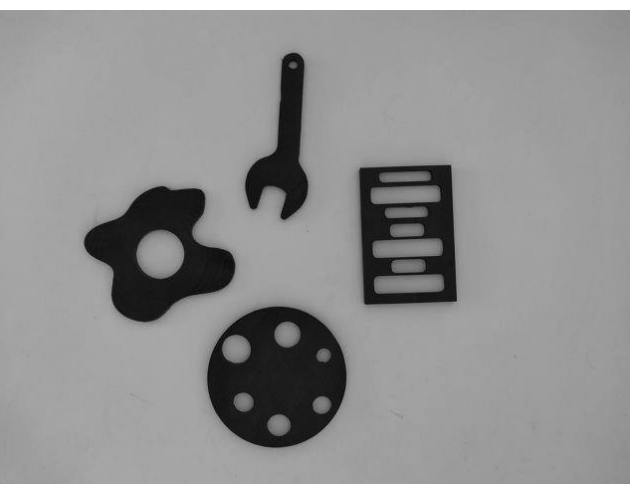
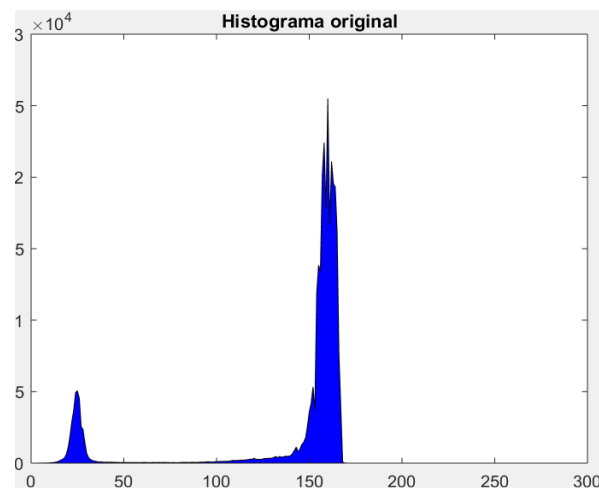
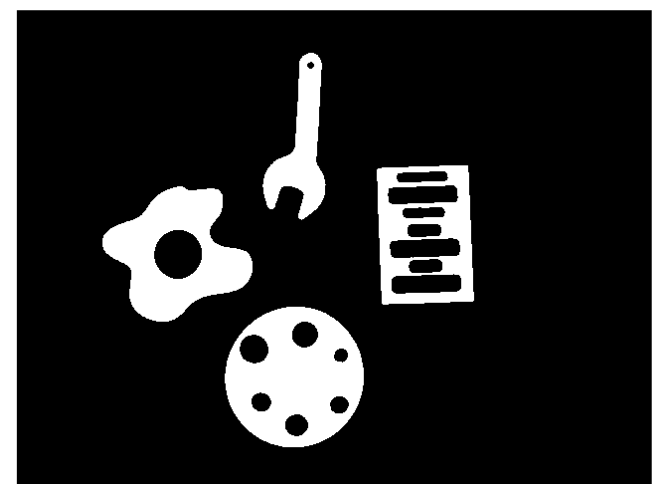


Imagen con 4 objetos



su histograma



$$\alpha = -0.5 \quad u = 70$$

# Método de Sahoo et al. basado en el cálculo de entropía de Renyi:

$\alpha = 0.9$     $\alpha = 0.5$     $\alpha = 0.1$

73

78

121

75

77

123

82

84

119

86

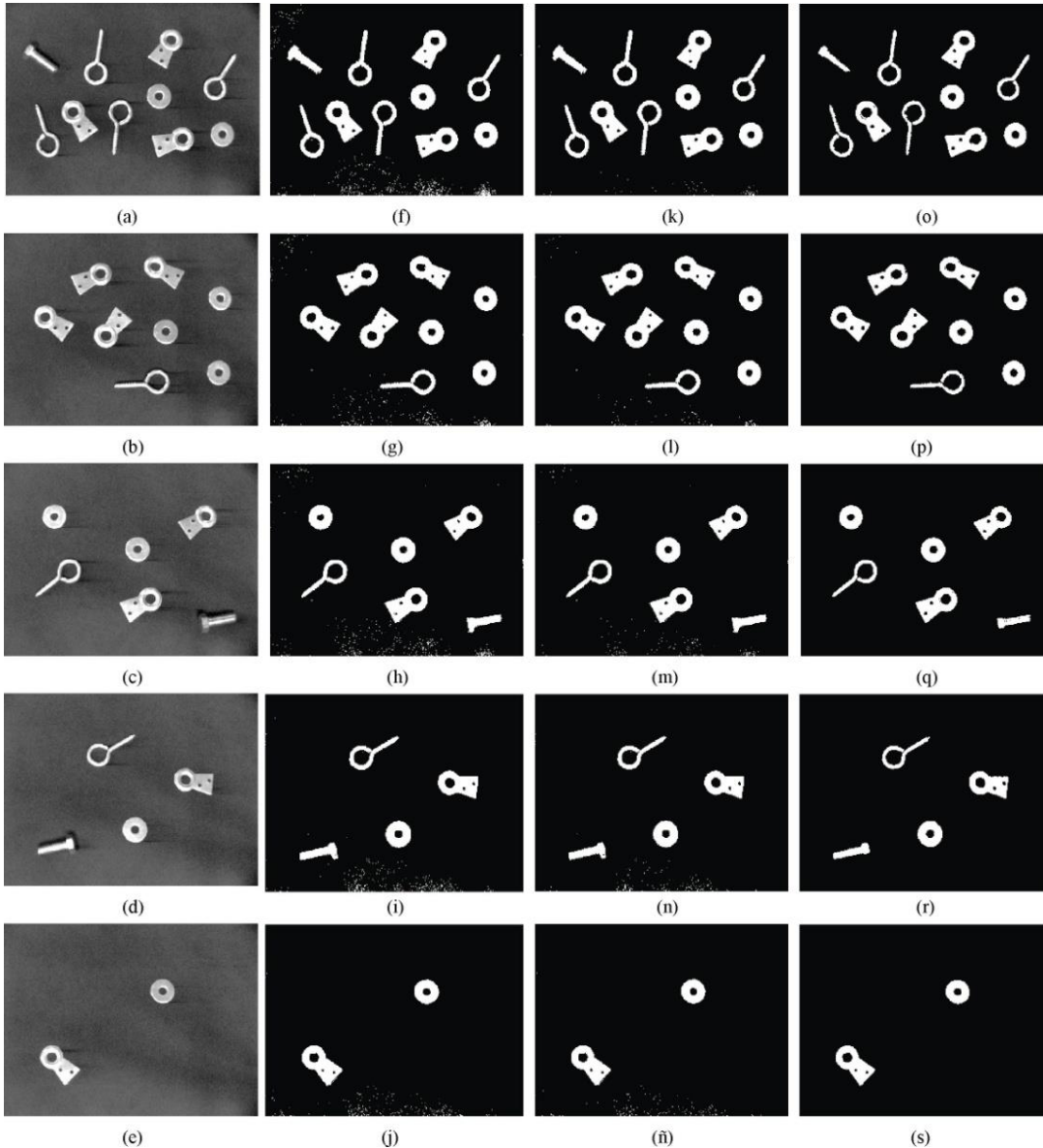
88

118

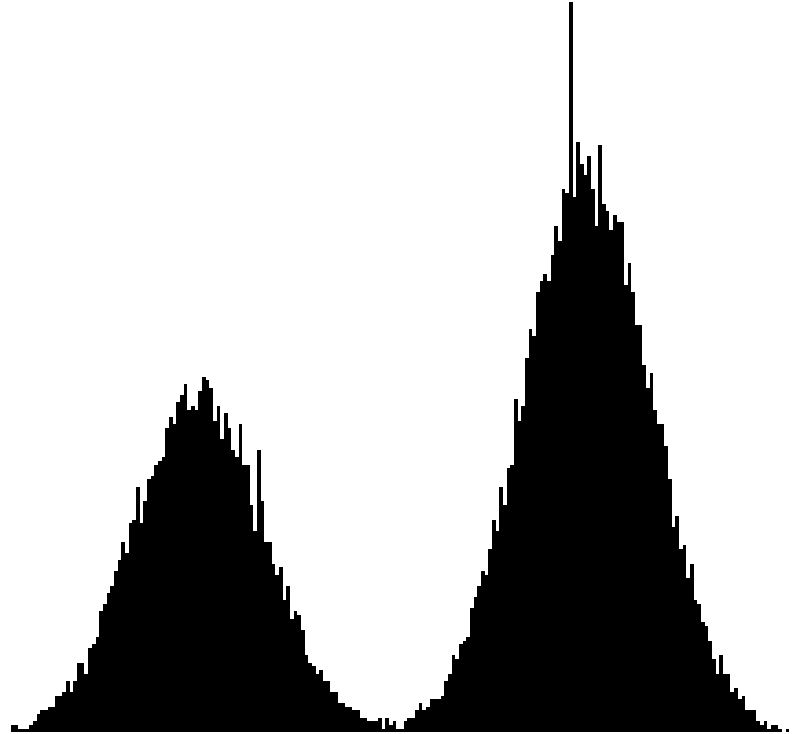
88

89

116

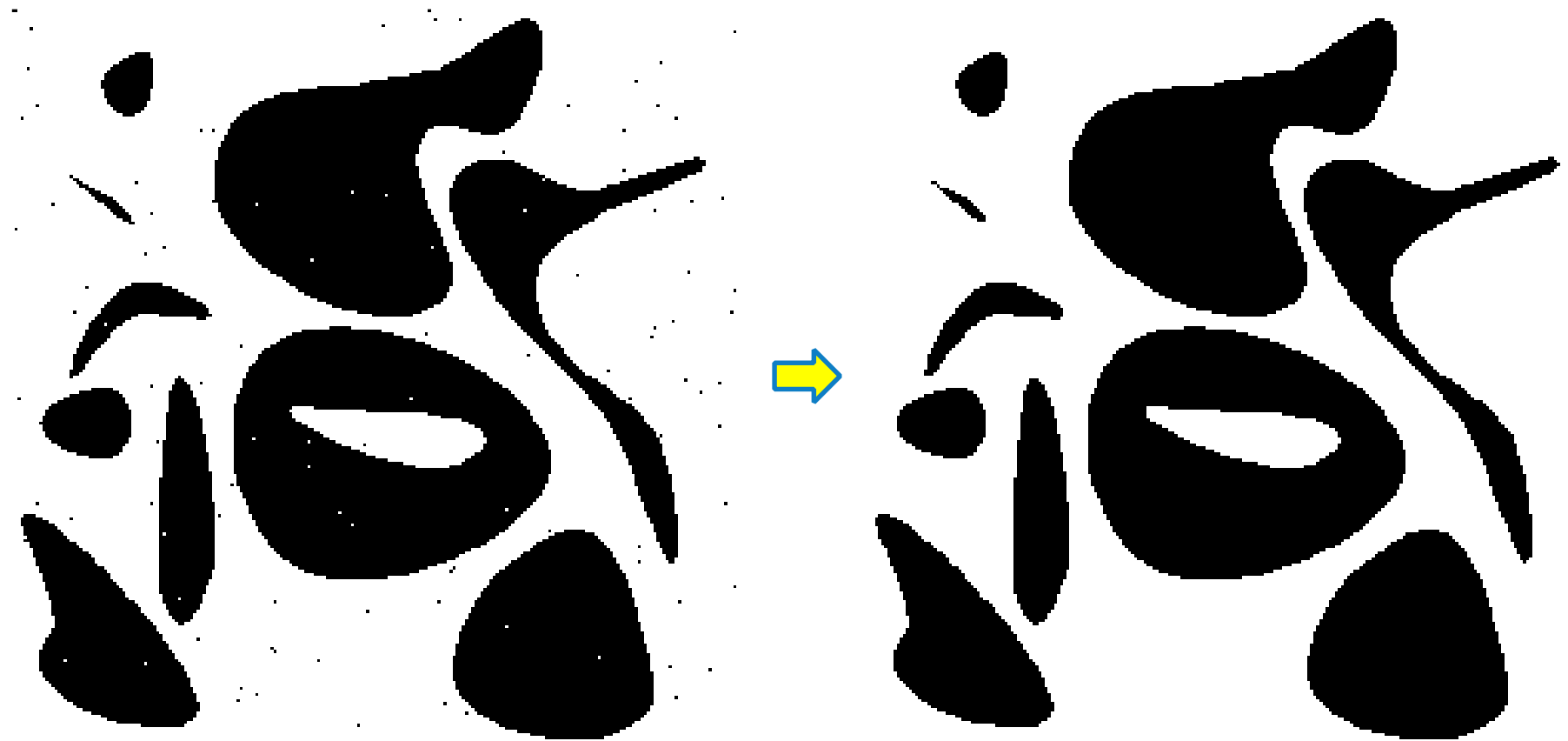


# Efecto del ruido en la segmentación:





## Solución mediante MM (apertura-cerradura):



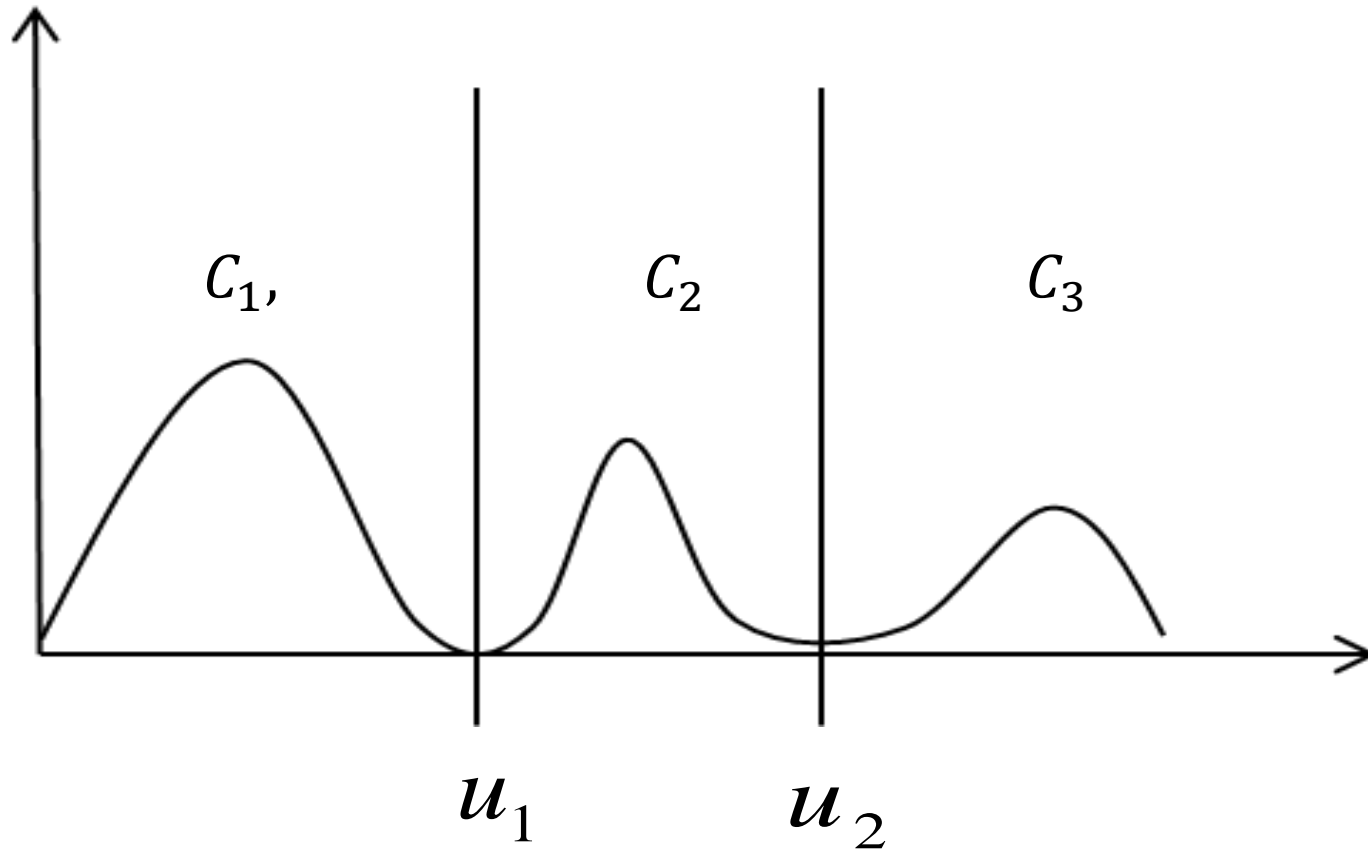
## Solución mediante filtrado mediano:



# **Extensión al caso de umbralado múltiple:**



## Histograma con múltiples modas:



## Extensión al caso umbralado múltiple:

Método de Otsu para tres clases:  $C_1, C_2$  y  $C_3$ :

$$C_1 \text{ para } \{1, \dots, u_1\} \quad C_2 \text{ para } \{u_1 + 1, \dots, u_2\} \quad C_3 \text{ para } \{u_2 + 1, \dots, L-1\}$$

Mediante dos umbrales:  $u_1 \quad u_2$

$$0 \leq u_1 < u_2 < L-1$$

$$J_i(u_1^*, u_2^*) = \arg \max_{1 \leq u_1 < u_2 < L-1} J(u_1, u_2), i = 1, 2, 3$$

## Extensión al caso umbralado múltiple:

En el caso del método de Kittler y Illingworth, si se suponen  $m$  modos, i.e. el histograma es una mezcla de  $m$  densidades normales. El criterio a minimizar es entonces:

$$J_{KI}(u_1, u_2, \dots, u_{m-1}) = 1 + 2 \sum_{i=1}^m \{P_i(u_i) [\log \sigma_i(u_i) - \log P_i(u_i)]\}$$

Donde:

$$P_i(u_i) = \sum_{r=T_{i-1}+1}^{u_i} h(r) \quad \mu_i(u_i) = \frac{1}{P_i(u_i)} \sum_{r=T_{i-1}+1}^{u_i} r h(r)$$

$$\sigma_i^2(u_i) = \frac{1}{P_i(u_i)} \sum_{r=T_{i-1}+1}^{u_i} [r - \mu_i(u_i)]^2 h(r) \quad \begin{aligned} T_m &= L-1, \\ T_0 &= -1 \end{aligned}$$

## Extensión al caso umbralado múltiple:

Para todos los métodos arriba descritos, el número de posibles conjuntos de umbrales candidatos a ser seleccionado es muy grande, viniendo dado como:

$$\frac{(n+1)!}{(n+2-m)!(m-1)!}$$

$m$  el número de umbrales a seleccionar y  $n$  el número de niveles de gris menos uno de la imagen. Por ejemplo para  $m = 3, n = 255$ , el número de puntos para el cual la función criterio debe ser evaluado es 32,640.

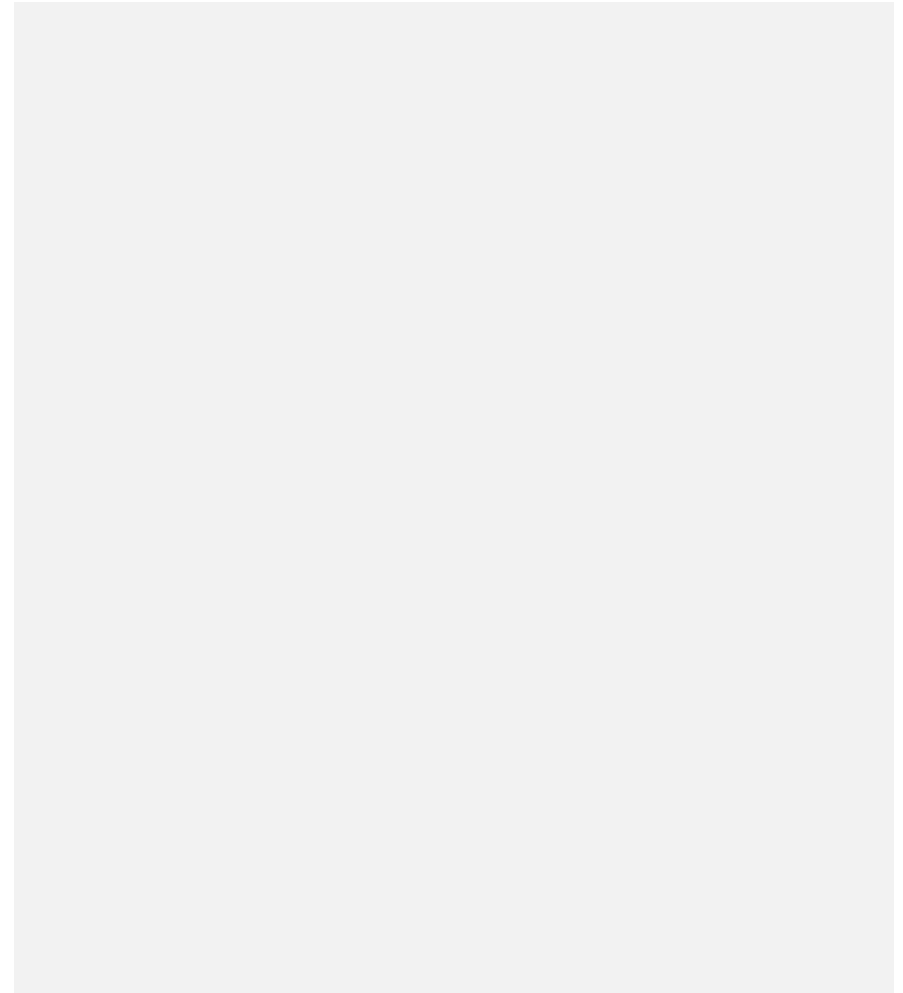
# **Etiquetado de regiones o componentes conectadas:**

# Etiquetado de regiones o componentes conectadas (IDEA):

(a)

(b)

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



# Etiquetado de regiones o componentes conectadas (IDEA):

(a)
(b)

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

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

Idea: Producir a partir de una imagen binaria una imagen con regiones conectadas etiquetadas. ¿Cómo lo hacemos?

# ALGORITMO RECURSIVO:

Dada una imagen binaria  $b(x, y)$ :

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ .



# ALGORITMO RECURSIVO:

Dada una imagen binaria  $b(x, y)$ :

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ .
2. Asignar en forma recursiva la etiqueta  $E$  a todos sus vecinos, según la métrica elegida.

# ALGORITMO RECURSIVO:

Dada una imagen binaria  $b(x, y)$ :

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ .
2. Asignar en forma recursiva la etiqueta  $E$  a todos sus vecinos, según la métrica elegida.
3. Detenerse si durante el recorrido no se encuentran más píxeles de tipo objeto.

# ALGORITMO RECURSIVO:

Dada una imagen binaria  $b(x, y)$ :

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ .
2. Asignar en forma recursiva la etiqueta  $E$  a todos sus vecinos, según la métrica elegida.
3. Detenerse si durante el recorrido no se encuentran más píxeles de tipo objeto.
4. Regresar al paso 1.

# ALGORITMO RECURSIVO:

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

	1		1		2	2	
	1	1	1				
	1	1	1			1	
						1	
						1	
	1	1			1	1	

	1		1		2	2	
	1	1	1				
	1	1	1			3	
						3	
						3	
	1	1			3	3	

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

# ALGORITMO ITERATIVO:

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ . Esto da como resultado la imagen  $e1(x, y)$ .

# ALGORITMO ITERATIVO:

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ . Esto da como resultado la imagen  $e1(x, y)$ .
2. Barrer  $e1(x, y)$  de arriba hacia abajo hasta encontrar un píxel etiquetado y propagar su etiqueta a sus vecinos, según la métrica elegida. Esto da como resultado la imagen  $e2(x, y)$ .

# ALGORITMO ITERATIVO:

1. Barrer  $b(x, y)$  hasta encontrar un píxel de tipo objeto (con valor 1 o  $L-1$ ) aún no etiquetado y asignarle una nueva etiqueta  $E$ . Esto da como resultado la imagen  $e1(x, y)$ .
2. Barrer  $e1(x, y)$  de arriba hacia abajo hasta encontrar un píxel etiquetado y propagar su etiqueta a sus vecinos, según la métrica elegida. Esto da como resultado la imagen  $e2(x, y)$ .
3. Barrer  $e2(x, y)$  de abajo hacia arriba hasta encontrar un píxel etiquetado y propagar su etiqueta a sus vecinos, según la métrica elegida. Esto da como resultado la imagen  $e3(x, y)$ .

# ALGORITMO ITERATIVO:

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



# ALGORITMO ITERATIVO:

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

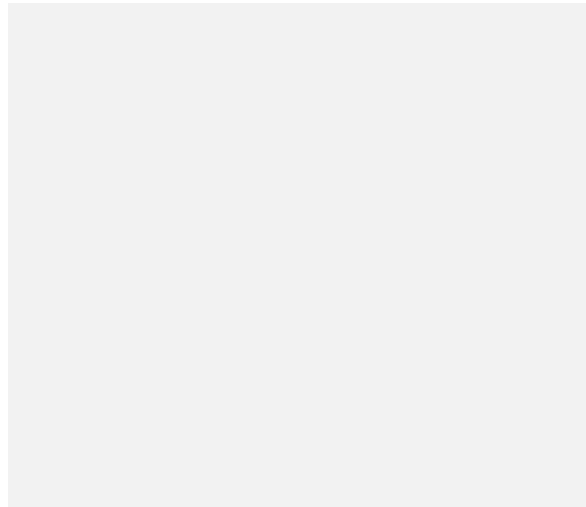
	1		2		3	4	
	5	6	7				
	8	9	10			11	
						12	
						13	
	14	15			16	17	

# ALGORITMO ITERATIVO:

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

	1		2		3	4	
	5	6	7				
	8	9	10			11	
						12	
						13	
	14	15			16	17	

	1		2		3	3	
	1	1	1				
	1	1	1			11	
						11	
						11	
	14	14			16	11	



# ALGORITMO ITERATIVO:

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

	1		2		3	4	
	5	6	7				
	8	9	10			11	
						12	
						13	
	14	15			16	17	

	1		2		3	3	
	1	1	1				
	1	1	1			11	
						11	
						11	
	14	14			16	11	

	1		1		3	3	
	1	1	1				
	1	1	1			11	
						11	
						11	
	14	14			11	11	

## **ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:**

Este algoritmo realiza dos pasadas sobre la imagen y una tabla de equivalencias para etiquetar una imagen de entrada.

## **ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:**

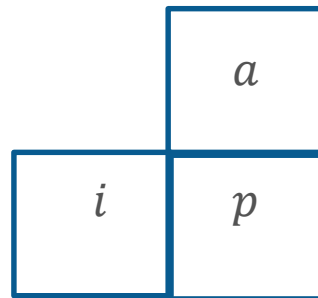
1. Barrer imagen de arriba abajo y de izquierda a derecha.

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

1. Barrer imagen de arriba abajo y de izquierda a derecha.
2. Si  $p$  es un píxel de fondo, esto es  $p = 0$ , desplazarse a la siguiente posición.

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

1. Barrer imagen de arriba abajo y de izquierda a derecha.
2. Si  $p$  es un píxel de fondo, esto es  $p = 0$ , desplazarse a la siguiente posición.
3. Si  $p$  es un píxel de objeto, esto es  $p = 1(L - 1)$ :
  - a) Si  $a$  o  $i$  son de fondo, entonces asignar a  $p$  nueva etiqueta.
  - b) Si  $a$  o  $i$  es de objeto, entonces asignar a  $p$  la etiqueta de dicho píxel.
  - c) Si  $a$  y  $i$  son de tipo objeto, asignar a  $p$  la etiqueta de cualquiera de ellos.
  - d) Si  $a$  y  $i$  son de tipo objeto pero con etiquetas diferentes, asignar a  $p$  etiqueta más pequeña y almacenar en la TE hecho que las 2 dos etiquetas son equivalentes.
  - e) De otra manera, asignar a  $p$  nueva etiqueta, poner su etiqueta en TE.



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

1. Barrer imagen de arriba abajo y de izquierda a derecha.
2. Si  $p$  es un píxel de fondo, esto es  $p = 0$ , desplazarse a la siguiente posición.
3. Si  $p$  es un píxel de objeto, esto es  $p = 1(L - 1)$ :
  - a) Si  $a$  o  $i$  son de fondo, entonces asignar a  $p$  nueva etiqueta.
  - b) Si  $a$  o  $i$  es de objeto, entonces asignar a  $p$  la etiqueta de dicho píxel.
  - c) Si  $a$  y  $i$  son de tipo objeto, asignar a  $p$  la etiqueta de cualquiera de ellos.
  - d) Si  $a$  y  $i$  son de tipo objeto pero con etiquetas diferentes, asignar a  $p$  etiqueta más pequeña y almacenar en la TE hecho que 2 dos etiquetas son equivalentes.
  - e) De otra manera, asignar a  $p$  nueva etiqueta, poner su etiqueta en TE.
4. Si hay más píxeles en la imagen, entonces regresar al paso 2.



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

1. Barrer imagen de arriba abajo y de izquierda a derecha.
2. Si  $p$  es un píxel de fondo, esto es  $p = 0$ , desplazarse a la siguiente posición.
3. Si  $p$  es un píxel de objeto, esto es  $p = 1(L - 1)$ :
  - a) Si  $a$  o  $i$  son de fondo, entonces asignar a  $p$  nueva etiqueta.
  - b) Si  $a$  o  $i$  es de objeto, entonces asignar a  $p$  la etiqueta de dicho píxel.
  - c) Si  $a$  y  $i$  son de tipo objeto, asignar a  $p$  la etiqueta de cualquiera de ellos.
  - d) Si  $a$  y  $i$  son de tipo objeto pero con etiquetas diferentes, asignar a  $p$  etiqueta más pequeña y almacenar en la TE hecho que 2 dos etiquetas son equivalentes.
  - e) De otra manera, asignar a  $p$  nueva etiqueta, poner su etiqueta en TE.
4. Si hay más píxeles en la imagen, entonces regresar al paso 2.
5. Encontrar etiqueta más pequeña para cada conjunto de etiquetas en la TE.

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

1. Barrer imagen de arriba abajo y de izquierda a derecha.
2. Si  $p$  es un píxel de fondo, esto es  $p = 0$ , desplazarse a la siguiente posición.
3. Si  $p$  es un píxel de objeto, esto es  $p = 1(L - 1)$ :
  - a) Si  $a$  o  $i$  son de fondo, entonces asignar a  $p$  nueva etiqueta.
  - b) Si  $a$  o  $i$  es de objeto, entonces asignar a  $p$  la etiqueta de dicho píxel.
  - c) Si  $a$  y  $i$  son de tipo objeto, asignar a  $p$  la etiqueta de cualquiera de ellos.
  - d) Si  $a$  y  $i$  son de tipo objeto pero con etiquetas diferentes, asignar a  $p$  etiqueta más pequeña y almacenar en la TE hecho que 2 dos etiquetas son equivalentes.
  - e) De otra manera, asignar a  $p$  nueva etiqueta, poner su etiqueta en TE.
4. Si hay más píxeles en la imagen, entonces regresar al paso 2.
5. Encontrar etiqueta más pequeña para cada conjunto de etiquetas en la TE.
6. Barrer de nuevo la imagen y reemplazar cada etiqueta por la más pequeña en su conjunto de equivalencia.

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1									

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1								

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	<b>1</b>	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1							

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1									



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1								

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2					

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3			

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		<b>1</b>	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4									

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

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

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

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



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

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

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	<b>1</b>	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>			

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			<b>1</b>			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4								

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2					

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	<b>1</b>	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2				

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	<b>1</b>			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	<b>1</b>		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5										

2 y 3 equivalentes



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	<b>1</b>	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5										

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	<b>1</b>	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5									

2 y 3 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	<b>1</b>						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>								

2 y 3 equivalentes  
4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		

2 y 3 equivalentes

4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>			<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>		<b>1</b>	<b>1</b>	<b>1</b>			
							<b>1</b>	<b>1</b>	<b>1</b>		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5										

2 y 3 equivalentes

4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	<b>1</b>	1	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5									

2 y 3 equivalentes

4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	<b>1</b>	1			1	1	1		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4								

2 y 3 equivalentes  
4 y 5 equivalentes



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4							

2 y 3 equivalentes  
4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7				

2 y 3 equivalentes

4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7			

2 y 3 equivalentes  
4 y 5 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	<b>1</b>		
	1	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	<b>1</b>	1	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5										

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	<b>1</b>	1	1		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5									

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4								

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	<b>1</b>		1	1	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4							

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes



## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8					

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	<b>1</b>	1			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	7				

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

7 y 8 equivalentes

# ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

		1	1	1	1						
		1	1	1							
						1		1	1		
		1	1			1	1	1	1		
			1			1	1	1			
	1		1								
	1	1	1						1		
	1	1	1	1			1	1	1		
	1	1	1	1		1	1	<b>1</b>			
							1	1	1		

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	<u>7</u>	7			

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

7 y 8 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	<u>7</u>	7			
							7				

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

7 y 8 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	<u>7</u>	7			
							7	7			

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

7 y 8 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	<u>7</u>	7			
							7	7	7		

2 y 3 equivalentes

4 y 5 equivalentes

6 y 7 equivalentes

7 y 8 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

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

		1	1	1	1						
		1	1	1							
						2		3	3		
		4	4			2	2	<u>2</u>	2		
			4			2	2	2			
	5		4								
	5	5	<u>4</u>						6		
	5	5	4	4			7	7	<u>6</u>		
	5	5	4	4		8	<u>7</u>	7			
							7	7	7		

2 y 3 equivalentes

4 y 5 equivalentes

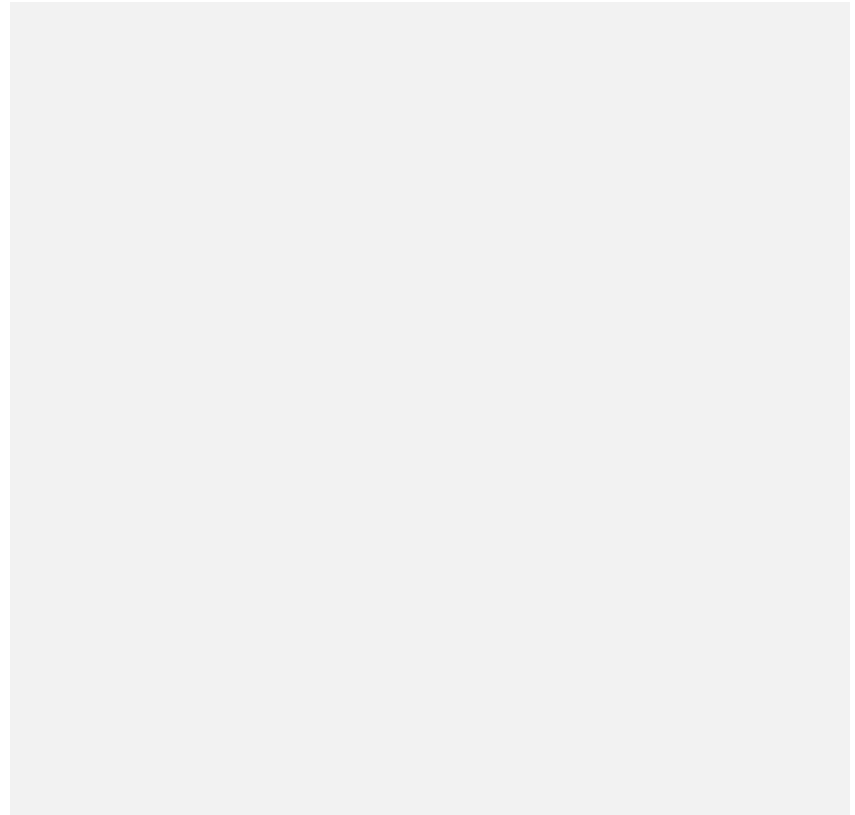
6 y 7 equivalentes

7 y 8 equivalentes

## ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

Tabla de equivalencias
$1 \equiv 1$
$2 \equiv 3$
$4 \equiv 5$
$6 \equiv 7 \equiv 8$

Por transitividad 6, 7 y 8 son equivalentes

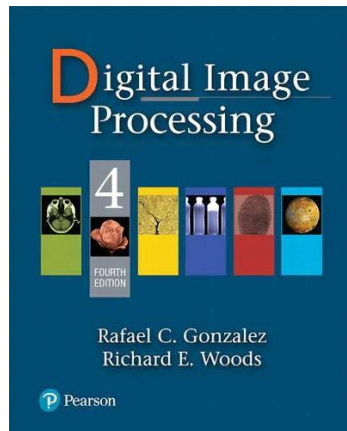




# ALGORITMO DE DOS PASOS QUE USA UNA TABLA DE EQUIVALENCIAS:

Tabla de equivalencias
$1 \equiv 1$
$2 \equiv 3$
$4 \equiv 5$
$6 \equiv 7 \equiv 8$

Por transitividad 6, 7 y 8 son equivalentes



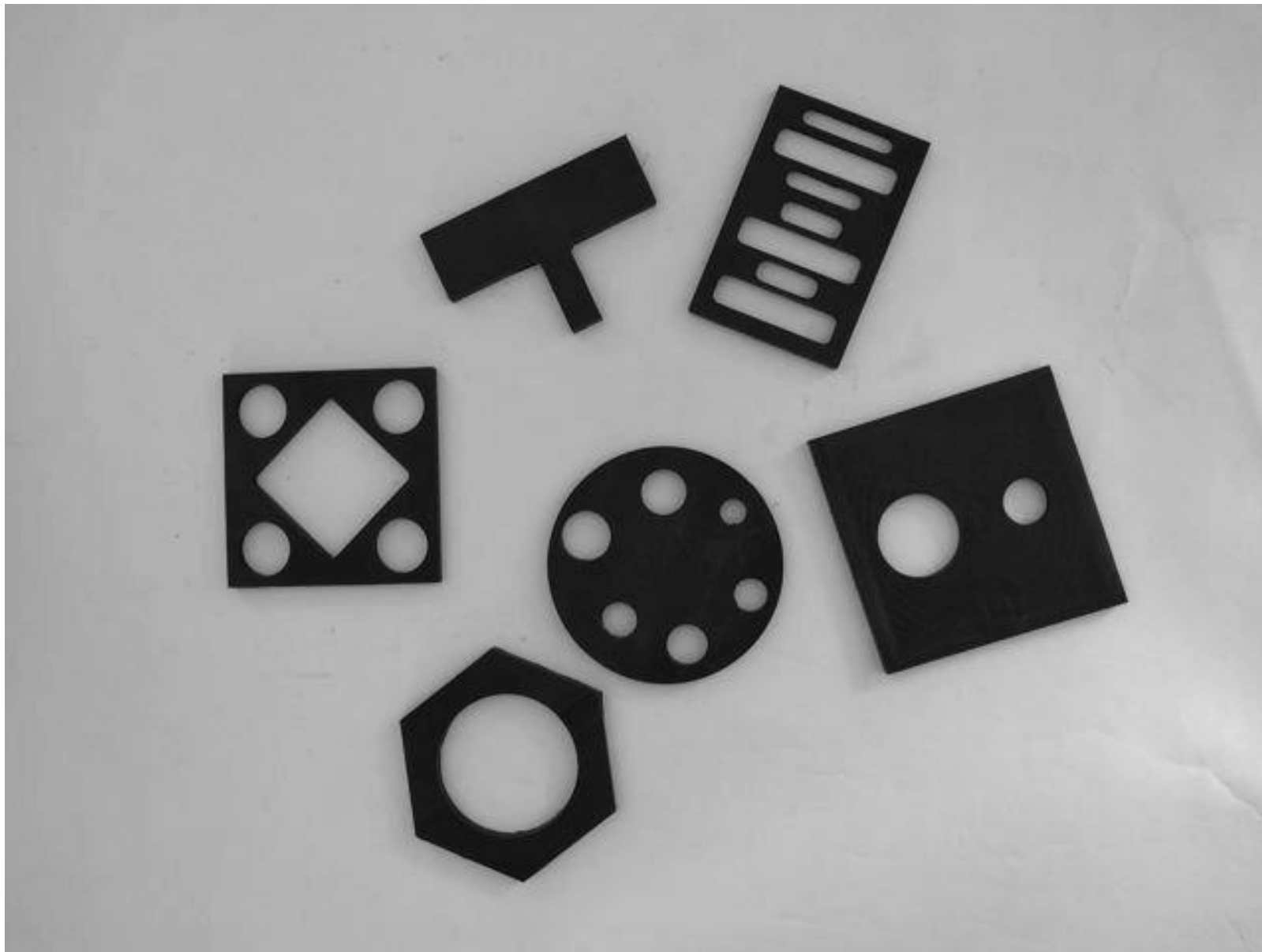
		1	1	1	1						
		1	1	1							
						2		2	2		
		4	4			2	2	2	2		
			4			2	2	2			
	4		4								
	4	4	4						6		
	4	4	4	4			6	6	6		
	4	4	4	4		6	6	6			
							6	6	6		

```
% Programa para etiquetado de componentes conectados.

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

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

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



**Imagen original con 6 objetos**

```
th = graythresh(f); % calcula umbral th de imagen.  
  
g = imbinarize(f,th); % umbrala imagen con th.  
  
g_c = imcomplement(g); % complementa imagen.  
  
figure(2) % se abre una ventana auxiliar.  
imshow(g_c) % se muestra imagen f(x,y).  
title('Imagen binaria') % se asigna título de esta imagen.
```

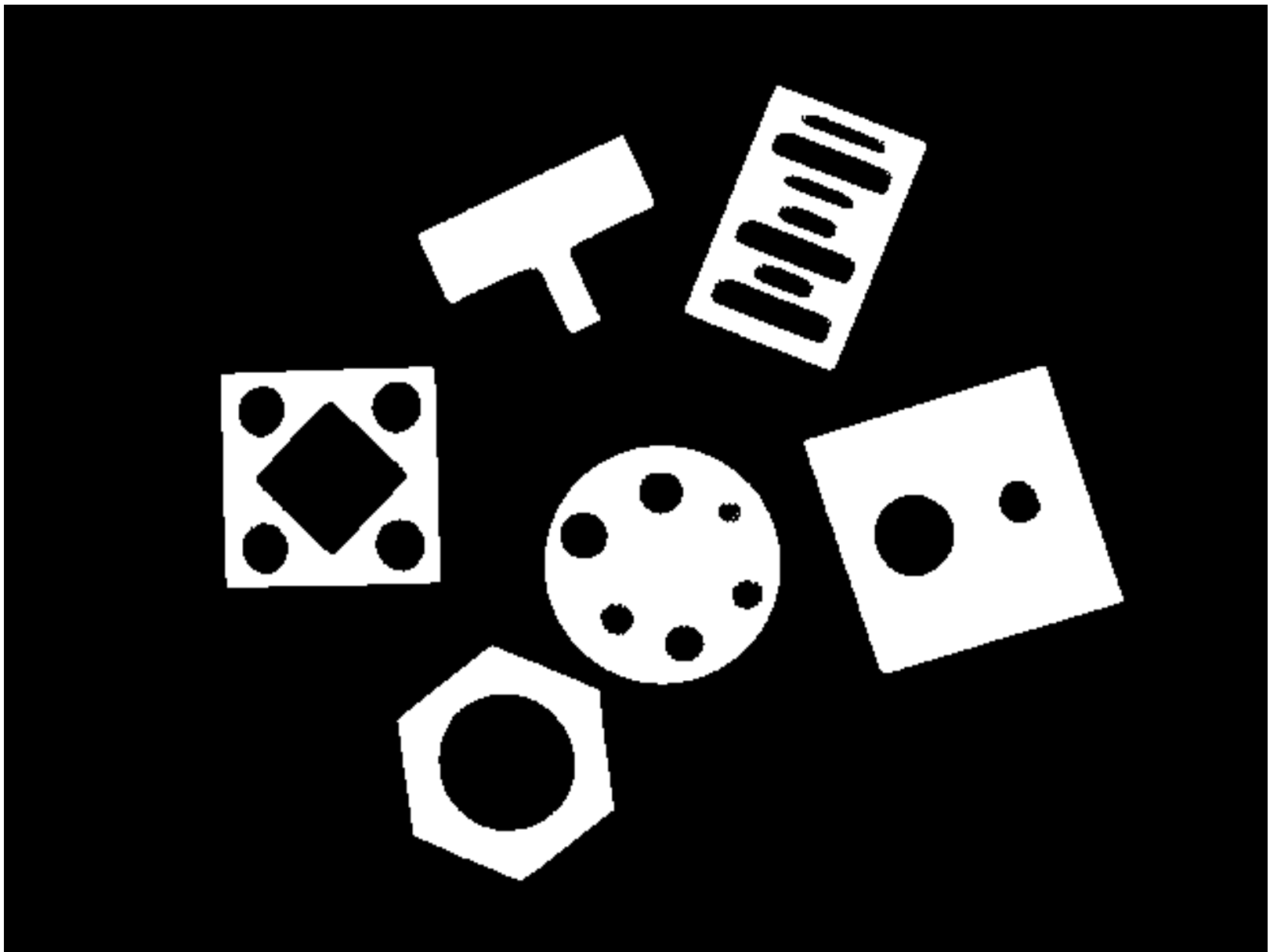


Imagen umbralada

```
se = strel('disk',1); % disk de radio 1.
g_open = imopen(g_c,se); % apertura de imagen.

figure(3)
imshow(g_open) % muestra imagen filtrada.
title('Imagen abierta') % titula imagen.

cc = bwconncomp(g_open,4)
% regresa los CC encontrados en la imagen g_open
% 4-conectado

etiquetada = labelmatrix(cc); % produce imagen etiquetada.
RGB_label = label2rgb(etiquetada,@copper,'w','shuffle');

figure(4)
imshow(RGB_label) % muestra imagen en color y fondo blanco "w".
```



Imagen original con 6 objetos

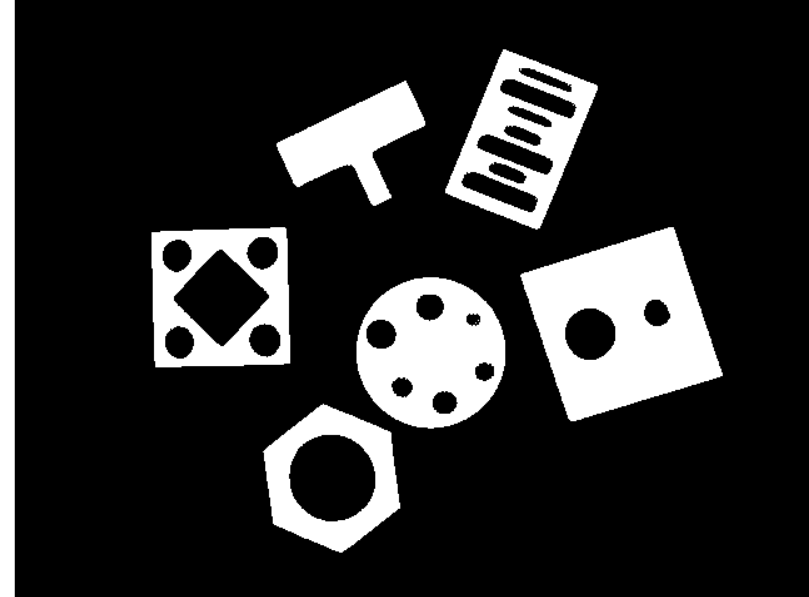


Imagen umbralada

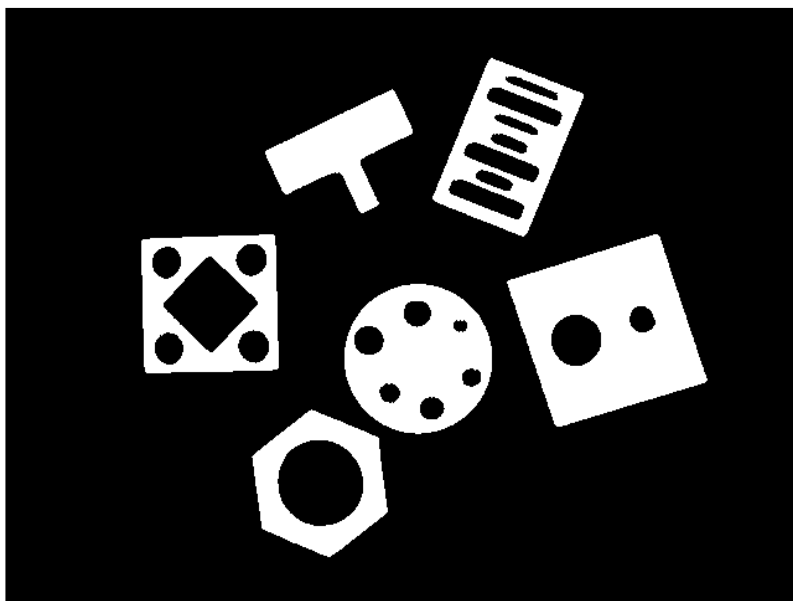


Imagen abierta

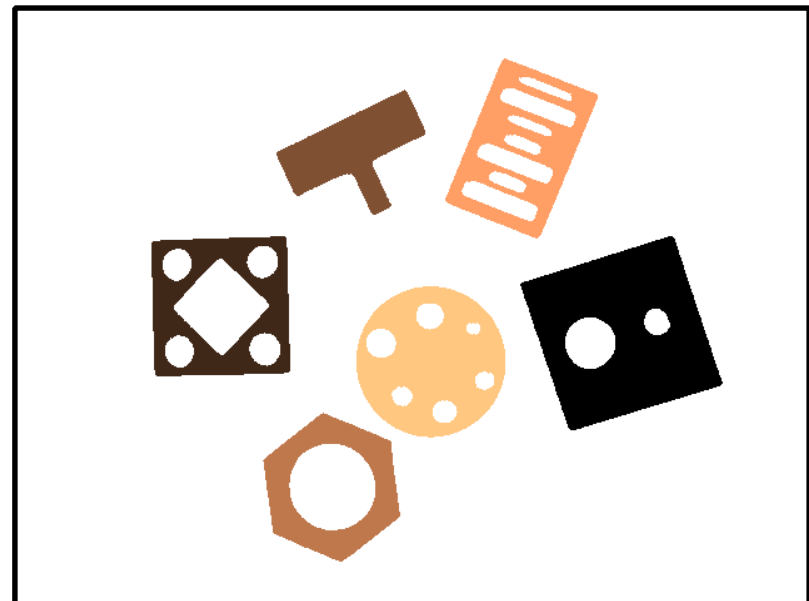


Imagen etiquetada con colores

# Resultados:

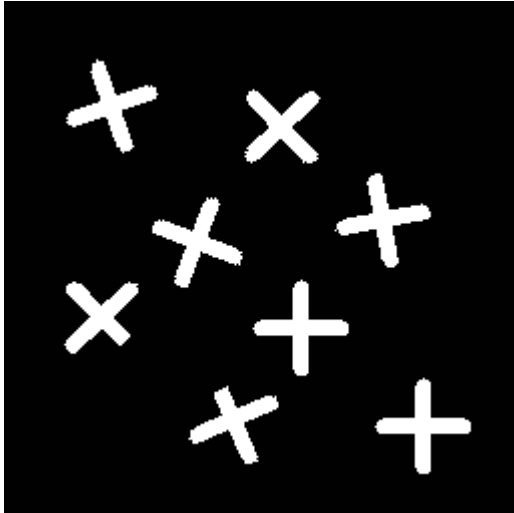
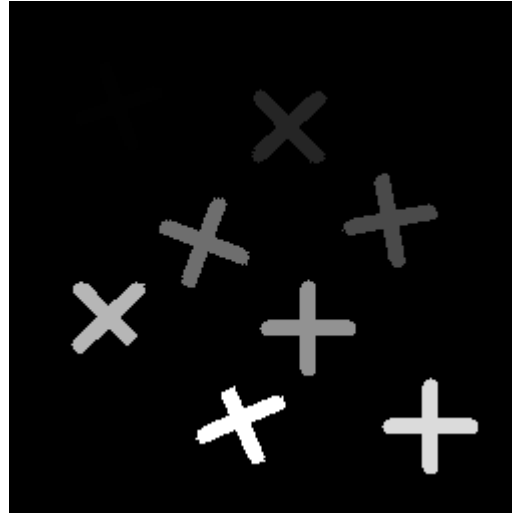
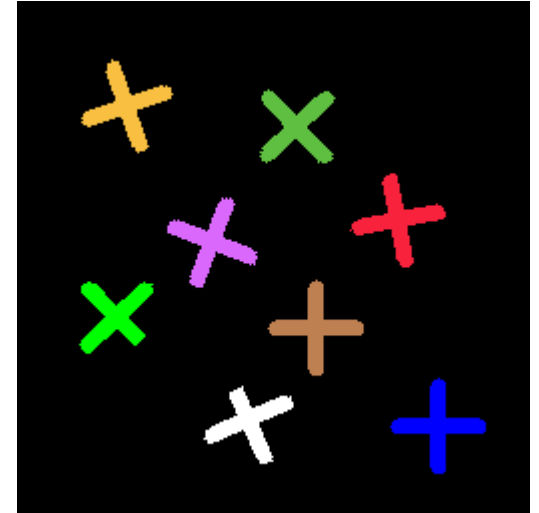


Imagen binaria



Etiquetas en grises



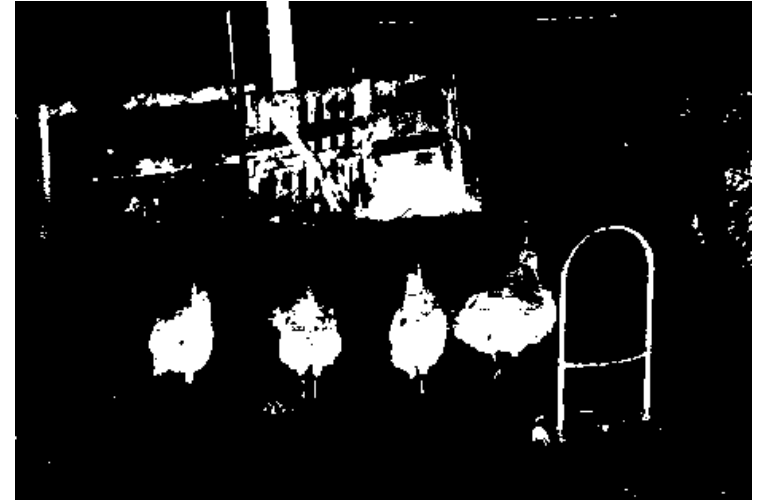
Etiquetas en colores



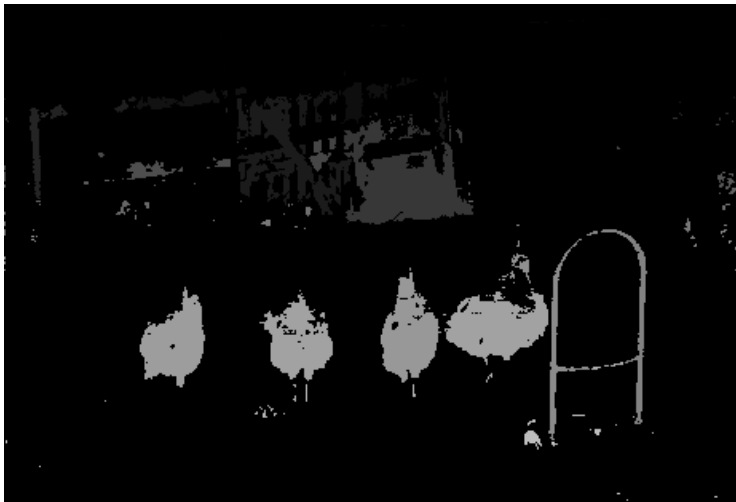
## Otros resultados:



**Imagen en niveles de gris**

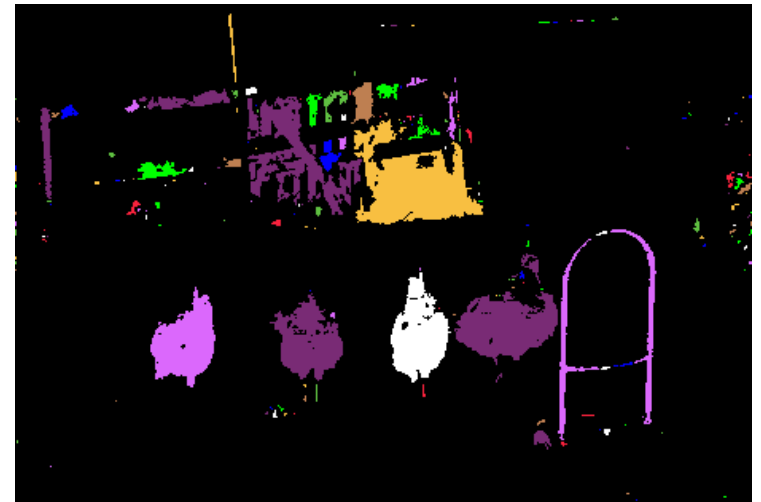


**Imagen binaria**



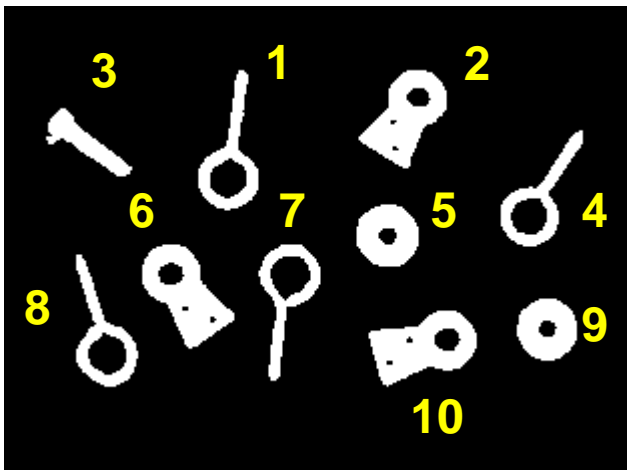
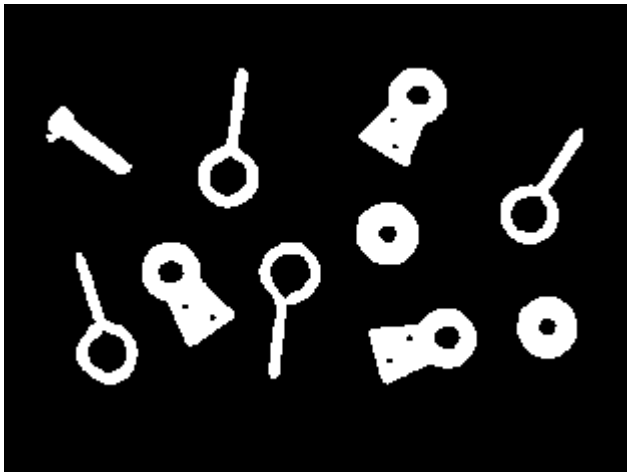
**Etiquetas en grises**

Muchas  
etiquetas  
ruidosas

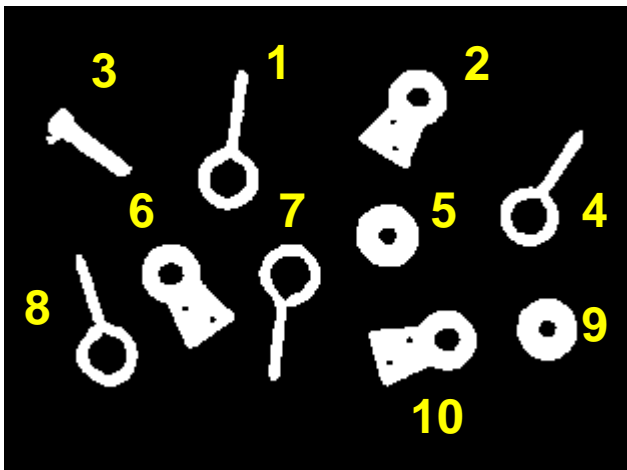
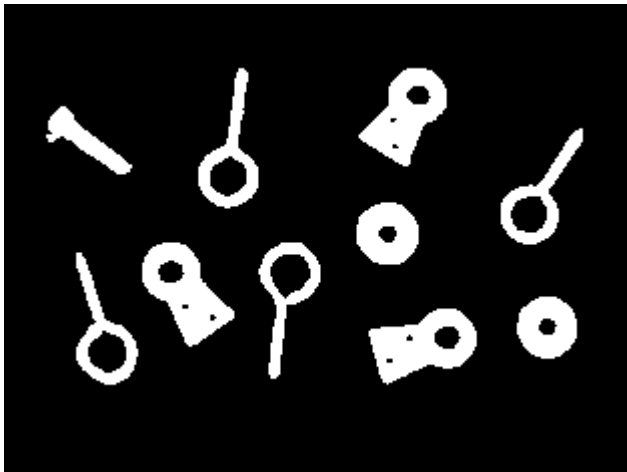


**Etiquetas en colores**

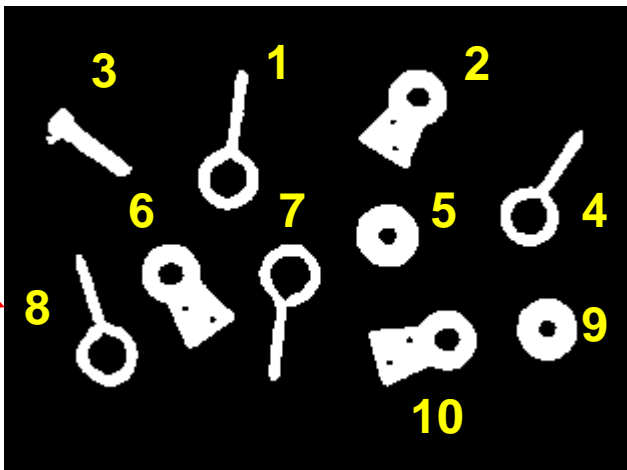
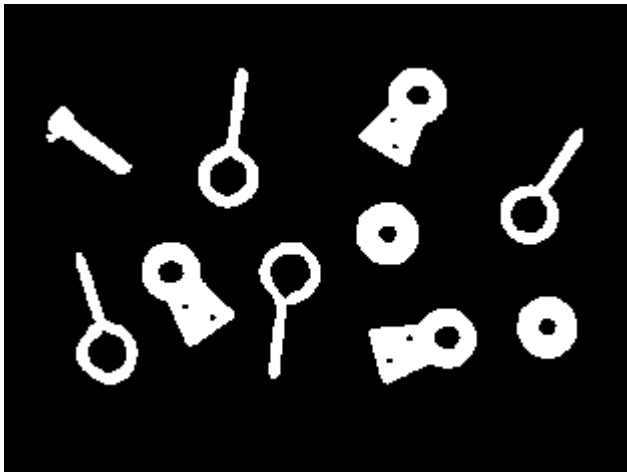
# **Resultado del proceso de etiquetado:**



**Ya vimos que al etiquetar  
los objetos en una  
imagen quedan como se  
muestra:**



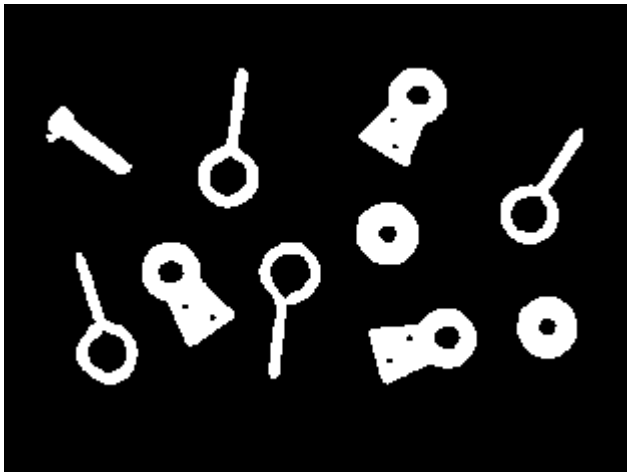
Queremos ahora  
distinguir entre objetos:



$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$



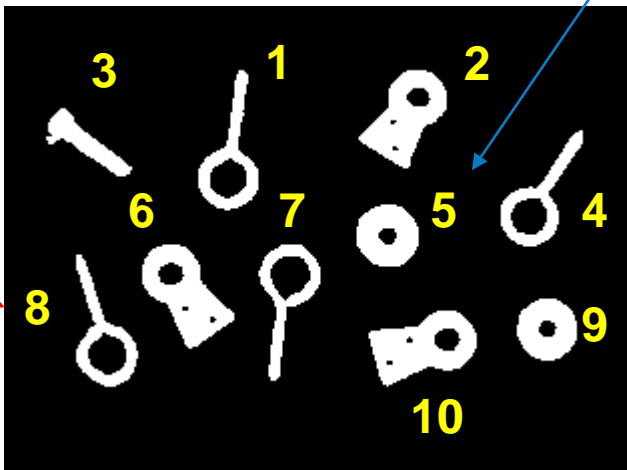
Para esto, describimos  
cada objeto en términos  
de un conjunto de  $n$   
rasgos:



Para esto, describiremos  
cada objeto en términos  
de un conjunto de  $n$   
rasgos:



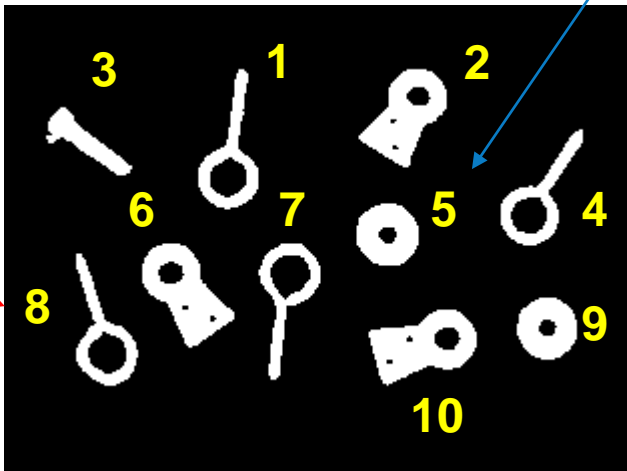
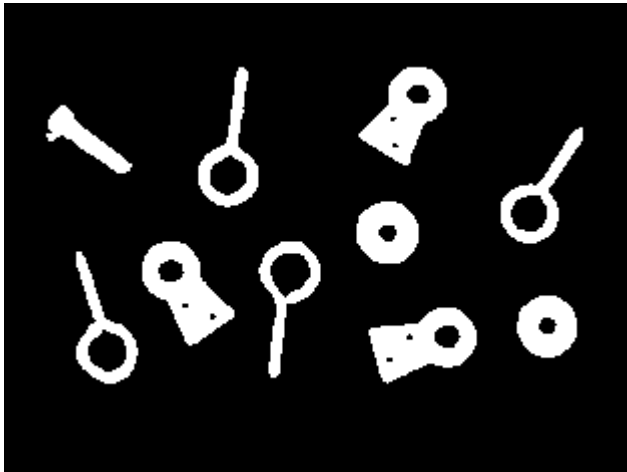
$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$



$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

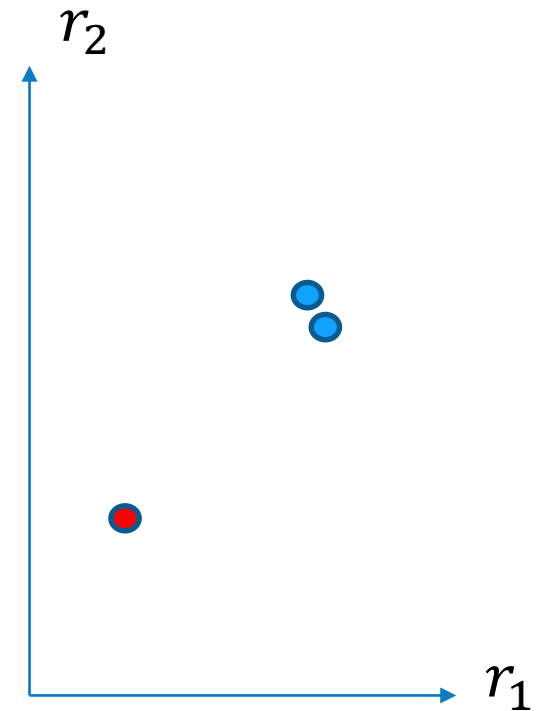
$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

En 2 dimensiones  
los objetos quedan descritos  
como puntos como  
se muestra:



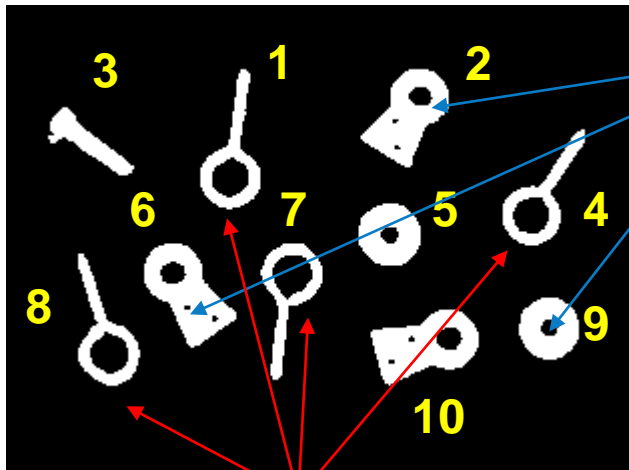
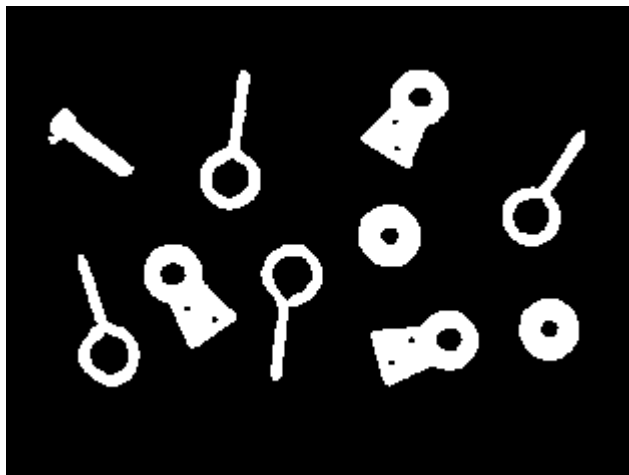
$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$



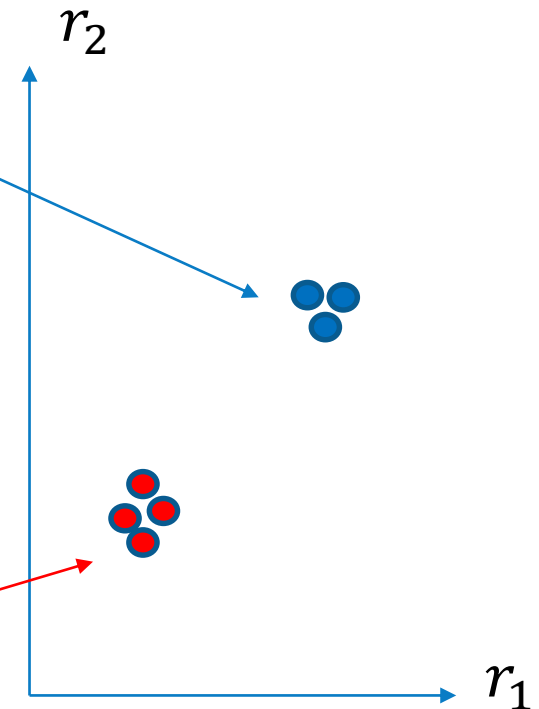
$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

Cuando se adquieren varias muestras descriptivas de un mismo objeto, estas aparecen como se muestra:



$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

$$\begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$





UNIVERSIDAD  
INTERNACIONAL  
DE LA RIOJA

**unir**

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