



# Tratamiento de imágenes

## Segmentación

**Héctor Alejandro Montes**

[h.a.montes@fi.uaemex.mx](mailto:h.a.montes@fi.uaemex.mx)

<http://fi.uaemex.mx/h.a.montes>

# Advertencia

No use estas diapositivas como referencia única de estudio durante este curso. La información contenida aquí es una guía para las sesiones de clase y de estudio futuro. Para obtener información más completa, refiérase a la bibliografía listada en la última diapositiva.

# **I. Detección de discontinuidades**

# Segmentación

Divide una imagen en las regiones u objetos que la forman.

Es considerada una de las tareas más complicadas, pero fundamental en procesamiento de imágenes

# Segmentación

Los algoritmos de segmentación se basan generalmente en dos estrategias:

## **Discontinuidad**

La partición de la imagen se obtiene a partir de cambios bruscos en la intensidad

## **Similaridad**

La partición se obtiene a partir de criterios predefinidos (ej: Thresholding)

# Detección de discontinuidades en escala de grises

Es común usar máscaras para esta tarea

M4	M3	M2
M5	M0	M1
M6	M7	M8

Máscara de  
gradiente

$$R0 = \sum_{i=0}^8 P_i \cdot M_i$$

R es una imagen  
intermedia, no la  
imagen Q de salida

Donde  $P_i$  son los píxeles de la  
imagen de entrada

# Detección de puntos

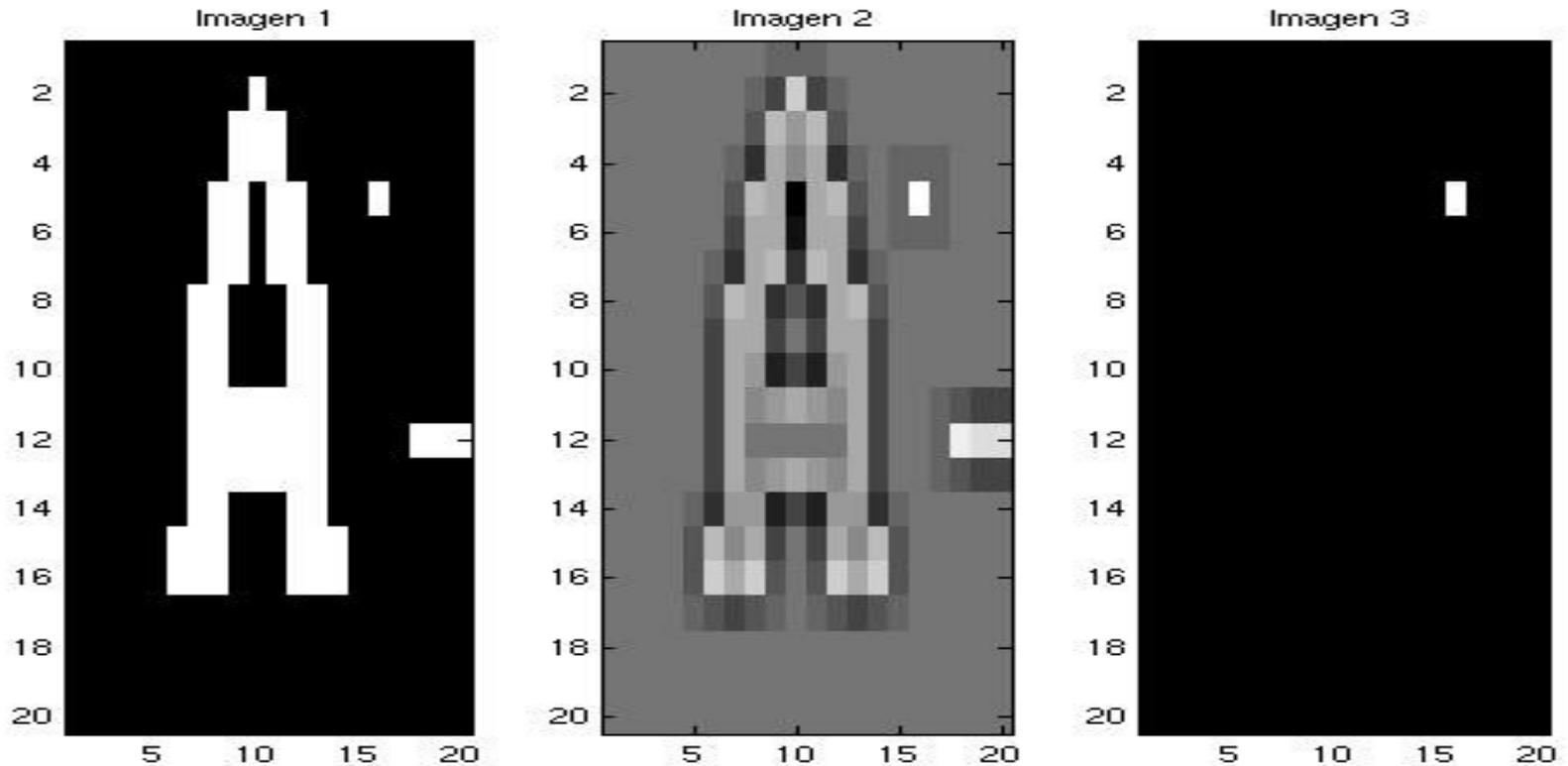
Se aplica la máscara de gradiente

-1	-1	-1
-1	8	-1
-1	-1	-1

Luego se buscan aquellos puntos que son mayores a un umbral dado

$$Q0(|R0|8)=1$$

# Detección de puntos por segmentación



# Detección de líneas

Se aplican las máscaras de gradiente

Horz.	<table border="1"><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>2</td><td>2</td><td>2</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	2	2	2	-1	-1	-1	<table border="1"><tr><td>-1</td><td>-1</td><td>2</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>2</td><td>-1</td><td>-1</td></tr></table>	-1	-1	2	-1	2	-1	2	-1	-1	45°
	-1	-1	-1																		
2	2	2																			
-1	-1	-1																			
-1	-1	2																			
-1	2	-1																			
2	-1	-1																			
Vert.	<table border="1"><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr></table>	-1	2	-1	-1	2	-1	-1	2	-1	<table border="1"><tr><td>2</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>2</td></tr></table>	2	-1	-1	-1	2	-1	-1	-1	2	-45°
-1	2	-1																			
-1	2	-1																			
-1	2	-1																			
2	-1	-1																			
-1	2	-1																			
-1	-1	2																			

# Detección de líneas

Y luego se busca aquellos que sean mayores que un umbral en algunas de las imágenes intermedias

$$Q_0(|R_H| \geq t \vee |R_V| \geq t \vee |R_{45}| \geq t \vee |R_{-45}| \geq t) = 1$$

Donde  $t$  es un umbral dado.

# Detección de líneas

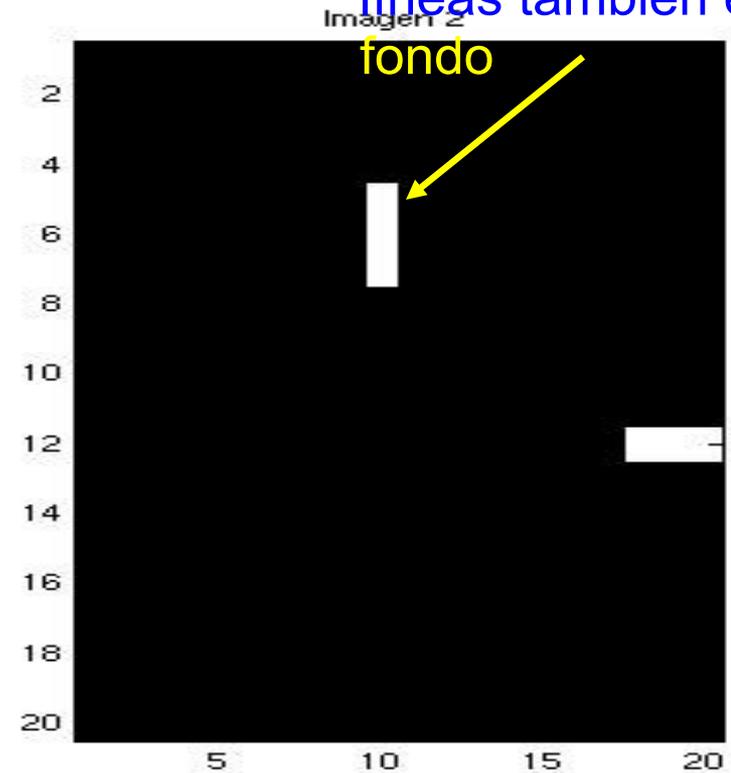
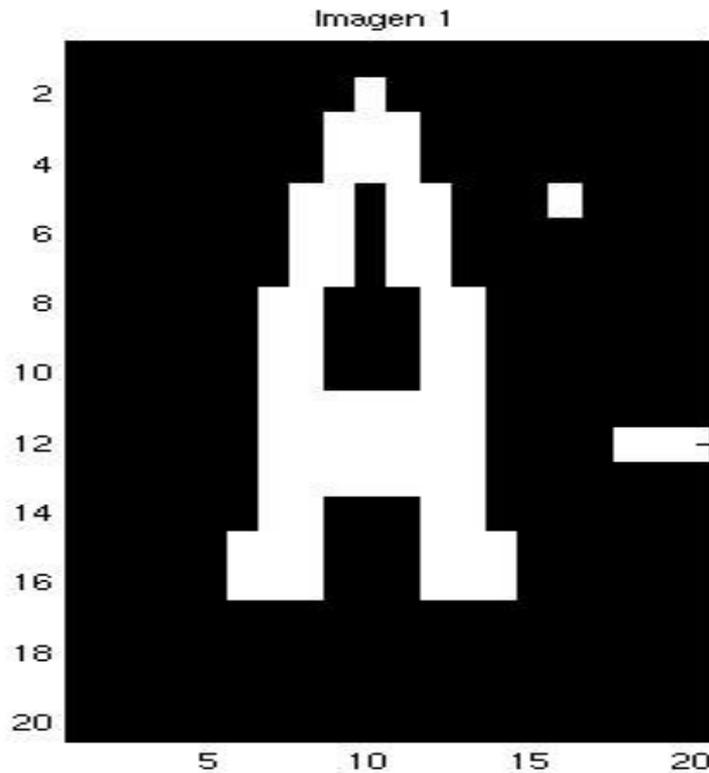
Este método sólo busca líneas de 1 píxel de grosor.

Podemos detectar sólo una dirección si se aplica sólo una máscara de gradiente, o sólo las direcciones que queramos

# Detección de líneas

Threshold=4

Observa que detecta líneas también en el fondo



## **II. Detección de Orillas**

# Detección de orillas

La detección de orillas es la forma más común de realizar la segmentación.

Existen muchas técnicas que podemos agrupar en dos tipos:

**Aproximación de plantillas** (template matching, **TM**)

**Gradiente diferencial** (differential gradient **DG**)

# Detección de orillas

Ambas **TM** y **DG** buscan encontrar donde la magnitud del **gradiente g** es suficientemente grande para indicar un filo.

**TM** es más sencilla y más rápida pero más inexacta que **DG**

# Detección de orillas

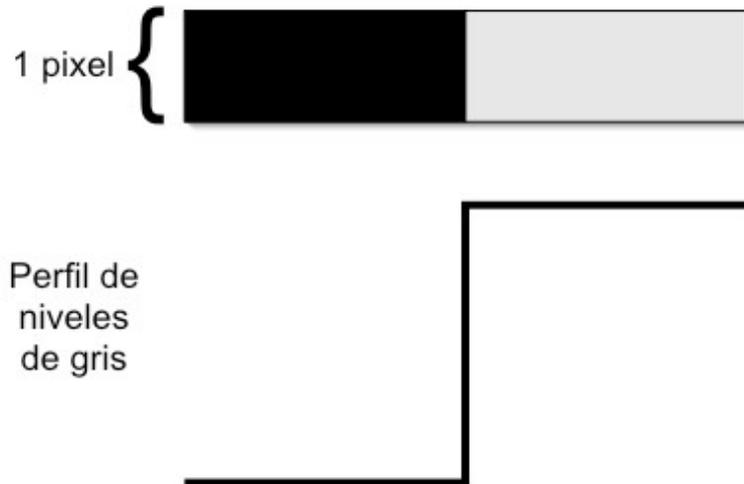
## Perfil de niveles de gris

Función continua que representa la intensidad del gris a lo largo de una línea de píxeles (no necesariamente horizontal)

Si fuese ideal sería un escalón

Lo normal es que sea una pendiente

# Detección de orillas



La pendiente es inversamente proporcional al grado de difuminado (blur) de la imagen

# Detección de orillas

El grosor de la orilla, no es de 1 píxel, sino que depende de la pendiente

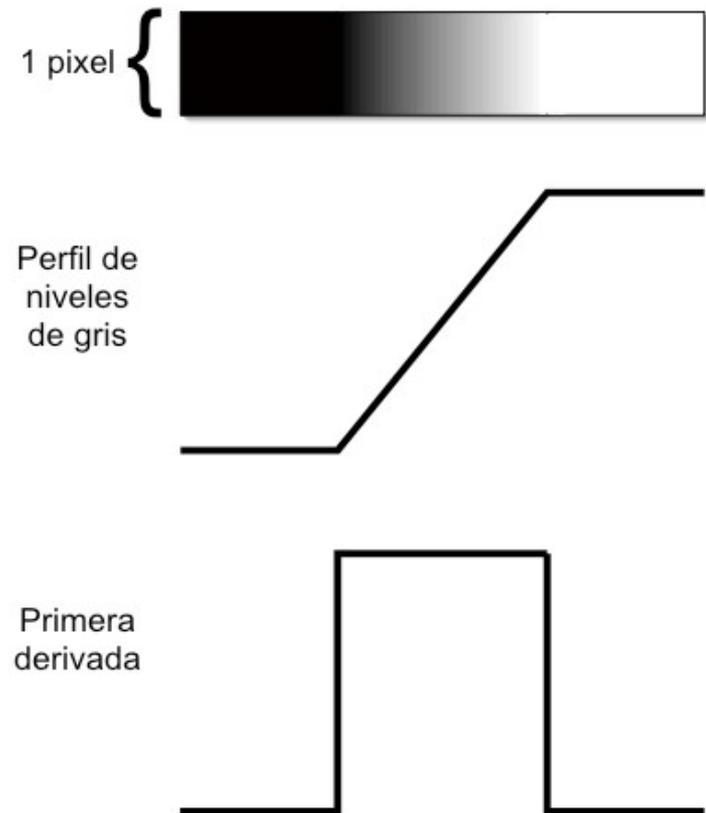
Imágenes más difuminadas resultan en orillas más gruesas

Imágenes más nítidas resultan en orillas más delgadas

La orilla es un conjunto de píxeles conectados.

# Detección de fillos

El gradiente de la función de perfil de niveles de gris permite detectar cuando hay una discontinuidad fuerte, i.e. Donde hay una orilla.



# Detección de orillas por TM

Se aplican  $n$  (8 y 12) máscaras de convolución

La magnitud del gradiente  $g$  se calcula como el máximo de las distintas máscaras

$$g = \max(g_i, i=1..n)$$

La orientación de la orilla se estima simplemente como la de la máscara que produce el valor máximo

# Detección de orillas

## Máscaras de gradiente comunes

Prewitt

Kirsch

Robinson

# Máscaras de gradiente para TM

Prewitt

135°

1	1	1
1	-2	-1
1	-1	-1

90°

1	1	1
1	-2	1
-1	-1	-1

45°

1	1	1
-1	-2	1
-1	-1	1

0°

-1	1	1
-1	-2	1
-1	1	1

180°

1	1	-1
1	-2	-1
1	1	-1

225°

1	-1	-1
1	-2	-1
1	1	1

270°

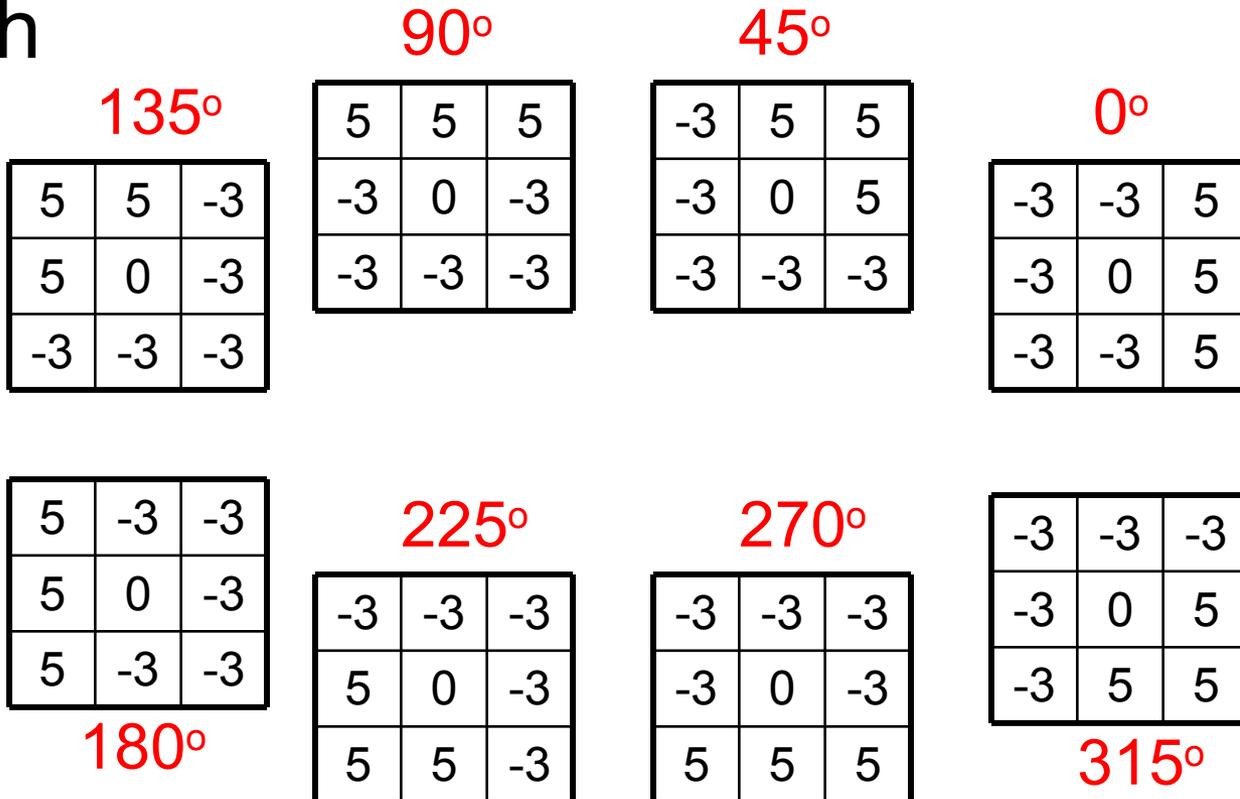
-1	-1	-1
1	-2	1
1	1	1

315°

-1	-1	1
-1	-2	1
1	1	1

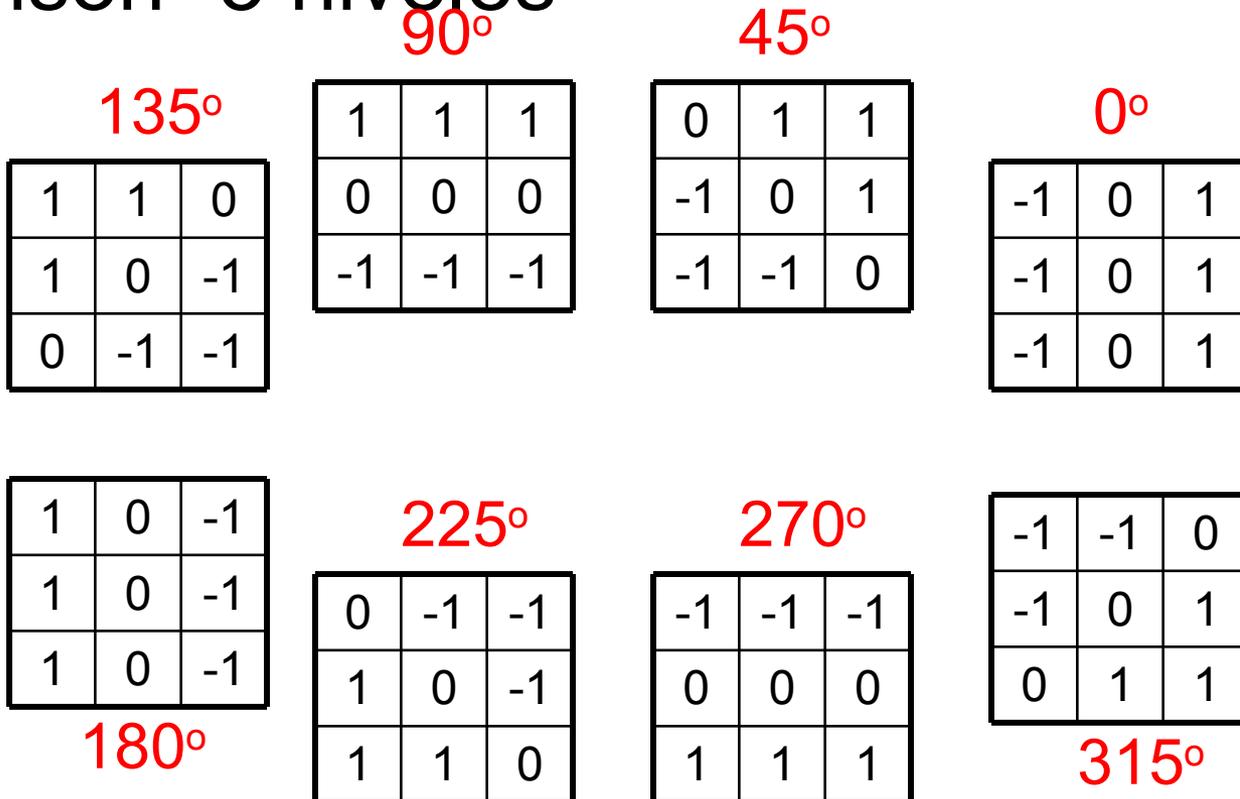
# Máscaras de gradiente para TM

Kirsch



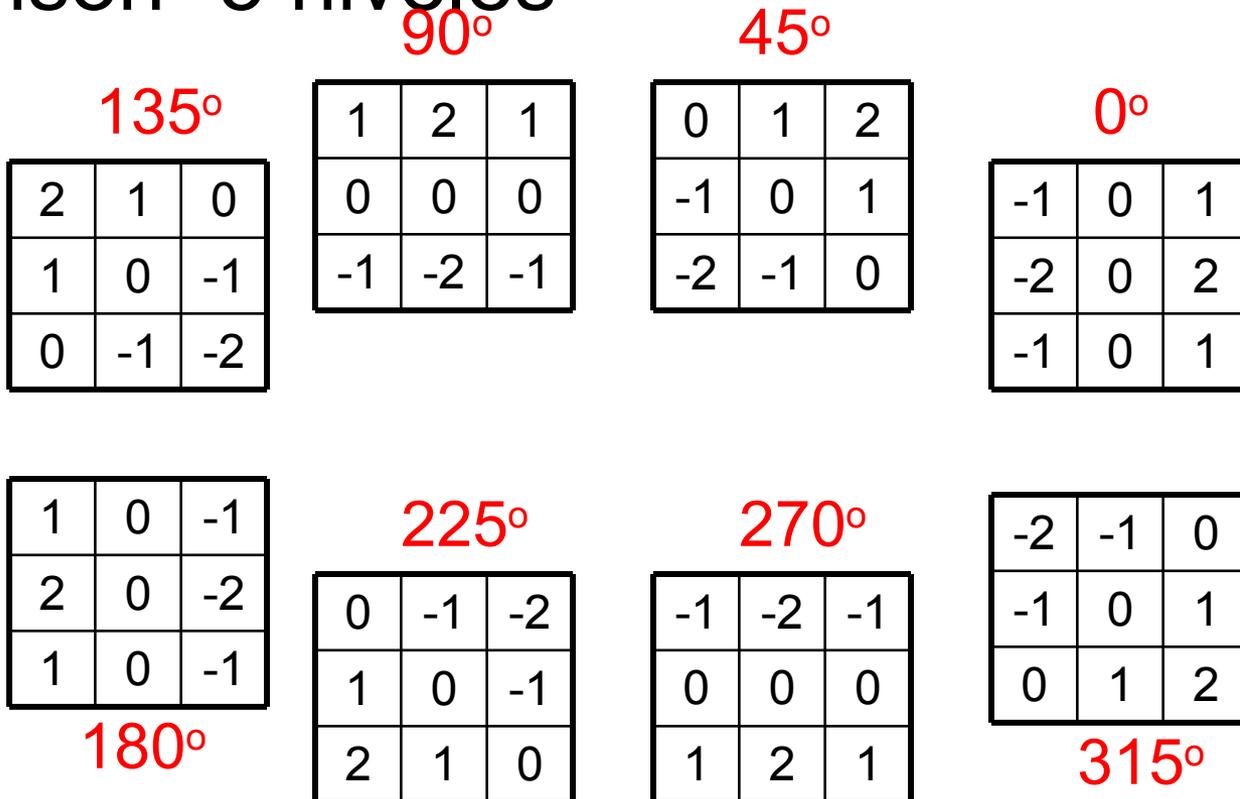
# Máscaras de gradiente para TM

Robinson “3 niveles”



# Máscaras de gradiente para TM

Robinson “5 niveles”



# Detección de orillas por DG

La magnitud del gradiente  $g$  se calcula de forma vectorial con la transformación

$$g = \sqrt{g_x^2 + g_y^2}$$

# Detección de orillas por DG

La orientación de la orilla se estima por la ecuación

$$\theta = \arctan(g_y / g_x)$$

# Detección de orillas

## Máscaras de gradiente comunes

Roberts

Prewitt

Sobel

# Máscaras de gradiente para DG

Roberts

$$G_x. \begin{array}{|c|c|} \hline -1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|c|} \hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array} Gy.$$

No tienen un centro claramente definido

# Máscaras de gradiente para DG

Prewitt

Gx Horz

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

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

45°

Gy Vert.

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

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

-45°

# Máscaras de gradiente para DG

## Sobel

Gx Horz.

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

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

45°

Gy Vert.

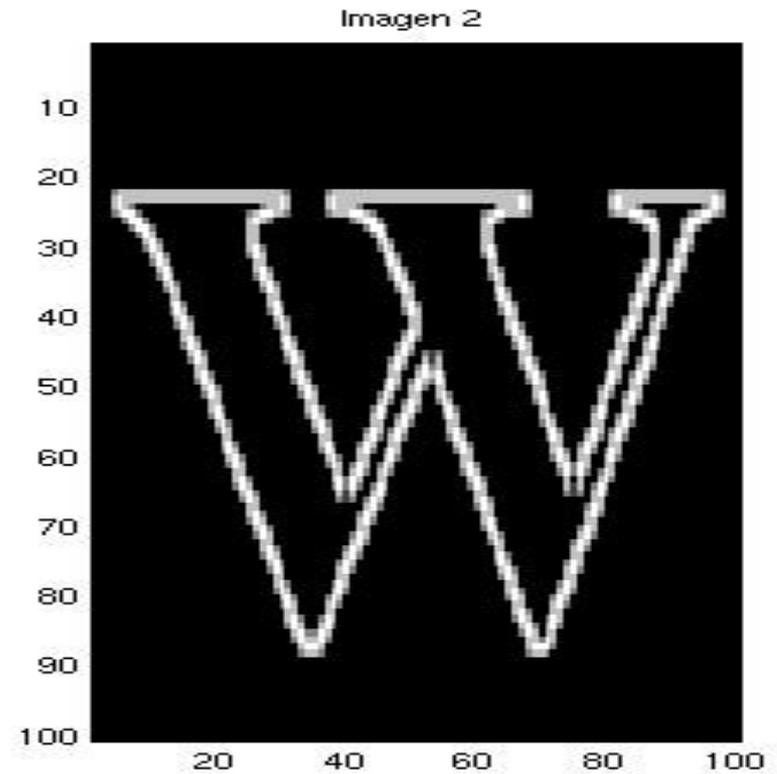
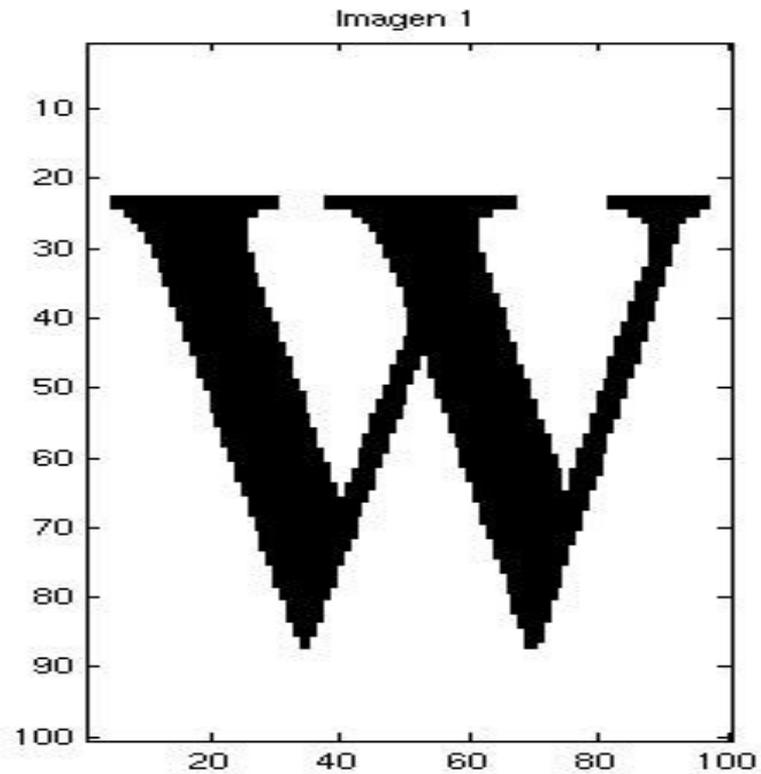
-1	0	1
-2	0	2
-1	0	1

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

-45°

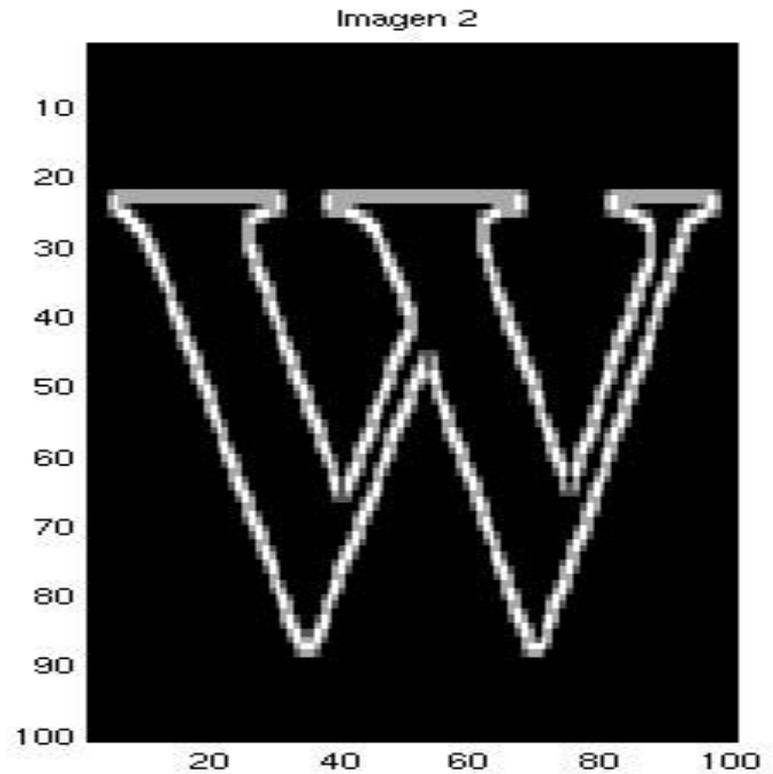
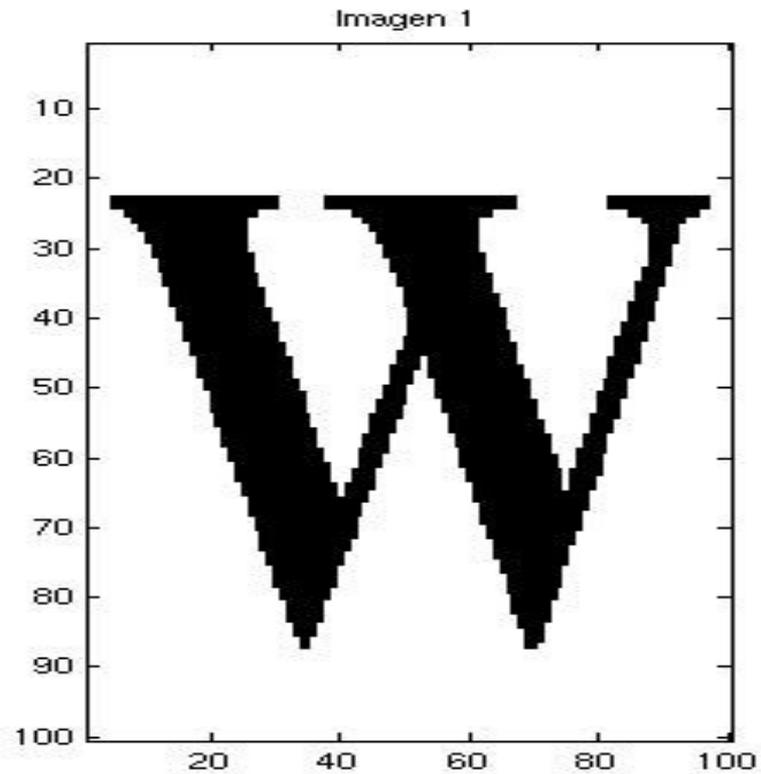
# Gradiente

Prewitt



# Gradiente

Sobel



# Gradiente

Prewitt

Imagen 1

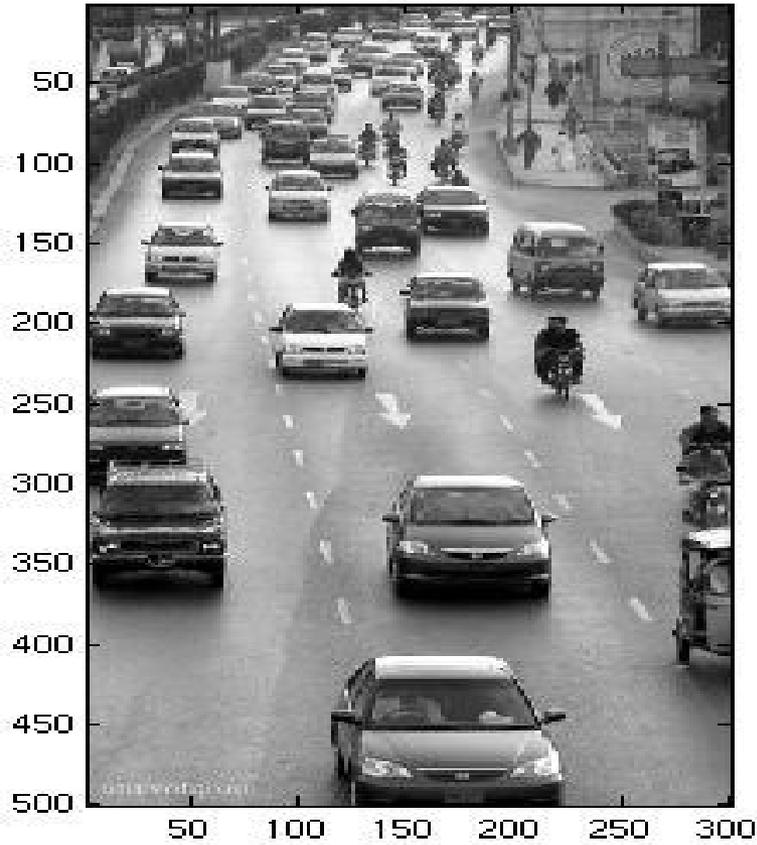
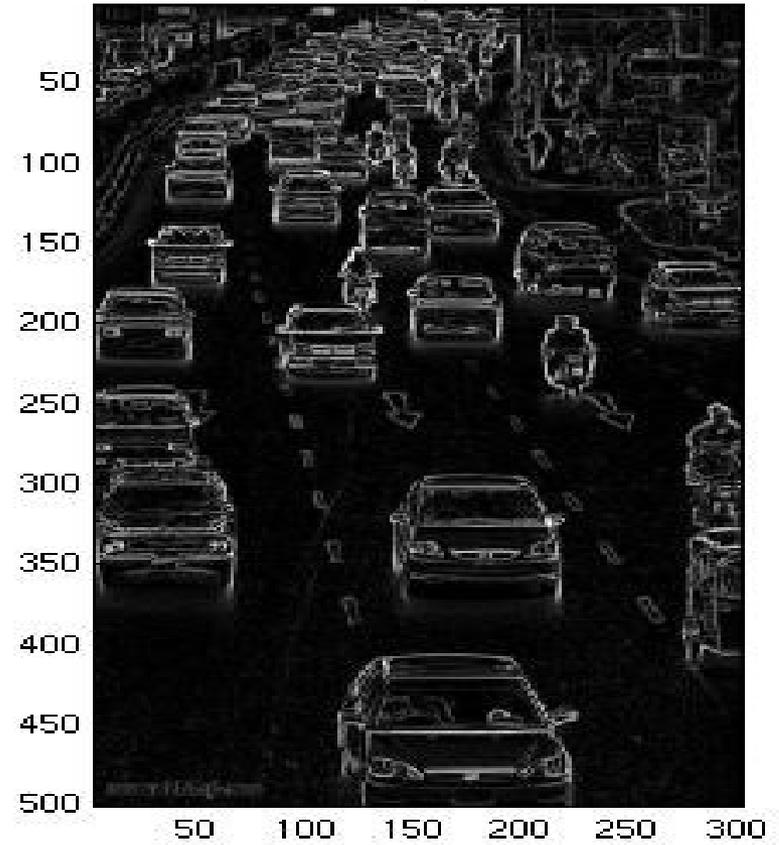


Imagen 2



# Gradiente

## Sobel

Imagen 1

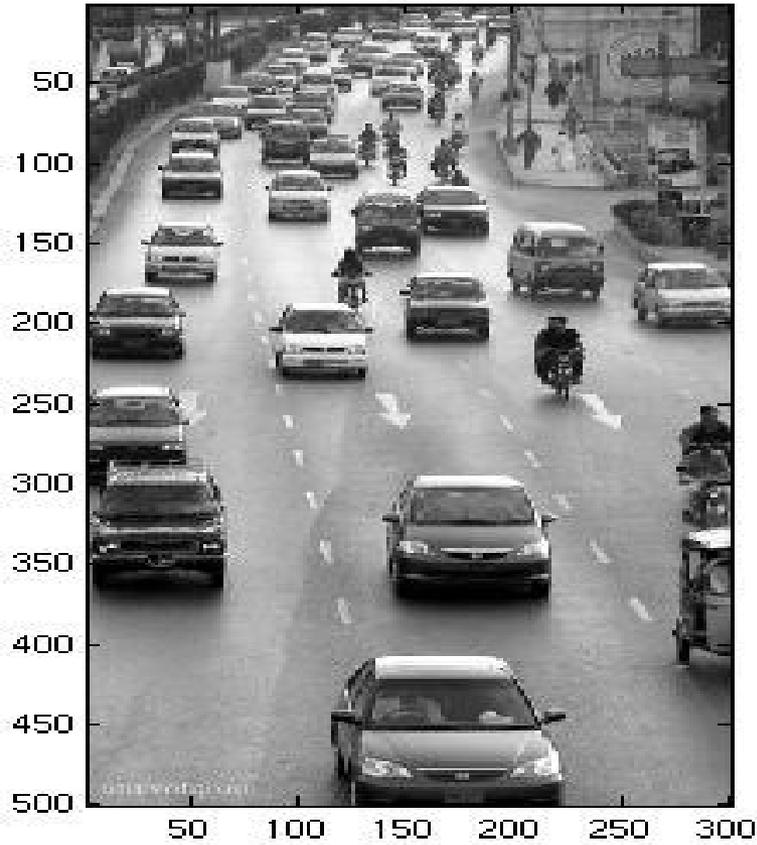
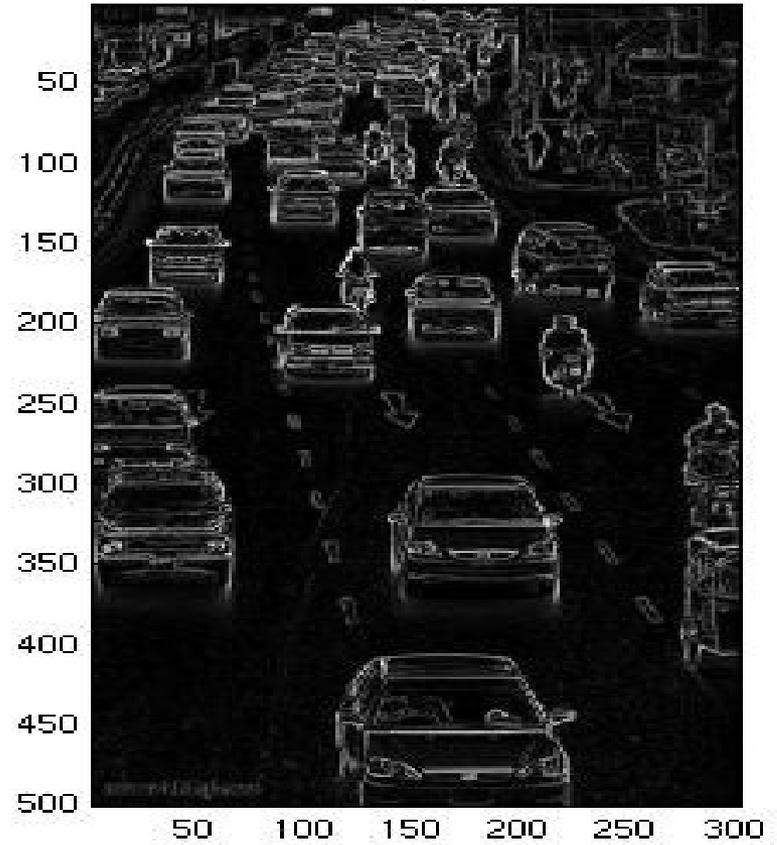
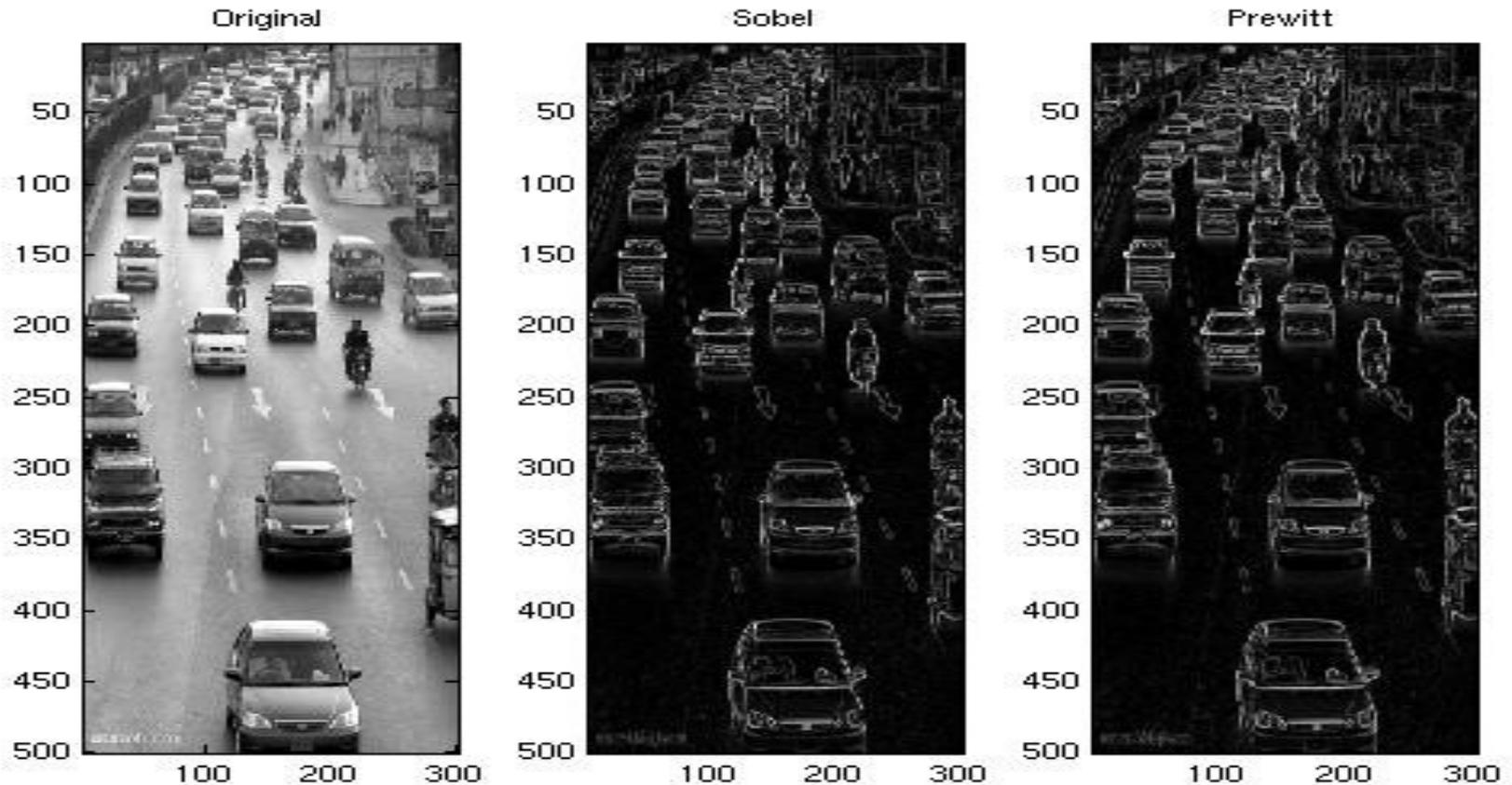


Imagen 2



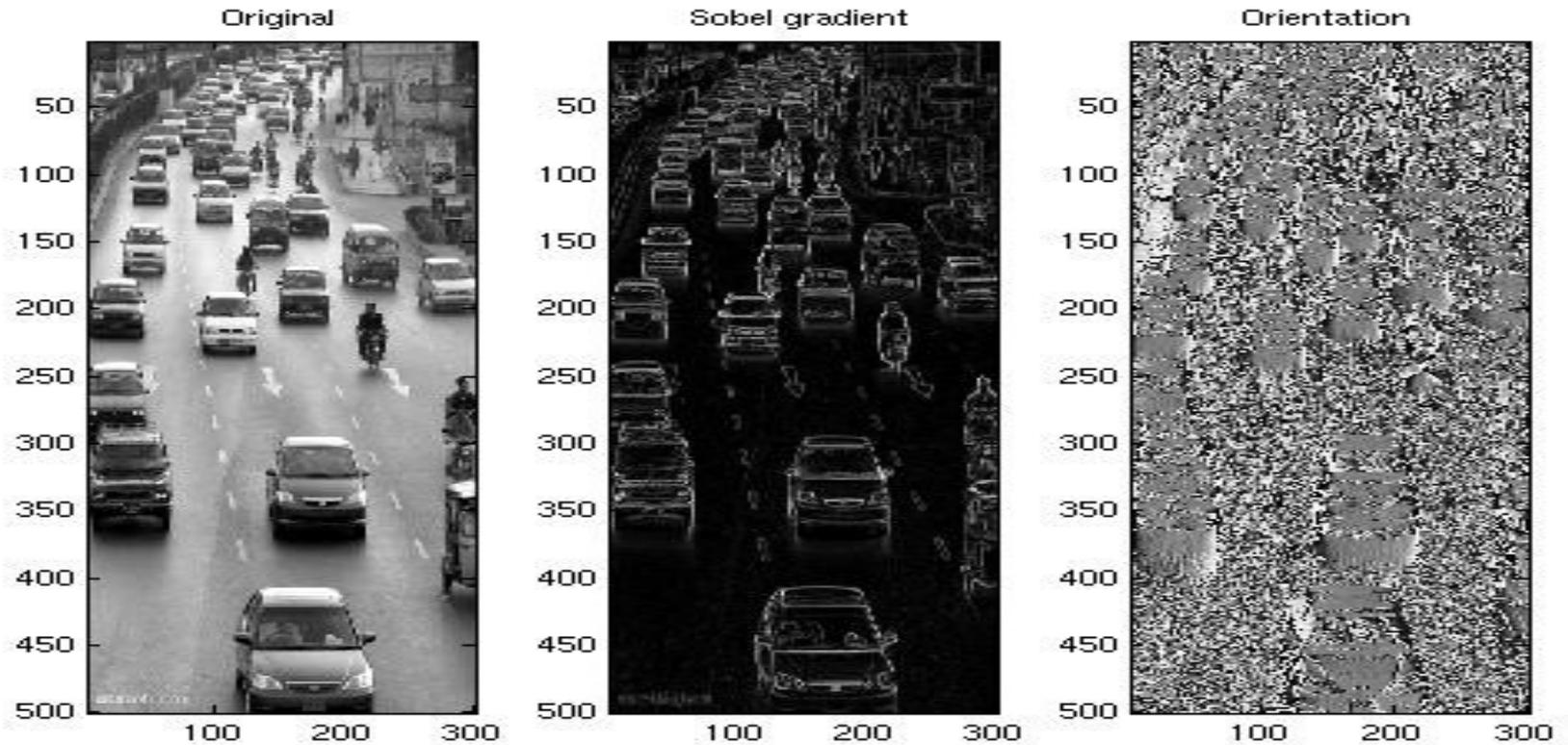
# Gradiente

## Prewitt vs Sobel



# Gradiente

## Orientation



## **III. Umbrales (thresholds)**

# Segmentación por umbrales

Recordatorio:

La operación de thresholding convierte una imagen a binaria.

$$\text{THRESH} = [[Q0 = P0 < \text{thresh} ? 0 : 1]]$$

# Segmentación por umbrales

Thresholding es la forma más básica de segmentar o partir una imagen en regiones.

La clave de una buena segmentación por thresholding es la elección del umbral

# Segmentación por umbrales

La forma de determinar el umbral puede ser:

**Local:** La imagen se subdivide en regiones

**Global:** Se busca el umbral en toda la imagen

La forma de determinar el umbral puede ser:

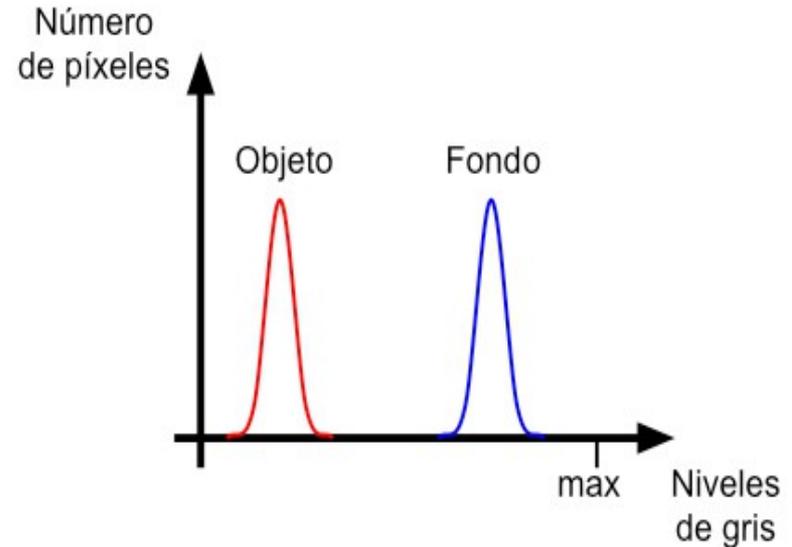
**Fijo:** Todos los píxeles comparten los umbrales

**Adaptativo:** Los umbrales dependen de la localización del píxel

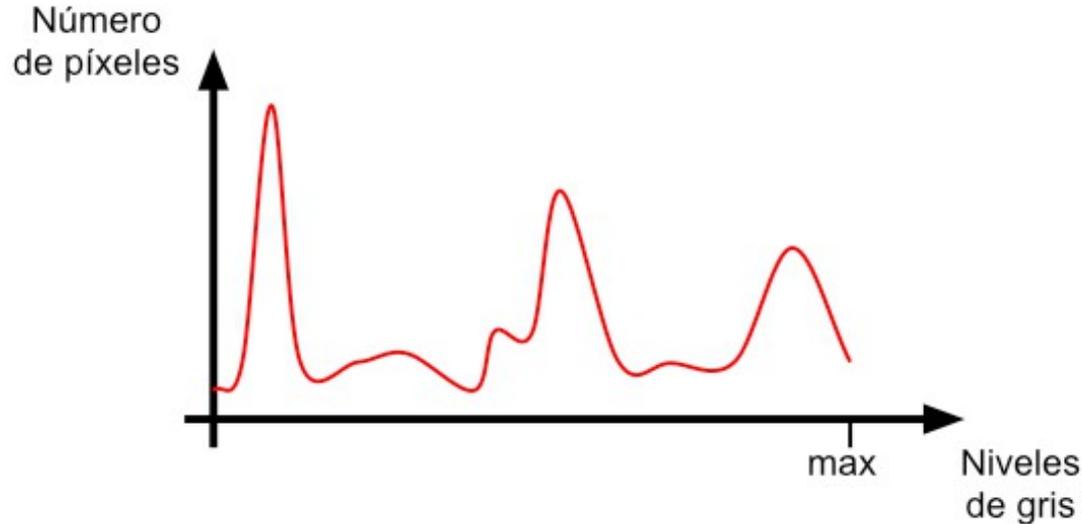
# Segmentación por umbrales

En el caso más simple, el histograma del objeto y del fondo están claramente separados...

Basta tomar como umbral algún punto intermedio entre los dos bloques de píxeles



# Segmentación por umbrales



En un caso un poco más general, habrá **varias modas** (más de 2) en el histogramas...

Se puede entonces usar múltiples umbrales (**multiple thresholding**)

# Segmentación por umbrales

Es difícil encontrar histogramas “ideales”

Uno de los grandes problemas es la  
**iluminación no uniforme**

# Segmentación por umbrales

Cuando no es tan sencillo encontrar un umbral “a mano” es bueno contar con un **algoritmo que halle el mejor umbral de forma automática...**

Existen varios algoritmos de este tipo, revisaremos aquí dos de ellos:

Umbral básico global o búsqueda automática del umbral (**global basic thresholding**)

Umbral óptimo (**optimum thresholding**)

# Búsqueda automática de umbral

Es una técnica que busca el mejor umbral posible de forma global

Desafortunadamente **sólo es válido para un histograma bimodal** (con un valle)

Podemos encontrar el algoritmo con ligeras variaciones

Este sitio ofrece un buen ejemplo(que es el que vamos a ver):  
<http://www.generation5.org/content/2003/segmentation.asp>

# Búsqueda automática de umbral

Toma un umbral inicial,  $t$  (por ejemplo a la mitad del histograma).

Calcula las medias del histograma por debajo ( $m1$ ) y por encima ( $m2$ ) del umbral  $t$ .

Calcula el nuevo umbral como:

$$t_{new} = (m1 + m2) / 2.$$

Si el umbral se estabilizó ( $t = t_{new}$ ), entonces este es el mejor umbral. Sino, asigna  $t_{new}$  a  $t$  y reitera desde el paso 2.

# Búsqueda automática de umbral

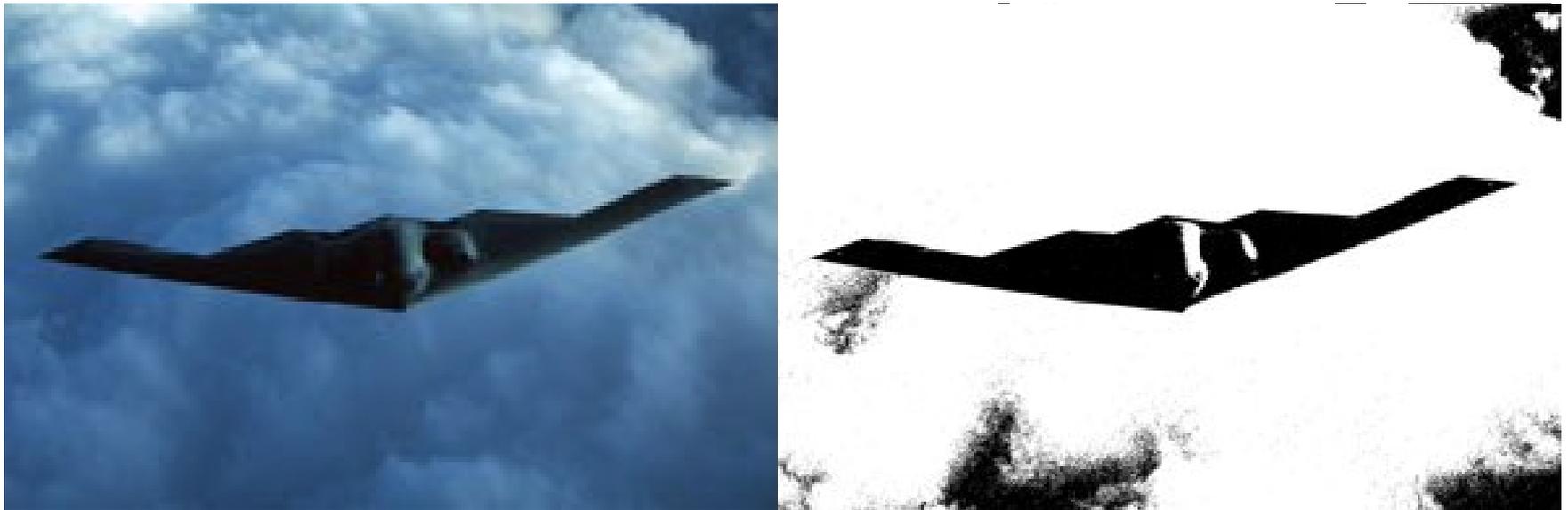
Imagen obtenida de:  
[www.generation5.org](http://www.generation5.org)



Umbral fijado a mano en 128 (sobre 256 colores)

# Búsqueda automática de umbral

Imagen obtenida de:  
[www.generation5.org](http://www.generation5.org)



Umbral fijado automáticamente en 74 (sobre 256 colores)

# Umbral óptimo - Chow y Kaneko

Desarrollada por Chow y Kaneko en 1972

Chow, C. K. And Kaneko, T. “*Automatic boundary detection of the left ventricle from cineangiograms*”  
Computers and Biomedical Research Vol 5 no 4  
pp 338-410 AUG (1972)

Es una técnica local!

Busca el umbral por cada región de la imagen...

# Umbral óptimo - Chow y Kaneko

Divide la imagen en regiones pequeñas con cierta superposición.

Busca las regiones que no sean unimodales (aquellas que tienen más de una región diferenciada)

En las regiones multimodales aplica un algoritmo de búsqueda de umbral automático vía un ajuste a distribuciones Gaussianas de las diferentes modas.

Observa que tendrás umbrales adaptativos (i.e. dependientes de la localización del píxel)

# Algunas otras técnicas para búsqueda del umbral

Tener en cuenta no sólo el histograma sino también gradiente (scattergram)

Da lugar a varias técnicas

Buscar “valles” en la distribución de intensidades con parámetros no necesariamente igual al gradiente

Buscar picos en regiones con un alto gradiente.

Basadas en entropía

Basadas en varianza

Basadas en máxima probabilidad (maximum likelihood)

De relajación al seleccionar el umbral

Iterativas y/o recursivas

Otras...

# IV. Trazado de contornos

# Trazado de contornos

La aplicación de máscaras de gradiente busca identificar píxeles pertenecientes a filos.

Estos píxeles no caracterizan un contorno

- Por presencia de ruido

- Por roturas

- Por iluminación no uniforme

- Por otros efectos...

# Trazado de contornos

Para trazar el contorno a partir de los píxeles en los fillos se aplican algoritmos de enlace (**linking**)

El enlace puede ser

Local

Global

# Trazado de contornos

## Local:

Se busca en la vecindad ( $3 \times 3$ ,  $5 \times 5$ , ...) de un píxel  $(x, y)$  atendiendo a **dos criterios**

Gradiente

Orientación

Un punto  $(x_0, y_0)$  en la vecindad de  $(x, y)$  está enlazado a  $(x, y)$  si cumple ambos criterios

# Trazado de contornos

## Local: Criterio del Gradiente:

Se busca que el pixel  $(x_0, y_0)$  en la vecindad de  $(x, y)$  sea de magnitud del gradiente similar a  $(x, y)$

La magnitud del gradiente de  $(x_0, y_0)$  es similar a la magnitud del gradiente de  $(x, y)$  si

$$|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq \varepsilon$$

donde  $\varepsilon$  es un umbral dado no negativo

# Trazado de contornos

## **Local:** Criterio de la Orientación:

Se busca que el pixel  $(x_0, y_0)$  en la vecindad de  $(x, y)$  tenga un ángulo de orientación similar a  $(x, y)$

El ángulo de orientación de  $(x_0, y_0)$  es similar al ángulo de orientación de  $(x, y)$  si

$$|\alpha(x, y) - \alpha(x_0, y_0)| \leq \alpha$$

donde  $\alpha$  es un umbral angular dado

# Trazado de contornos

**Global:** Mediante la transformada de Hough

Veremos la transformada de Hough en el último tema (opcional), así que por ahora queda pendiente...

# V. Segmentación orientada a regiones

# Segmentación orientada a regiones

Como se indicó al comienzo del tema, los algoritmos de segmentación se basan generalmente en dos estrategias:

**Discontinuidad** (segmentación por cambios bruscos en la intensidad)

**Similaridad** (segmentación por criterios predefinidos)

La **segmentación orientada a regiones** en lugar de buscar los filos busca encontrar las regiones directamente buscando algún criterio de **similaridad**

# Region Growing

Se toman unos puntos que serán las “semilla” de las distintas regiones

Crece por dilatación los puntos hasta alcanzar un criterio de parada previamente definido

Si dos regiones al crecer comparten puntos las regiones se unen en una sola.

# Region Growing

Para elegir las semillas se puede optar por diferentes formas:

aleatoria (no muy buena),  
con conocimiento del problema,  
a mano,  
por búsqueda de clusters (centroides),  
etc...

# Region Growing

El criterio de parada depende del problema,  
pero puede tomar entre otras la forma de:

- Conectividad (este es obligatorio, aunque normalmente se usa en conjunción con algún otro...)
- un nivel de intensidad predeterminado,
- descriptores de forma,
- Textura,
- “historia” -cómo se ha ido formando la región
- Etc...

# Region Growing

Puedes encontrar algo más de información sobre Region Growing en:

Zucker, S. W. “*Region growing: childhood and adolescence*”. Computer Graphics Image Processing Vol 5 pp190-202 (1976)

# VI. Otras segmentaciones

# Otras segmentaciones

Basada en color

Por marca de agua (watershed)

Por movimiento (motion)

Basada en frecuencia

Otras

# Segmentación por color

Considerada una de las tareas más difíciles en procesamiento de imágenes

Separa las distintas bandas de color o filtros de la imagen

Realiza la segmentación en una de las bandas como si fuese segmentación en escala de grises

Muy utilizada en imágenes multispectrales (ej: imágenes satelitales)

# Segmentación por color

Existen técnicas de detección de fillos en color

Buscan un gradiente en algún espacio de color (HSI, RGB...) que no es necesariamente en el que esté almacenada la imagen

Para saber más: Cumani, A “*Edge detection in multispectral images*” Reporte Técnico 373 Instituto Electrónico Nacional “Galileo Ferraris” Italia Apr (1989)

# Segmentación por color

También se puede segmentar atendiendo a texturas:

Normalmente se usan wavelets (que no es más que la serie de Fourier donde las bases ya no son senos y cosenos)

# Segmentación por color

Puedes encontrar un buenos ejemplos de segmentación por color en:

Mirmehdi, Mahid and Petrou, Maria “*Segmentation of Color Textures*” IEEE Transactions on Pattern Analysis and Machine Intelligence vol 22. No. 2 pp 142-159 (2000)

Starbek, Wladyslaw and Koschan, Andreas “*Colour Image Segmentation - A survey*” Reporte Técnico 94-32 Universidad Técnica de Berlín

# Segmentación por marca de agua (watershed)

La imagen se entiende como una superficie donde la altura de la superficie es la intensidad del pixel.

Se “inyecta agua” por los mínimos locales, para ir rellenando la superficie.

No se permiten que se junten diferentes depósitos de agua, sino que se construyen “presas” para evitar que el agua salte de un valle a otro.

Cuando se ha cubierto la superficie completa, las líneas que conforman las “presas” forman las regiones de la imagen

# Segmentación por marca de agua (watershed)

Puedes encontrar un buen ejemplo de este tipo de segmentación en [Gonzalez y Woods]