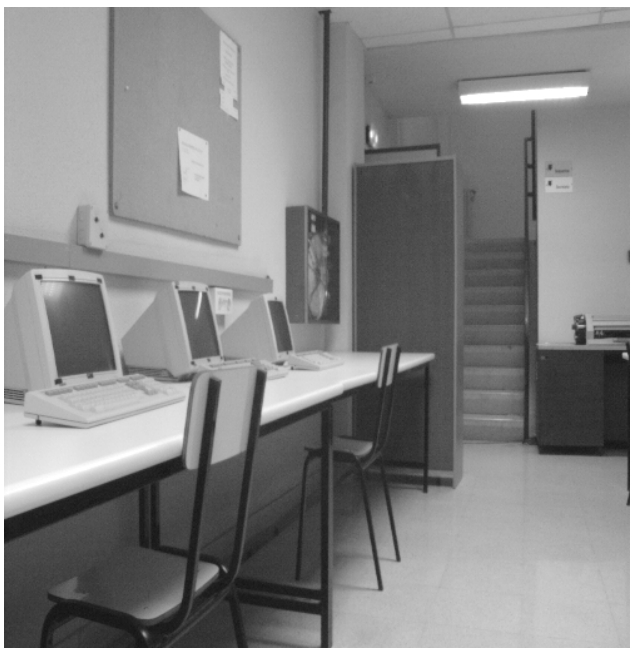
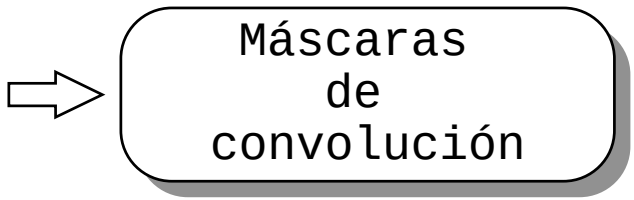
Detección de contornos

- Extracción de contornos: Separación de elementos significativos de la superficie de los objetos
- Los contornos resumen la mayor parte de la información de una imagen



Extracción de Contornos

- Detección de puntos de contorno
  - Cambios de luminosidad en la imagen
    - » Máximos del Gradiente (1ª derivada de la luminosidad)
    - » Cruces por cero del Laplaciano (2ª derivada de la luminosidad)

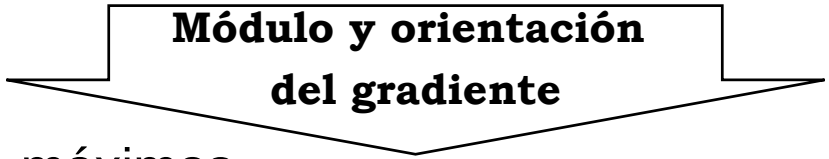


- Segmentación de contornos
  - Agrupación de puntos de contorno
  - Eliminación de ruidos

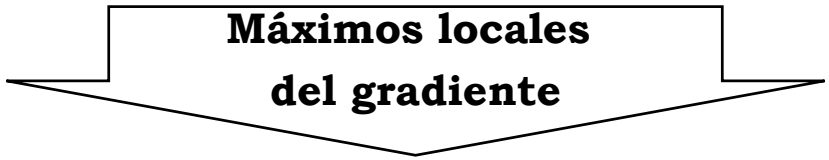
- Ajuste de rectas o curvas

Ejemplo: Contornos de Canny

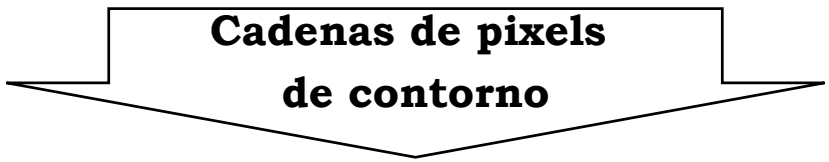
- Operador de Canny: derivada de la Gaussiana



- Detección de máximos



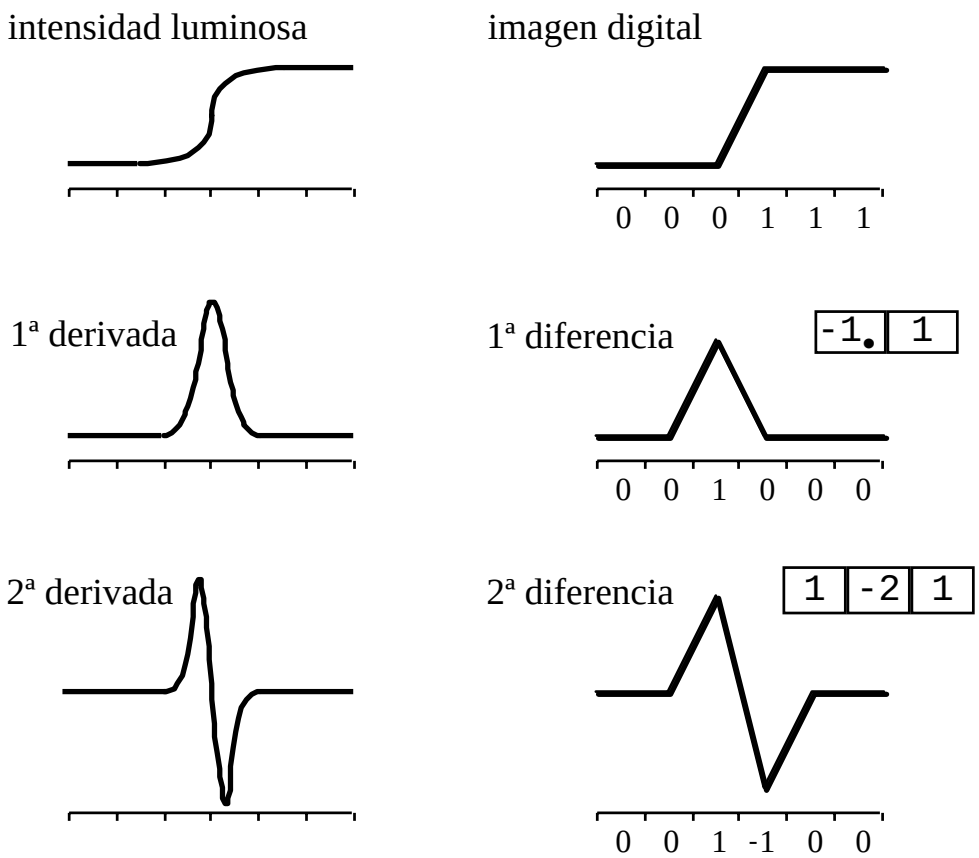
- Segmentación de contornos:
  - Seguimiento local + Umbral con histéresis



- Ajuste de rectas o curvas

Detección de Puntos de Contorno

- Máximo local del gradiente (1ª derivada)
- Cruce por cero del Laplaciano (2ª derivada)



Detección con el Gradiente

- Gradiente:

Gradiente $\vec{\nabla} f(x,y) = (\nabla_x, \nabla_y) = \left(\frac{\partial f}{\partial x} \quad , \quad \frac{\partial f}{\partial y} \right)$

Módulo $|\nabla f| = \sqrt{(\nabla_x)^2 + (\nabla_y)^2} \approx |\nabla_x| + |\nabla_y|$

Orientación $\theta = \operatorname{atan} 2(\nabla_x, \nabla_y)$

- Operadores:

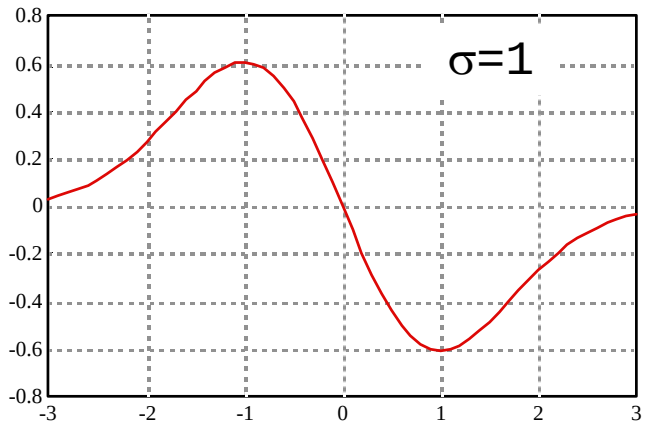
	∇_x	∇_y
1ª *diferencia*	$\begin{bmatrix} -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Dividir por la suma de los valores positivos

Gradiente: Operador de Canny

- Optimiza tres criterios:
  - Robustez de detección frente al ruido
  - Precisión en la localización
  - Unicidad de la respuesta
- Similar a la derivada de una Gaussiana

$$G'_\sigma(x) = \frac{-x}{\sigma^2}\exp\left(\frac{-x^2}{2\sigma^2}\right)$$



- Tamaño de la máscara: 5σ–7σ
- Ejemplo, para σ = 1

0.2707	0.6065	0.0	-0.6065	-0.2707

Dividir por la suma de los valores positivos

Gradiente: Operador de Canny

- Cálculo de gradiente en x e y
  - Pueden utilizarse máscaras separadas

$$\nabla_x = G'_\sigma(x) * G_\sigma(y) * f(i,j)$$
$$\nabla_y = G_\sigma(x) * G'_\sigma(y) * f(i,j)$$

  - Para el caso $\sigma = 1$, n = 5

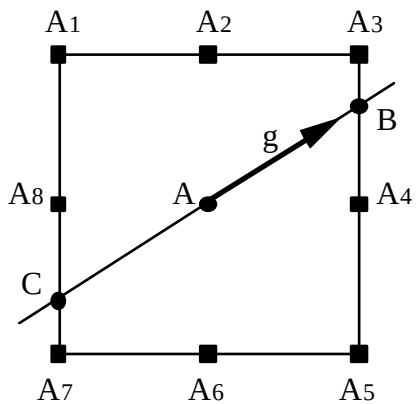
$$\nabla_x = \frac{1}{K}\begin{bmatrix} -0.2707 & -0.6065 & 0.0 & 0.6065 & 0.2707 \end{bmatrix} * \begin{bmatrix} 0.1353 \\ 0.6065 \\ 1.0 \\ 0.6065 \\ 0.1353 \end{bmatrix} * f(i,j)$$

$$\nabla_y = \frac{1}{K}\begin{bmatrix} 0.1353 & 0.6065 & 1.0 & 0.6065 & 0.1353 \end{bmatrix} * \begin{bmatrix} 0.2707 \\ 0.6065 \\ 0.0 \\ -0.6065 \\ -0.2707 \end{bmatrix} * f(i,j)$$

Máximos del Gradiente

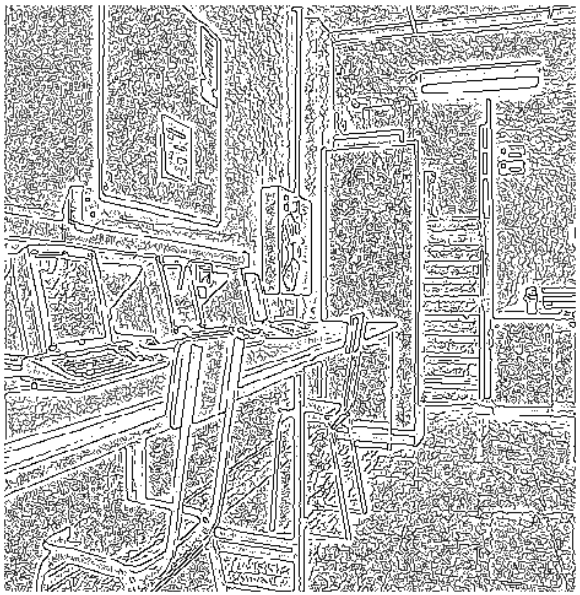
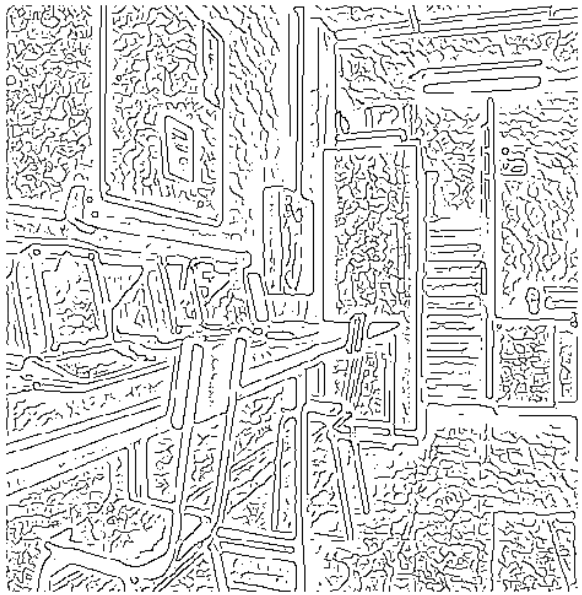
Máximos locales del módulo en la dirección del Gradiente

- Para cada pixel con módulo mayor que un umbral, se toma una ventana 3x3

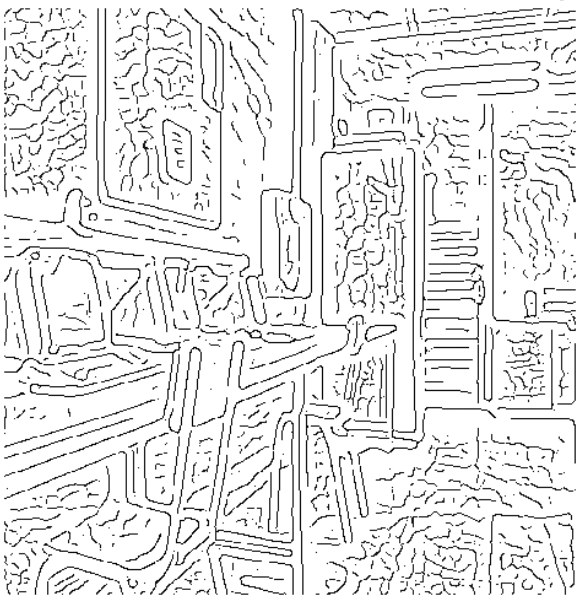
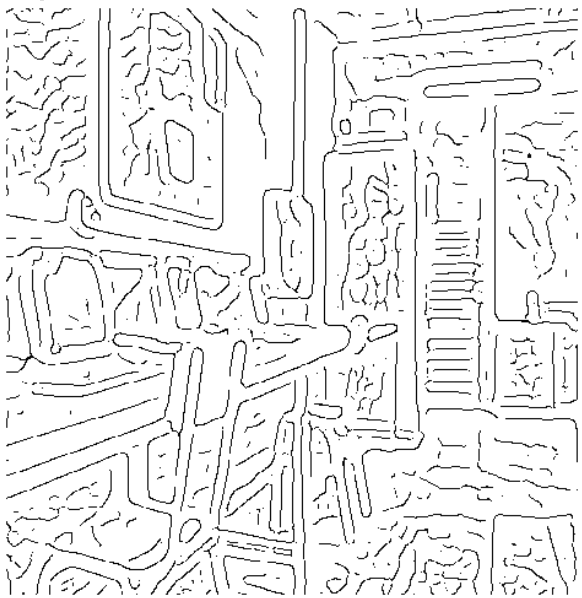


  - Se calcula el módulo del gradiente en B y C, interpolando linealmente entre los valores de A3-A4 y A7-A8
  - El pixel A es punto de contorno si su módulo es mayor que el de B y mayor o igual que el de C

Máximos del Gradiente

$\sigma = 1, 2, 3, 4$

Detección con la 2ª derivada

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

	$\nabla^2 x$			$\nabla^2 y$
				1
2ª *diferencia*	1	−2	1	−2
				1

- Operadores para el Laplaciano

0	1	0
1	−4	1
0	1	0

1	4	1
4	−20	4
1	4	1

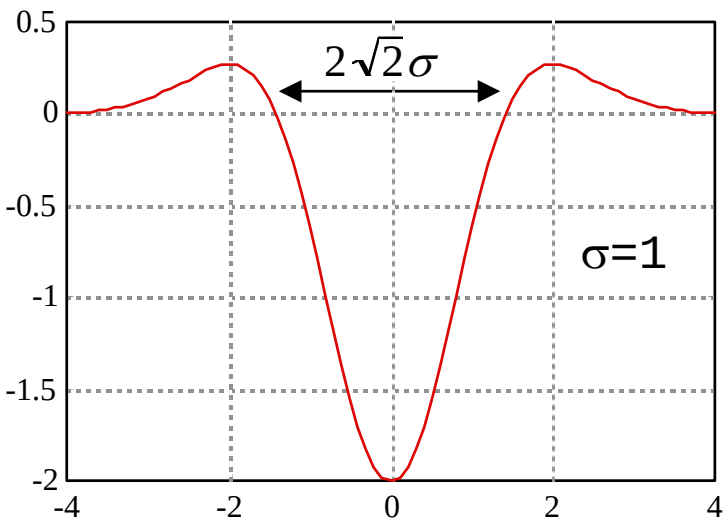
  - Cruces por cero: contornos conexos, cerrados, y delgados
  - La 2ª derivada amplifica mucho el ruido: necesario filtrar

Operador de Marr-Hildreth

- Filtro Gaussiano + Operador Laplaciano

$$\nabla^2 * G_\sigma * f(i,j) = \left(\nabla^2 G_\sigma\right) * f(i,j)$$

$$\nabla^2 G_\sigma(r) = \frac{r^2 - 2\sigma^2}{\sigma^4} \exp\left(\frac{-r^2}{2\sigma^2}\right)$$



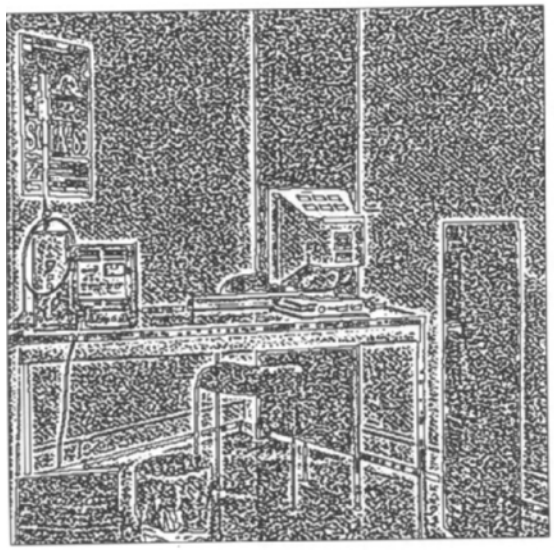
Sombrero mexicano (invertido)

Contornos: Cruces por cero

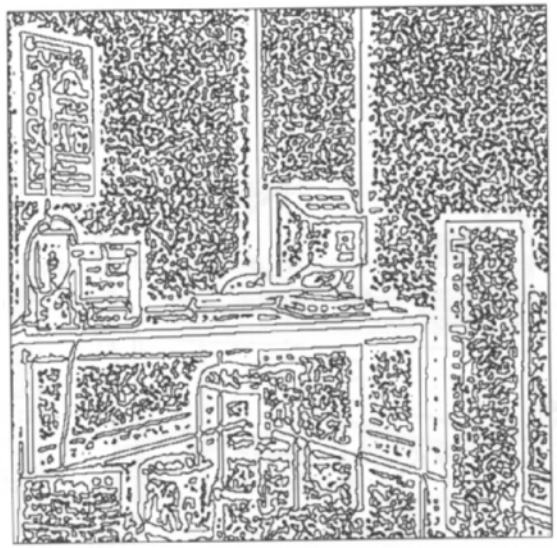
- Se consideran contornos los cruces por cero de magnitud superior a un umbral

```
my:= 0
mx:= 0
si g(i,j) >= 0 entonces
  si g(i+1,j) < 0 entonces
    my:= g(i,j) - g(i+1,j)
  fsi
  si g(i,j+1) < 0 entonces
    mx:= g(i,j) - g(i,j+1)
  fsi
sino
  si g(i+1,j) >= 0 entonces
    my:= g(i+1,j) - g(i,j)
  fsi
  si g(i,j+1) >= 0 entonces
    mx:= g(i,j+1) - g(i,j)
  fsi
fsi
magnitud(i,j):= max(my,mx)
```

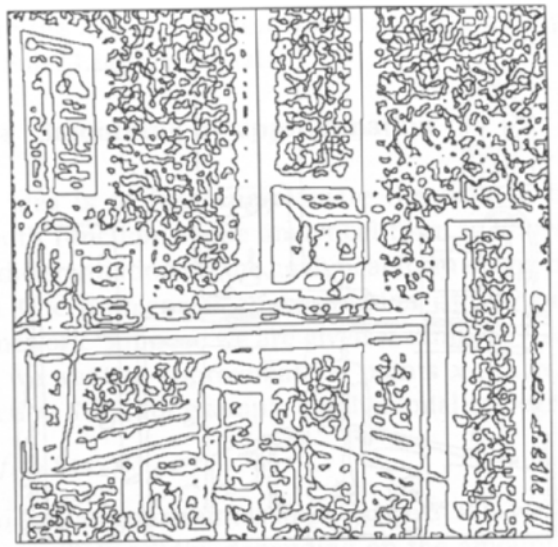
Contornos: Cruces por cero



Zero crossings of $\nabla^2 G$ with σ=1.



Zero crossings of $\nabla^2 G$ with σ=2.



Zero crossings of $\nabla^2 G$ with σ=3.

Figure 4.10 The zero-crossings of the Marr-Hildreth operator applied to the

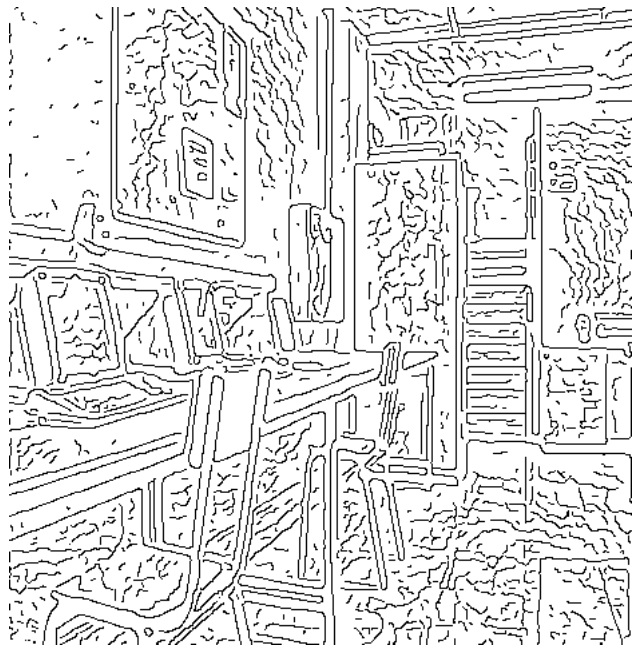
Segmentación de Contornos

- Objetivos:
  - Extracción de rectas o curvas
  - Eliminación de contornos espúreos
- Técnicas:
  - Análisis local:
    - » Seguimiento de contornos
      - Umbral con histéresis
    - » Ajuste de rectas o curvas
      - División recursiva
      - Mínimos cuadrados
  - Transformada de Hough
  - Método de Burns: segmentación de rectas usando la orientación del gradiente

Seguimiento de Contornos

- Objetivo: cadenas de puntos de contorno de una cierta “intensidad” (módulo del gradiente o magnitud del cruce por cero)
- Elegir un valor de umbral es difícil

Umbral = 8
Contornos rotos



Umbral = 4
Demasiado ruido

Umbral con histéresis

- Cada cadena de contorno debe tener:
  - Todos los pixels >= Uinf
  - Al menos un pixel >= Usup
- Además filtrado por longitud del contorno

Uinf=4, Usup=8

Uinf=4, Usup=8, Long > 30

Seguimiento de Contornos

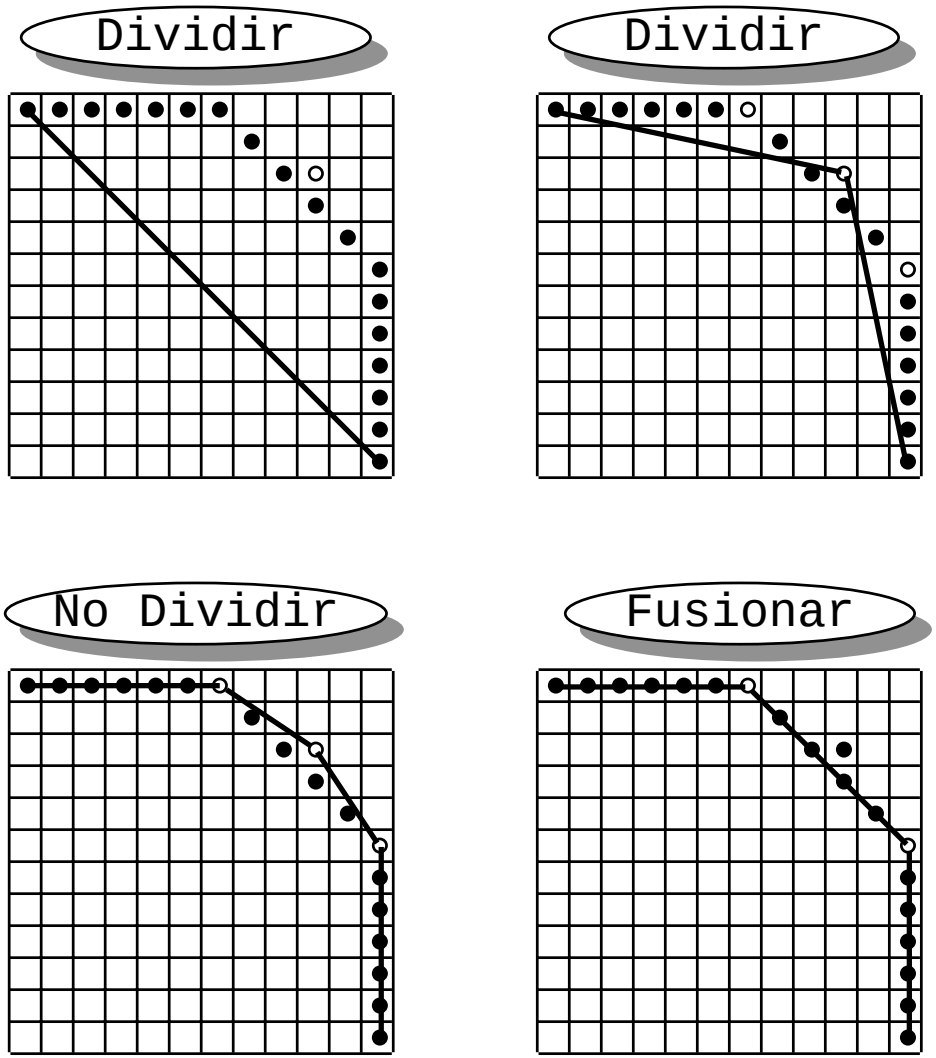
```
procedimiento encadenar_contornos
  para i:= 1 hasta nfilas
    para j:= 1 hasta ncolumnas
      inicial:=(i,j)
      si modulo(inicial) >= Umbral_sup entonces
        n:= 0
        seguir(inicial,cadena,n)
        reverse(cadena, n)
        seguir(siguiente(inicial),cadena,n))
        si n >= longitud_minima entonces
          guardar(cadena,n)
        fsi
      fsi
    fpara
  fpara
fin encadenar_contornos

procedimiento seguir(actual,var cadena,var n)
  mientras actual <> nulo
    n:=n+1
    cadena(n):= actual
    modulo(actual):= 0
    actual:= siguiente(actual)
  fmientras
fin seguir

funcion siguiente(actual) devuelve pixel
  para sigte en vecinos(actual)
    si modulo(sigte) >= Umbral_inf entonces
      devolver sigte
    fsi
  fpara
  devolver nulo
fin siguiente
```

División recursiva en rectas

1. División recursiva de una cadena:
   - Recta que une los extremos de la cadena
   - Punto que más se separa de la recta
   - Si separación > umbral (ej: 2 pixels), partir la cadena por dicho punto
   - Repetir para las dos subcadenas obtenidas
2. Fusión de segmentos:
   - Dos segmentos adyacentes se fusionan si la desviación del pixel más alejado de la recta es menor que el umbral
3. Eliminar segmentos muy cortos o con gradiente pequeño
4. Recta definida por los extremos de la cadena, o ajustada por mínimos cuadrados

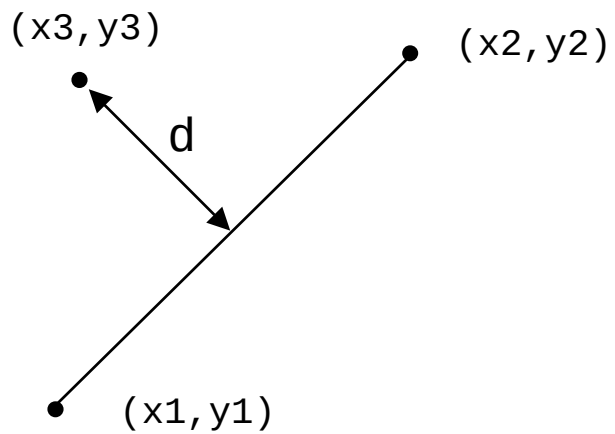
División recursiva en rectas

- Ejemplo



División recursiva en rectas

- Distancia de un punto a la recta que pasa por otros dos puntos:



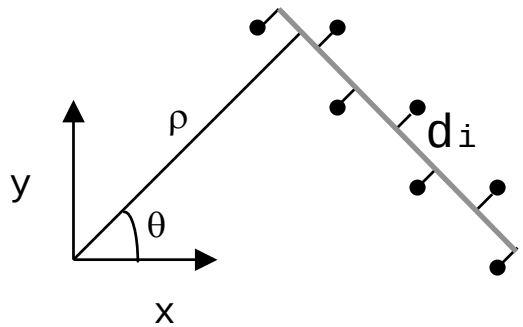
$$d = \frac{\left| x_3(y_1 - y_2) - y_3(x_1 - x_2) + y_2 x_1 - y_1 x_2 \right|}{\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}}$$

División y ajuste de rectas

- Cadenas de contornos y rectas obtenidas

Ajuste de rectas

- Mínimos cuadrados (regresión total)
  - Minimizar la suma del cuadrado de las distancias pixel-recta



Ecuación de la recta: $x\cos\theta + y\sin\theta - \rho = 0$

Distancia del pixel i: $d_i = x_i\cos\theta + y_i\sin\theta - \rho$

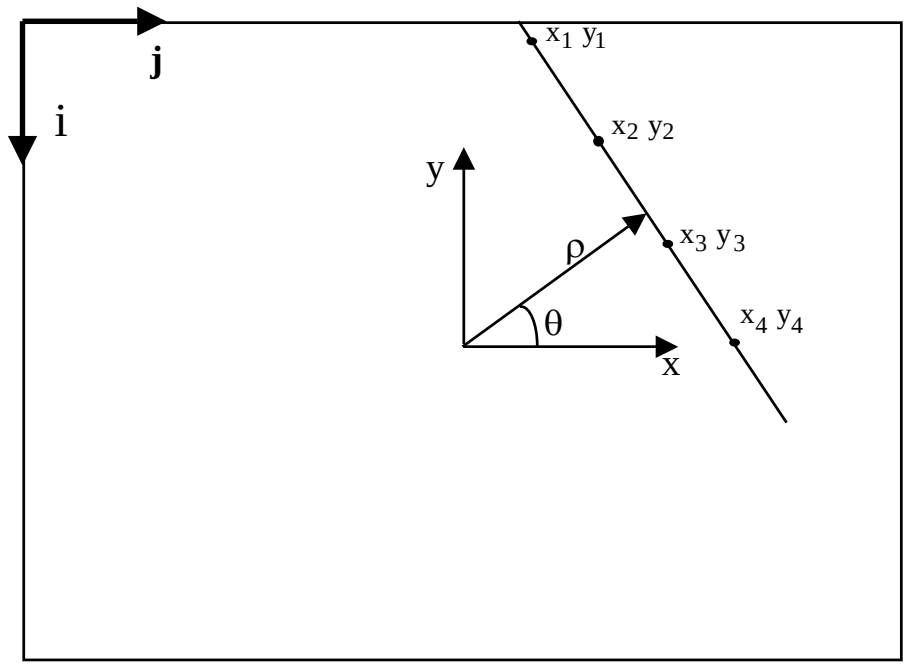
Minimizar: $D^2 = \sum_i \left(x_i\cos\theta + y_i\sin\theta - \rho\right)^2$

- Solución: eje de “minima inercia”
  - Igual que el calculo de la posición y orientación de un blob.
- Extremos: proyección del pixel inicial y final sobre la recta obtenida

4.2 Transformada de Hough

- Detección global de rectas o curvas
- Se representan de forma paramétrica
- Parámetros para las rectas: ρ y θ

$$\rho = x \cos\theta + y \operatorname{sen}\theta$$

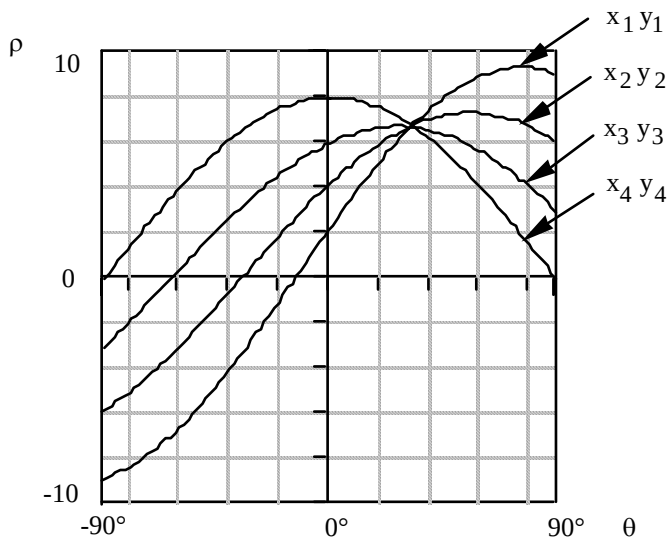


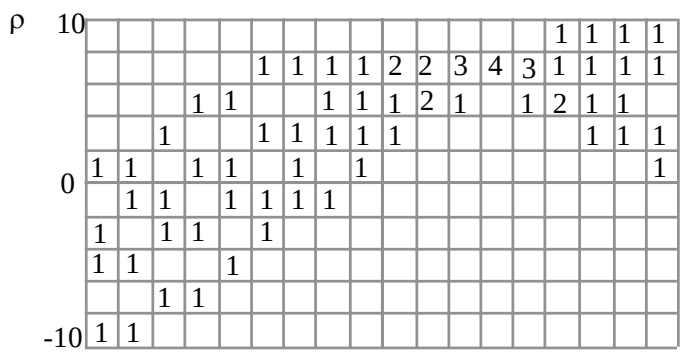
$$x = j - ncolumnas/2$$

$$y = nfilas/2 - i$$

Transformada de Hough

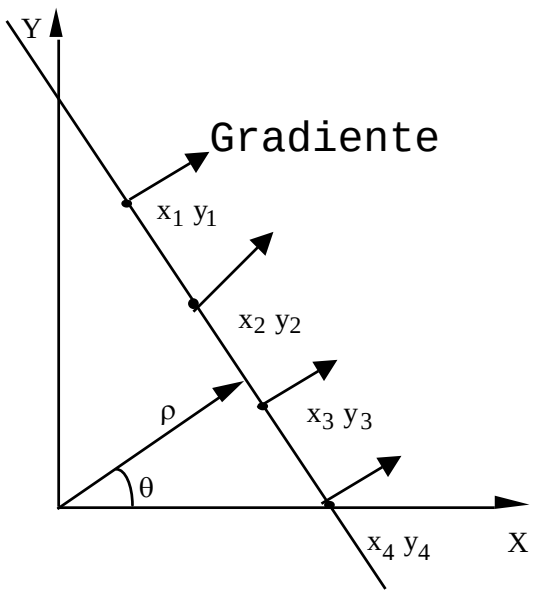
- Un pixel de coordenadas (x_i, y_i) puede pertenecer a cualquier recta de ecuación: $\rho = x_i \cos\theta + y_i \operatorname{sen}\theta$
- Se divide el espacio (ρ,θ) en celdas
- Cada pixel vota a todas las rectas a las que puede pertenecer
- Buscar las rectas más votadas

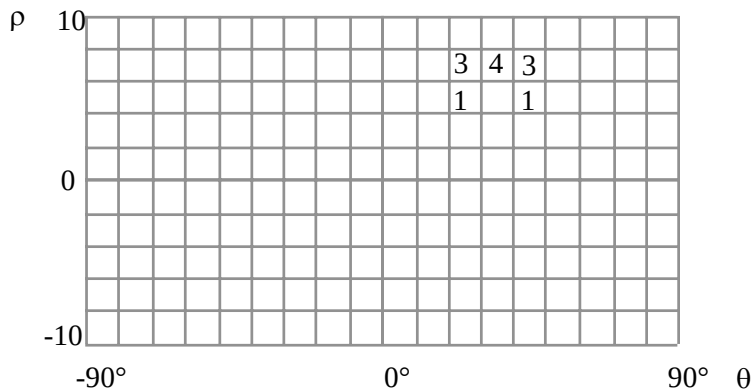




Tr. Hough: Uso de la orientación

- Tiempo de cálculo excesivo
- Mejora: cada pixel solo vota en orientaciones similares a la orientación de su gradiente





Tr. Hough: Uso de la orientación

Sin Orientación

```
para i:= 1 hasta nfilas
  para j:= 1 hasta ncolumnas
    si modulo(i,j)>=Umbral
      x:= j - ncolumnas/2
      y:= nfilas/2 - i
      para θ:=θmin hasta θmax
        ρ:= x*cos(θ) + y*sin(θ)
        votar_recta(i,j,ρ,θ)
      fpara
    fsi
  fpara
fpara
```

Con Orientación

```
para i:= 1 hasta nfilas
  para j:= 1 hasta ncolumnas
    si modulo(i,j)>=Umbral
      x:= j - ncolumnas/2
      y:= nfilas/2 - i
      θ:= orientacion(i,j)
      ρ:= x*cos(θ) + y*sin(θ)
      votar_recta(i,j,ρ,θ)
    fsi
  fpara
fpara
```

Tr. de Hough: Círculos

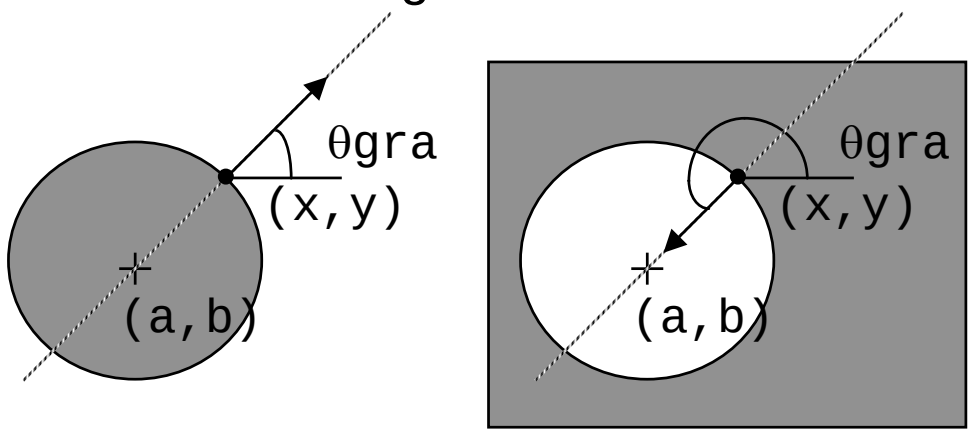
- También sirve para curvas, pero el número de parámetros es mayor
- Elemplo: círculos
  - Radio y Centro (r, a, b)
  - Forma paramétrica (polar)

$$r^2 = (x-a)^2 + (y-b)^2$$

$$x = a + r\cos\theta$$

$$y = b + r\sin\theta$$

  - Usando la orientación del gradiente:



$$a = x - r\cos\theta \qquad a = x + r\cos\theta$$

$$b = y - r\sin\theta \qquad b = y + r\sin\theta$$

Tr. de Hough: Círculos

- Tabla de acumulación 3D: (r,a,b)

Con Orientación

```
para i:= 1 hasta nfilas
  para j:= 1 hasta ncolumnas
    si modulo(i,j)>=Umbral entonces
      x:= j - ncolumnas/2
      y:= nfilas/2 - i
      θ:= orientacion(i,j)
      para r:= rmin hasta rmax
        a:= x - r*cos(θ)
        b:= y - r*sin(θ)
        votar_circulo(i,j,r,a,b)
      fpara
    fsi
  fpara
fpara
```

Aplicación: punto de fuga de un pasillo

- Cada punto de contorno vota a un punto de la línea del horizonte