



Div. Ingeniería de Sistemas y Automática

Universidad Miguel Hernández

VISIÓN POR COMPUTADOR

Tema 5. Detección de bordes



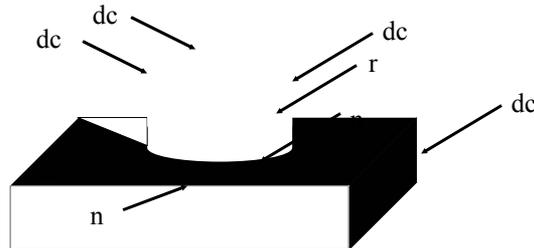
GRUPO DE TECNOLOGÍA
INDUSTRIAL

Tabla de Contenidos

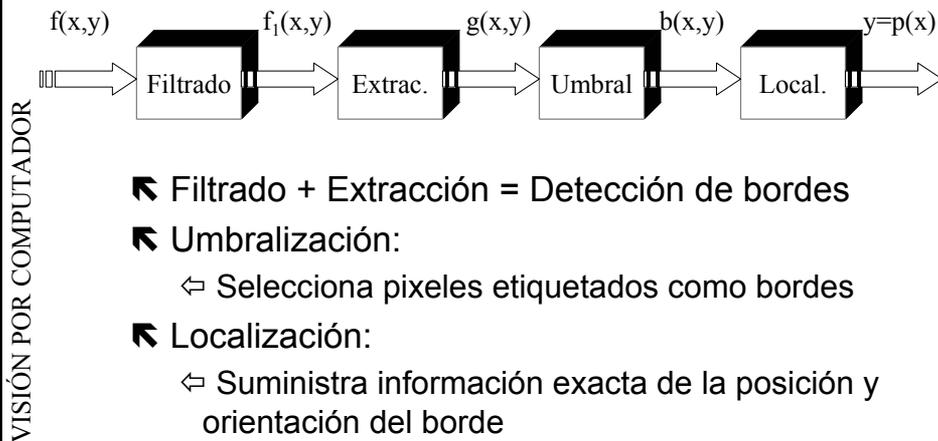
VISIÓN POR COMPUTADOR

- 📄 Definición de Borde
- ↖ Extracción de Bordes
- ↖ Operadores Derivada
- ↖ Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- ↖ Umbralización

- ↖ Cualquier discontinuidad que sufre alguna función de intensidad sobre los puntos de la misma
- ↖ Tipos de bordes:
 - ⇨ Cambio brusco en la distancia cámara-objeto (dc)
 - ⇨ Cambio en la normal del objeto (n)
 - ⇨ Cambio en la reflectancia del objeto (r)
 - ⇨ Cambio en la proyección de la luz incidente (s)



- ↖ Definición de Borde
- 📄 Extracción de Bordes
- ↖ Operadores Derivada
- ↖ Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- ↖ Umbralización

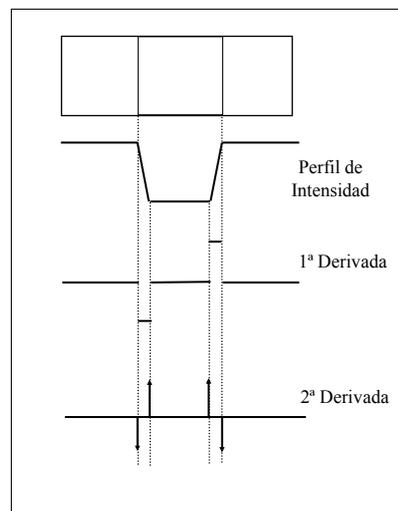


- VISION POR COMPUTADOR
- ↖ En el dominio espacial
 - ↖ Operadores derivada de los niveles de intensidad
 - ↖ Comparaciones de funciones locales de intensidad
 - ↖ Comparación con máscaras orientadas, previamente definidas (*template matching*)
 - ↖ Características exigibles
 - ↖ Precisión
 - ↖ Robustez
 - ↖ Calidad del etiquetado

- ↖ Definición de Borde
- ↖ Extracción de Bordes
- 📄 Operadores Derivada
- ↖ Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- ↖ Umbralización

Operadores Derivada

- ↖ Efecto
 - ↖ La primera derivada produce un resalte de las zonas en que la intensidad no es homogénea.
 - ↖ La segunda derivada origina un cambio de signo en la posición de borde.
 - ↖ "Zero-crossing" (paso por cero)



↩ Vector Gradiente

$$\nabla f(x,y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

$$\text{Mag}[\nabla f(x,y)] = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

$$\theta = \text{arctag} \frac{\frac{\partial f(x,y)}{\partial x}}{\frac{\partial f(x,y)}{\partial y}}$$

↩ Vector unitario en la dirección del gradiente

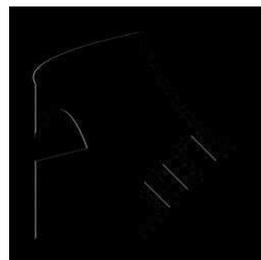
$$u_g = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \frac{1}{\sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}} \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

↩ Discretización del Vector Gradiente (Eje X)

$$\frac{\partial f(x,y)}{\partial x} \approx \nabla_x f(x,y) = f(x,y) - f(x-1,y)$$

-1	1
----	---

↔ Desplazamiento de los bordes en una cantidad positiva de 0.5 píxels

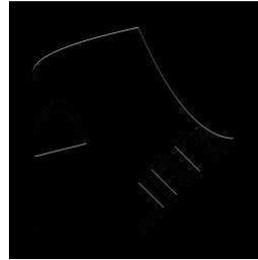


Discretización del Vector Gradiente (Eje Y)

$$\frac{\partial f(x,y)}{\partial y} = \nabla_y f(x,y) = f(x,y) - f(x,y-1)$$

-1
1

VISIÓN POR COMPUTADOR



Aproximaciones (Vector Gradiente)

$$\hat{M}ag = |\nabla_x f(x,y)| + |\nabla_y f(x,y)|$$

$$\hat{M}ag = |\nabla_r f(x,y)| + |\nabla_o f(x,y)|$$

siendo

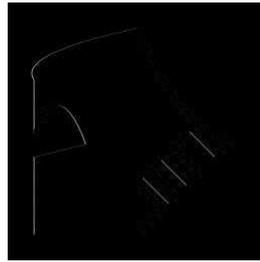
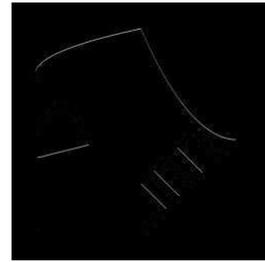
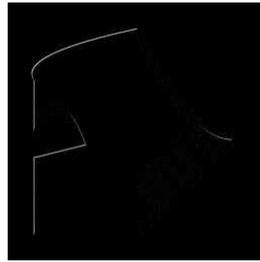
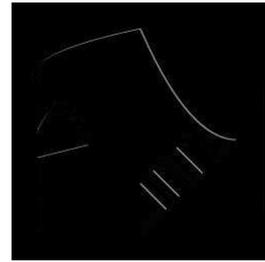
$$\nabla_r f(x,y) = f(x,y)^*$$

-1	0
0	1

$$\nabla_o f(x,y) = f(x,y)^*$$

0	-1
1	0

VISIÓN POR COMPUTADOR

 $\nabla_x f(x,y)$  $\nabla_y f(x,y)$  $\nabla_x f(x,y)$  $\nabla_y f(x,y)$

↖ Primera Derivada Direccional

↖ Derivada de la función $f(x,y)$ en la dirección α

$$D_\alpha f(x,y) = \nabla f(x,y)^T \cdot u_\alpha = \frac{\partial f(x,y)}{\partial x} \cdot \cos\alpha + \frac{\partial f(x,y)}{\partial y} \cdot \sin\alpha$$

↖ Derivada de la función $f(x,y)$ en la dirección del gradiente θ

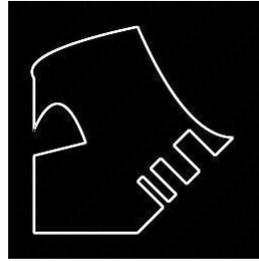
$$D_\theta f(x,y) = \nabla f(x,y)^T \cdot u_\theta = \frac{\partial f(x,y)}{\partial x} \cdot \cos\theta + \frac{\partial f(x,y)}{\partial y} \cdot \sin\theta$$

↔ Escogiendo el propio gradiente como vector

$$D_\theta f(x,y) = \nabla f(x,y)^T \cdot \nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x} \right)^2 + \left(\frac{\partial f(x,y)}{\partial y} \right)^2$$

↩ Primera Derivada Direccional (Ejemplo)

VISION POR COMPUTADOR



Primera Derivada Direccional

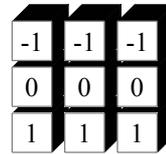
- ↩ Isotrópico (v)
- ↩ No Lineal (i)

VISION POR COMPUTADOR

↩ Gradiente de Roberts

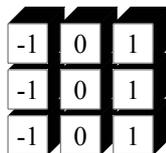


Detector horizontal



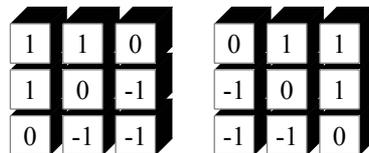
f_y

↩ Detector vertical



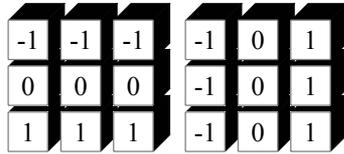
f_x

Detectores diagonales

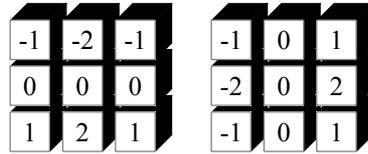


VISIÓN POR COMPUTADOR

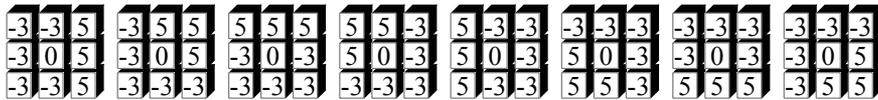
↔ Operador de Prewitt



Operador de Sobel



↔ Máscaras de Kirsch

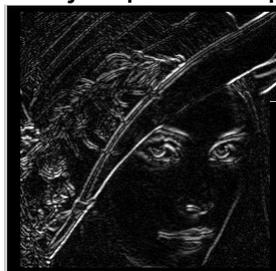


K_0 0° K_1 45° K_2 90° K_3 135° K_4 180° K_5 235° K_6 270° K_7 315°

Módulo del Gradiente = Máximo de estos valores
 Dirección = Correspondiente al máximo

VISIÓN POR COMPUTADOR

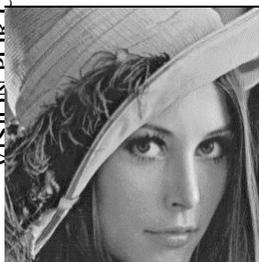
↖ Ejemplos de operadores derivada



Detector horizontal



Detector vertical



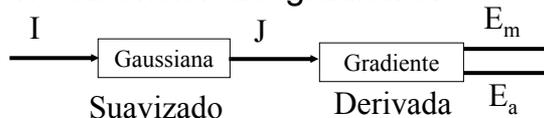
Detector de Sobel



↖ Algoritmo de Canny

- ↖ Obtención del gradiente
- ↖ Supresión no máxima al resultado del gradiente
- ↖ Histéresis de umbral a la supresión no máxima
- ↖ Cierre de contornos abiertos

1. Obtención del gradiente:



2. Supresión no máxima al resultado del gradiente

- a. Para todo punto se obtiene la dirección más cercana d_k a 0° , 45° , 90° y 135° en $E_a(i,j)$
- b. Si $E_m(i,j)$ es menor que uno de sus dos vecinos en la dirección d_k , $I_N(i,j)=0$. Si no $I_N(i,j)=E_m(i,j)$

3. Histéresis de umbral a la supresión no máxima

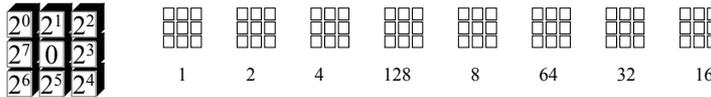
Permite eliminar máximos procedentes de ruido, etc.

- a. Entrada I_N , E_a , y dos umbrales T_1 y T_2 ($T_2 > T_1$)
- b. Para todo punto en I_N , y explorando en un orden:
 - a. Localizar el siguiente punto tal que $I_N(i,j) > T_2$
 - b. Seguir las cadenas de máximos locales a partir de $I_N(i,j)$ en ambas direcciones perpendiculares a la normal al borde siempre que $I_N > T_1$. Marcar los puntos explorados.
- c. La salida es un conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación.

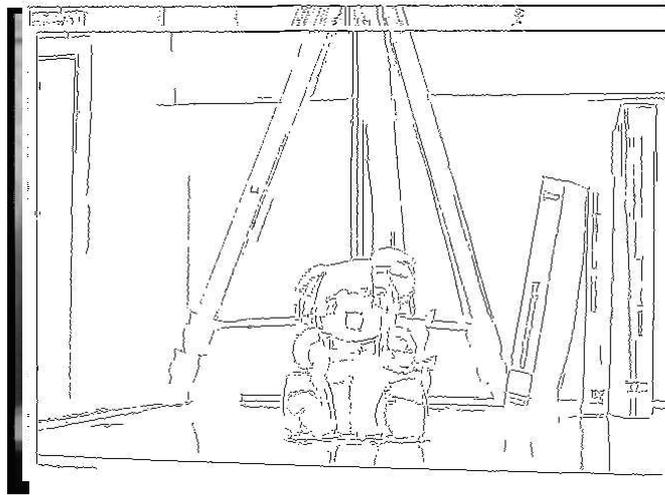
4. Cierre de contornos abiertos

(Algoritmo de Deriche y Cocquerez)

- a. La imagen de entrada es una imagen de contornos binarizada (1= borde; 0=no borde)
- b. Para cada punto de borde de un extremo abierto se le asigna un código que determina las direcciones de búsqueda para el cierre del contorno



- c. Para los pixels marcados con este código se marca como pixel de borde el de máximo gradiente en las tres direcciones posibles.
- d. Se repiten los pasos hasta que se cierren todos los contornos.



IVL SUPERVISIÓN DE NORMAS KIMOS
HISTERESIS DE UMBRAL

VISIÓN POR COMPUTADOR

↩ Operador Laplaciana

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

↩ Discretizaciones aproximadas

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

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

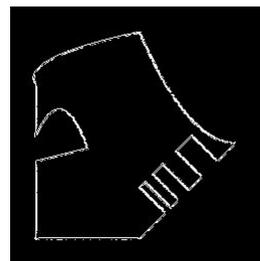
$$\frac{\partial^2 f(x,y)}{\partial y^2} \approx \Delta_y f(x,y) = f(x,y+1) - 2f(x,y) + f(x,y-1)$$

1
-2
1

VISIÓN POR COMPUTADOR

↩ Acción conjunta de las dos máscaras

0	1	0
1	-4	1
0	1	0



Operador Laplaciana

↩ Segunda Derivada Direccional

↩ En la dirección α

$$D_{\alpha}^2 f(x,y) = u_{\alpha}^T \cdot H \cdot u_{\alpha}$$

↔ siendo H la matriz Hessiana

$$H = \begin{pmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix}$$

↩ Escogiendo el vector unitario en la dirección del gradiente

$$D_{\alpha}^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} \cos^2 \theta + 2 \frac{\partial^2 f(x,y)}{\partial x \partial y} \sin \theta \cos \theta + \frac{\partial^2 f(x,y)}{\partial y^2} \sin^2 \theta$$

↔ Operador Laplaciana

0	1	0
1	-4	1
0	1	0

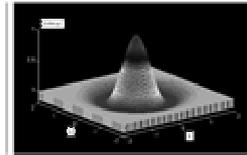
1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

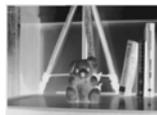
↩ Operador Laplaciana de la Gaussiana

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla^2 G(x, y) = k \left(2 - \frac{x^2+y^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- ↗ El filtro obtenido resulta bastante costoso en tiempo de computación
- ↗ Al ser un operador segunda derivada se reduce el efecto del ruido al suavizar la imagen
- ↗ Se detectan bordes en todas las direcciones
- ↗ Se puede trabajar a diferentes escalas al variar el valor de la desviación estándar de la Gaussiana. Cuanto mayor es esta desviación, habrá un menor número de pasos por cero.



Original + LG

Positivos - Blanco
Negativos - Negro

Imagen de Ceros

↩ Aproximación de Haralick y Shaphiro

- ↔ Se filtra la imagen con la Laplaciana de la Gaussiana
- ↔ Un pixel es declarado como cero:
 - Si es menor que $-t$ y uno de sus ocho vecinos es mayor que t
 - Si es mayor que t y uno de sus ocho vecinos es menor que $-t$



t=4000

Desviación = 2.0

- ↖ Definición de Borde
- ↖ Extracción de Bordes
- ↖ Operadores Derivada
- 📄 Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- ↖ Umbralización

- ↖ Funciones Gaussianas
 - ↖ Se comparan la imagen origen con imágenes filtradas con Gaussianas de distinta sigma

$$\hat{f}_1(x,y,\sigma_1) = f(x,y) * h(u,v,\sigma_1)$$

$$\hat{f}_2(x,y,\sigma_2) = f(x,y) * h(u,v,\sigma_2)$$

siendo h un filtro Gaussiano

- ↖ Realce de los bordes

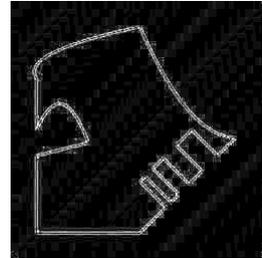
$$g(x,y) = \hat{f}_1(x,y,\sigma_1) - \hat{f}_2(x,y,\sigma_2)$$



Filtro: Gaussiano $\sigma=1$



Filtro: Gaussiano $\sigma=0.7$



Diferencia imágenes previas

- ↖ Funciones Exponenciales simétricas
- ↖ Operaciones Morfológicas
 - ↖ Imagen dilatada multinivel
 - ↖ Imagen erosionada multinivel
 - ↖ Realce de bordes: Diferencia entre ambas

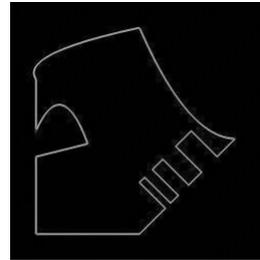
$$g(x,y) = [f(x,y) \oplus e(u,v)] - [f(x,y) \ominus e(u,v)]$$



Imagen Dilatada Multinivel



Imagen Erosionada Multinivel



Diferencia Dilatación-Erosión

- ↖ Definición de Borde
- ↖ Extracción de Bordes
- ↖ Operadores Derivada
- ↖ Comparación de Funciones Locales
- 📄 Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- ↖ Umbralización

- ↖ Conjunto de Máscaras
 - ↖ Cada una resalta la existencia de un borde en una determinada dirección
 - ↖ Tamaño y número
 - ↔ Dependiente de la precisión requerida
 - ↖ Realce de los bordes

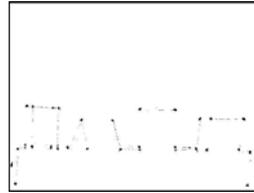
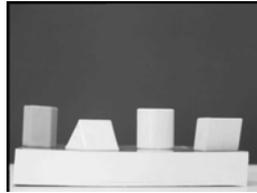
$$g(x, y, i) = \max \{ f(x, y) * h_i(u, v) \}$$

- ↖ Definición de Borde
- ↖ Extracción de Bordes
- ↖ Operadores Derivada
- ↖ Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- 📄 Extracción de Esquinas
- ↖ Umbralización

↖ Método de Kitchen y Rosenfeld

$$E = \frac{f_{xx}f_y^2 + f_{yy}f_x^2 - 2f_{xy}f_xf_y}{(f_x^2 + f_y^2)^{3/2}} \geq U_1$$

Siendo f_x, f_y las primeras derivadas y f_{xx}, f_{xy}, f_{yy} las segundas derivadas. U_1 es un umbral para determinar si el punto pertenece a una esquina.



↖ Curvatura Gaussiana (Beaudet)

$$K = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1 + f_x^2 + f_y^2)^2} \geq U_2$$

- ↖ Definición de Borde
- ↖ Extracción de Bordes
- ↖ Operadores Derivada
- ↖ Comparación de Funciones Locales
- ↖ Comparación con Máscaras Orientadas
- ↖ Extracción de Esquinas
- 📄 Umbralización

↩ Umbralización

↩ Etiquetado de los pixeles de borde

↩ Problemas

↔ Pérdidas de conectividad:

- bordes poco definidos
- filtrados muy fuertes
- umbral muy exigente

↔ Generación de bordes de anchura múltiple:

- efectos del filtrado
- umbral poco exigente

↔ Existencia de falsos bordes:

- presencia de ruido

↩ Tipos de umbralización

↔ Operadores basados en la primera derivada

 Umbralización simple

- Umbral T único, define la existencia de borde
- $g(x, y) > T$ entonces BORDE
- $g(x, y) < T$ entonces NO BORDE

 Umbralización múltiple

- Dos o más umbrales, que definen dos imágenes resultado

 Determinación de máximos locales del gradiente

↖ Ejemplos de imágenes de borde umbralizadas

VISIÓN POR COMPUTADOR

Imagen original



Bordes (Sobel)



Umbral



- ↖ Localización de la posición del borde
- ↖ Imagen umbralizada → Adelgazamiento
 - ↔ Bordes de anchura unidad
 - ↔ Información a nivel de pixel
 - ↖ Supresión de no máximos
 - ↔ Bordes de anchura unidad

VISIÓN POR COMPUTADOR