



UNIVERSIDAD DE JAÉN

# Detección de bordes en una imagen.

Departamento de Ingeniería electrónica, Telecomunicación y Automática.  
Área de Ingeniería de Sistemas y Automática

**OBJETIVOS:**

- Utilizar distintas máscaras empleadas para la detección de bordes.

**1 DETECCIÓN DE BORDES.**

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc.

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen.

**1.1. Operadores basadas en la primera derivada (Gradiente).**

La derivada de una señal continua proporciona las variaciones locales con respecto a la variable, de forma que el valor de la derivada es mayor cuanto más rápidas son estas variaciones.

En el caso de funciones bidimensionales  $f(x,y)$ , la derivada es un vector que apunta en la dirección de la máxima variación de  $f(x,y)$  y cuyo módulo es proporcional a dicha variación. Este vector se denomina gradiente y se define:

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

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

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

En el caso bidimensional discreto, las distintas aproximaciones del operador gradiente se basan en diferencias entre los niveles de grises de la imagen. La derivada parcial  $f_x(x,y)$  (gradiente de fila  $G_F(i,j)$ ) puede aproximarse por la diferencia de píxeles adyacentes de la misma fila.

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

-1	1
----	---

La discretización del vector gradiente en el eje Y ( $G_C(i,j)$ ), será:

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

-1
1

El gradiente de la fila  $G_F$  y de columna  $G_C$  en cada punto se obtienen mediante la convolución de la imagen con las máscaras  $H_F$  y  $H_C$ , esto es:

$$G_F(i, j) = F(i, j) \otimes H_F(i, j)$$

$$G_C(i, j) = F(i, j) \otimes H_C(i, j)$$

La magnitud y orientación del vector gradiente suele aproximarse por la expresión:

$$|G(i, j)| = \sqrt{G_F^2 + G_C^2} \approx |G_F(i, j)| + |G_C(i, j)|$$

Los operadores más utilizados son los de Roberts, Prewitt y Sobel y Frei-chen.

### Operador de Roberts.

Las máscaras utilizadas en este operador son:

Gradiente fila

0	0	0
0	0	1
0	-1	0

Gradiente columna

-1	0	0
0	1	0
0	0	0

Obtiene buena respuesta ante bordes diagonales. Ofrece buenas prestaciones en cuanto a localización. El gran inconveniente de este operador es su extremada sensibilidad al ruido y por tanto tiene pobres cualidades de detección.

### Operadores de Prewitt, Sobel y Frei-Chen.

Los tres operadores pueden formularse de forma conjunta con las siguientes máscaras de convolución mostradas a continuación.

Gradiente fila

$$\frac{1}{2+K} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline K & 0 & -K \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

Gradiente columna

$$\frac{1}{2+K} \begin{array}{|c|c|c|} \hline -1 & -K & -1 \\ \hline 0 & 1 & 0 \\ \hline 1 & K & 1 \\ \hline \end{array}$$

En el operador Prewitt ( $K=1$ ) se involucran a los vecinos de filas / columnas adyacentes para proporcionar mayor inmunidad al ruido.

El operador Sobel ( $K=2$ ), se supone que es más sensible a los bordes diagonales que el de Prewitt aunque en la práctica hay poca diferencia entre ellos.

Frei-Chen ( $K=\sqrt{2}$ ), el gradiente es el mismo para bordes verticales, horizontales y diagonales.

En Matlab la función que detecta bordes es **edge**.

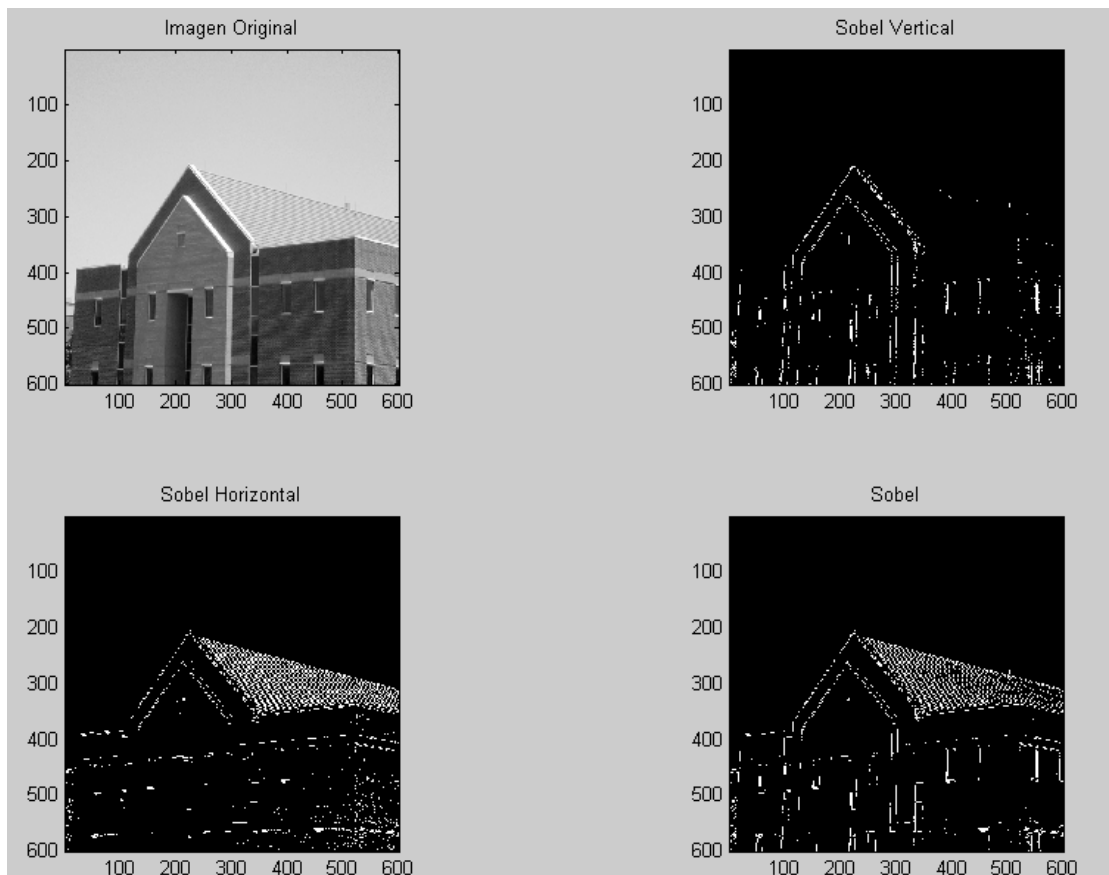
$$BW = edge(I, 'sobel', thresh, direction)$$

Esta función encuentra los bordes de una imagen de distintos niveles de intensidad. El resultado es una imagen binaria del mismo tamaño que la imagen original en la cual, “1” significa que ha detectado un borde y “0” es que no lo ha detectado.

El parámetro *thresh* indica el umbral de binarización. Si se elige el umbral de binarización, hay que ser consciente de que la función *edge* normaliza la imagen antes de procesarla, llevándola al intervalo [0,1]. También divide las máscaras empleadas en los filtros por un factor, siendo de  $\sqrt{2}$  para Roberts, 8 para Sobel y 6 para Prewitt.

Con esta función se obtiene una imagen de bordes binaria, y se puede obtener con diversos métodos: sobel, prewitt, robert, canny, combinado de suavizado y Laplaciana, operador LOG.

```
>>im1=imread('D:\visión por computador\imagenes\casa.jpg');  
>>b1=edge(im1,'sobel','vertical');  
>>b2=edge(im1,'sobel','horizontal');  
>>b3=edge(im1,'sobel');  
>>subplot(2,2,1),subimage(im1),Title('Imagen Original');  
>>subplot(2,2,2),subimage(b1),Title('Sobel Vertical');  
>>subplot(2,2,3),subimage(b2),Title('Sobel Horizontal');  
>>subplot(2,2,4),subimage(b3),Title('Sobel');
```



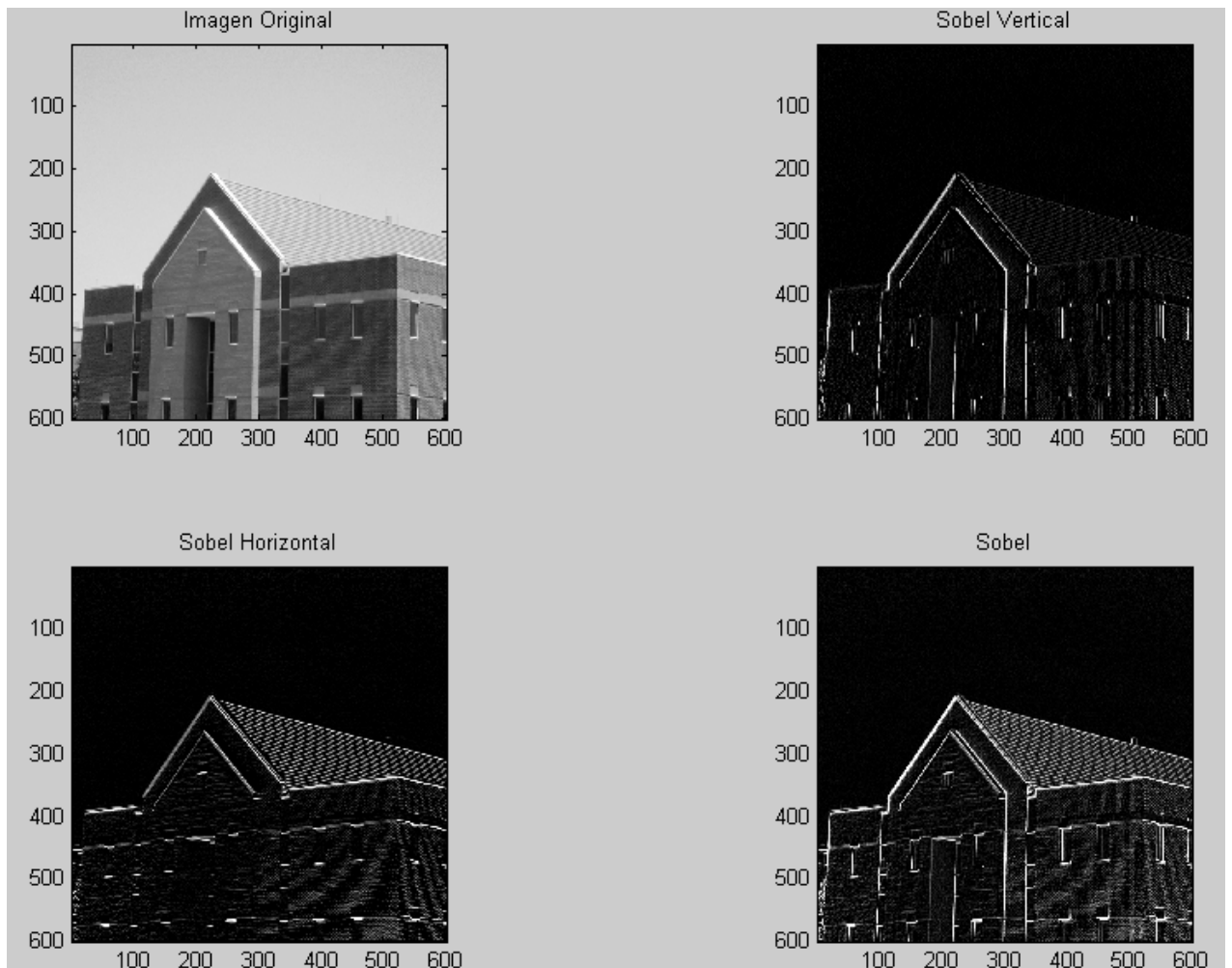
La función `edge` también puede indicar el umbral empleado en la detección de bordes:

$$[BW, umbral] = edge(I, 'sobel', ...)$$

También se podría aplicar el filtro de Sobel creando una máscara con la función `fspecial`, `imfilter`. De esta forma se obtiene una imagen de bordes en escala de grises.

Para obtener tanto los bordes horizontales como los verticales se utiliza la función **`imadd`**.

```
>>im1=imread('D:\visión por computador\imagenes\casa.jpg');
>>sf=fspecial('sobel');
>>sc=sf';
>>b1=imfilter(im1,sf);
>>b2=imfilter(im1,sc);
>>b3=imadd(b1,b2);
>>subplot(2,2,1),subimage(im1),Title('Imagen Original');
>>subplot(2,2,2),subimage(b2),Title('Sobel Vertical');
>>subplot(2,2,3),subimage(b1),Title('Sobel Horizontal');
>>subplot(2,2,4),subimage(b3),Title('Sobel');
```



Antes de aplicar algoritmos de segmentación la imagen de bordes ha de ser binaria. Para binarizar se pueden utilizar las funciones de Matlab **graythresh**, halla el umbral de binarización y la función **im2bw**, binariza la imagen.

### **Ejercicios**

1. Detectar los bordes de la imagen “bridge.gif”. Detectar primero los bordes horizontales, luego los verticales y componer en último lugar la imagen formada de bordes horizontales y verticales, utilizar para esto el filtro de sobel.
2. Hallar la imagen de bordes binaria de la imagen “lena.jpg”, aplicando todos los filtros posibles que admite la función edge. Realizar lo mismo pero en este caso no se empleará la función *edge*.

