

**CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL**



**Detección de Bordes – Laplaciano, Sobelx, Sobely, Canny.**

Practica 8

Victor Alejandro Dominguez Cruz

21310228

6°G

Visión Artificial

9 de junio de 2025

**Objetivo:** Dejar en la imagen solamente los bordes que deseamos y saber cuál es el mejor método.

**Código:**

```
import cv2 # Procesar imagenes

import numpy as np # Operaciones numericas

import matplotlib.pyplot as plt # Graficos


img = cv2.imread('descarga.png', cv2.IMREAD_GRAYSCALE)


laplaciano = cv2.Laplacian(img, cv2.CV_64F) #LaPlaciano(detecta bordes todas
direcciones)

laplaciano = cv2.convertScaleAbs(laplaciano) #Detecta bordes aplicando segunda
derivada


sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3) #Detecta bordes
verticas(calcula la derivada en X)

sobelx = cv2.convertScaleAbs(sobelx) #Escalamos valores absolutos de 8 bits


sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3) #Detecta bordes
horizontales(calcula derivada en Y para verticales)

sobely = cv2.convertScaleAbs(sobely) #Escalamos valores absolutos de 8 bits


canny = cv2.Canny(img, 100, 200)# Detecta bordes finos y precisos (umbrales de
100 (mínimo) y 200 (máximo))


plt.figure(figsize=(10, 6)) # Preparamos la figura para mostrar varias imágenes


plt.subplot(2, 3, 1)
```

```
plt.imshow(img, cmap='gray')  
plt.title('Imagen Original')  
plt.axis('off')
```

```
# Laplaciano
```

```
plt.subplot(2, 3, 2)  
plt.imshow(laplaciano, cmap='gray')  
plt.title('Laplaciano')  
plt.axis('off')
```

```
# Sobel X
```

```
plt.subplot(2, 3, 3)  
plt.imshow(sobelx, cmap='gray')  
plt.title('Sobel X')  
plt.axis('off')
```

```
# Sobel Y
```

```
plt.subplot(2, 3, 4)  
plt.imshow(sobely, cmap='gray')  
plt.title('Sobel Y')  
plt.axis('off')
```

```
# Canny
```

```
plt.subplot(2, 3, 5)  
plt.imshow(canny, cmap='gray')  
plt.title('Canny')
```

```
plt.axis('off')
```

```
plt.tight_layout() # Ajustamos el diseño para que no se encimen las imágenes
```

```
plt.show()
```

### Demostración:

