Maestro: Mauricio Alejandro Cabrera Arellano

Alumno: Nene Miranda José Said 22110319

Materia: Visión Artificial

Tarea: Practica 8

Fecha: 01-06-2025

Practica 8:

Objetivo:

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

Codigo:

```
# Importamos las librerías necesarias
import cv2  # OpenCV para procesamiento de imágenes
import numpy as np  # Para operaciones numéricas
import matplotlib.pyplot as plt  # Para mostrar imágenes con gráficos

# Cargar imagen en escala de grises
img = cv2.imread('Samus.png', cv2.IMREAD_GRAYSCALE)
# Leemos la imagen 'Samus.png' en escala de grises para facilitar la detección de bordes

# --------- Método 1: Laplaciano (detecta bordes en todas direcciones) ---------
laplaciano = cv2.Laplacian(img, cv2.CV_64F)
# Detecta bordes aplicando la segunda derivada a la imagen
laplaciano = cv2.convertScaleAbs(laplaciano)
# Convertimos los valores negativos y flotantes a escala de 8 bits para visualizar

# --------- Método 2: Sobel X (detecta bordes verticales) ---------
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
# Calcula la derivada en dirección X (cambios horizontales)
sobelx = cv2.convertScaleAbs(sobelx)
# Escalamos a valores absolutos de 8 bits

# --------- Método 3: Sobel Y (detecta bordes horizontales) ---------
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)
# Calcula la derivada en dirección Y (cambios verticales)
```

```
sobely = cv2.convertScaleAbs(sobely)

# Escalamos a valores absolutos de 8 bits


# --------- Método 4: Canny (detecta bordes finos y precisos) ---------

canny = cv2.Canny(img, 100, 200)

# Aplica el algoritmo de Canny con umbrales de 100 (mínimo) y 200 (máximo)


# --------- Mostrar resultados ---------

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


# Imagen original

plt.subplot(2, 3, 1)

plt.imshow(img, cmap='gray')

plt.title('Imagen Original')

plt.axis('off')


# Filtro Laplaciano

plt.subplot(2, 3, 2)

plt.imshow(laplaciano, cmap='gray')

plt.title('Laplaciano')

plt.axis('off')


# Filtro Sobel X

plt.subplot(2, 3, 3)

plt.imshow(sobelx, cmap='gray')

plt.title('Sobel X')

plt.axis('off')


# Filtro Sobel Y
```

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

    # Filtro Canny
    plt.subplot(2, 3, 5)
    plt.imshow(canny, cmap='gray')
    plt.title('Canny')
    plt.axis('off')

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

Demostración: