

Reporte

PRACTICA 8

ALEXIS DANIELCORTES TAVERA

Introducción

La detección de bordes es una de las operaciones más importantes en el procesamiento de imágenes y visión artificial. A través de ella, se pueden identificar contornos, formas y estructuras relevantes en una imagen, lo que facilita tareas como segmentación, reconocimiento de objetos y análisis de escenas.

En esta práctica se exploraron diferentes métodos clásicos de detección de bordes: **Laplaciano**, **SobelX**, **SobelY** y **Canny**. Cada uno de ellos analiza los cambios de intensidad en una imagen para resaltar sus transiciones más marcadas, que normalmente corresponden a bordes.

Desarrollo Teórico y Practico

En estas prácticas se usaron diversas herramientas para su realización que serán explicadas a continuación:

librerías:

- cv2: Biblioteca OpenCV, usada para el procesamiento de imágenes, captura de video en tiempo real, creación de máscaras y aplicación de filtros morfológicos.
- numpy: Biblioteca para manejo de arreglos y cálculos numéricos

Filtros utilizados:

- **Laplaciano:** Detecta bordes considerando cambios en todas direcciones. Es muy sensible al ruido, pero puede mostrar gran cantidad de detalle si se usa correctamente.
- **Sobel X y Sobel Y:** Detectan bordes verticales y horizontales respectivamente. Se utilizan para analizar los gradientes en direcciones específicas. Su resultado puede combinarse para obtener bordes completos, pero aquí se observan por separado.
- **Canny:** Considerado uno de los métodos más robustos. Integra varias etapas (suavizado, cálculo de gradiente, supresión no máxima y umbral por

Maestro:

Mauricio Alejandro Cabrera Arellano

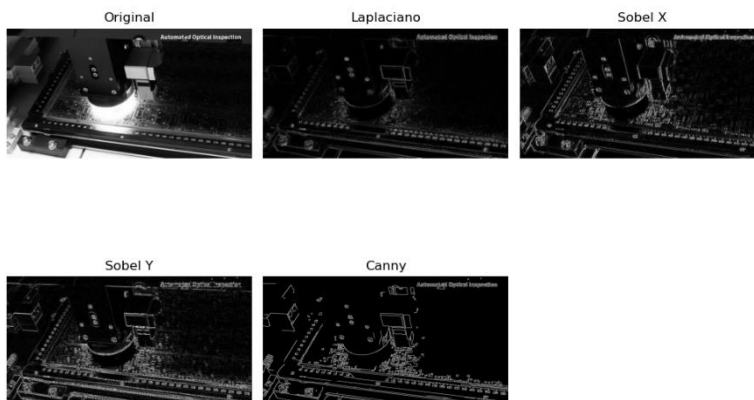
histéresis) para detectar los bordes más fuertes y consistentes en la imagen.

Practica 8:

El código carga una imagen, la convierte a escala de grises y le aplica cuatro métodos distintos de detección de bordes: **Laplaciano**, **Sobel X**, **Sobel Y** y **Canny**. Luego muestra todos los resultados en una sola figura para poder compararlos fácilmente.

- **Laplaciano** detecta bordes en todas direcciones.
- **Sobel X y Y** detectan bordes verticales y horizontales, respectivamente.
- **Canny** es el más preciso, combinando varios pasos para obtener bordes limpios.

Gracias a matplotlib, los resultados se visualizan juntos, lo que permite analizar cuál método ofrece una mejor detección según la imagen.



P3

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Cargar imagen y convertir a escala de grises
img = cv2.imread('imagen.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Laplaciano
```

Maestro:

Mauricio Alejandro Cabrera Arellano

```
laplacian = cv2.Laplacian(gray, cv2.CV_64F)
laplacian = np.uint8(np.absolute(laplacian))
# Sobel X y Y
sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
sobelx = np.uint8(np.absolute(sobelx))
sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
sobely = np.uint8(np.absolute(sobely))
# Canny
canny = cv2.Canny(gray, 100, 200)
# Visualización
titles = ['Original', 'Laplaciano', 'Sobel X', 'Sobel Y', 'Canny']
images = [gray, laplacian, sobelx, sobely, canny]
plt.figure(figsize=(10, 8))
for i in range(5):
    plt.subplot(2, 3, i+1)
    plt.imshow(images[i], cmap='gray')
    plt.title(titles[i])
    plt.axis('off')
plt.tight_layout()
plt.show()
```

Conclusión

Esta práctica estuvo bastante interesante porque permitió ver claramente cómo diferentes métodos detectan los bordes de una imagen. Aunque todos cumplen la misma función, cada uno lo hace a su manera: algunos muestran más detalle, otros son más limpios o bruscos.

El método de **Canny** destacó por dar un resultado más definido y ordenado, mientras que **Laplaciano** mostró muchos detalles, pero también más ruido. **Sobel X y Y** ayudan a entender cómo se ven los bordes dependiendo de su orientación.

Al final, experimentar con todos al mismo tiempo fue útil para entender cuál se adapta mejor dependiendo de lo que se quiera lograr con la imagen. Sin duda, detectar bordes es una de las partes más importantes en visión artificial y esta práctica lo demuestra muy bien.

Referencias:

Ibm. (2025, 6 enero). Computer Vision. *Vision Artificial*. <https://www.ibm.com/mx-es/topics/computer-vision>

Maestro:

Mauricio Alejandro Cabrera Arellano