

Centro de Enseñanza Técnica Industrial

Visión Artificial.

Ingeniería Mecatrónica.



Practica# 8: Detección de Bordes

Platel: CETI Colomos.

Nombre: Ruiz Macías Luis Enrique - 21310196

Grado/Grupo: 6°G

Objetivo: Dejar en la imagen los bordes que deseamos en y determinar cuál es el mejor método de detección de bordes para nuestro caso.

En el presente código se presenta un código que presenta la funcionalidad de detectar bordes con diferentes filtros y con diferentes características cada uno, siendo estos el sobel Y sobel X, el canny y el laplaciano.

El detectar bordes nos ayuda a discriminar información y mantener la más relevante según sea lo necesario, de modo que prescindimos de lo que no nos sirve y mantenemos la información de interés.

A continuación, se presenta el Código:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Cargar imagen en escala de grises
img = cv2.imread('IVECO.jpg', cv2.IMREAD_GRAYSCALE)
img = cv2.resize(img, (350, 200)) # Cambia a las dimensiones que desees (ancho,
alto)

# Verificar que se cargó correctamente
if img is None:
    print("Error: No se pudo cargar la imagen.")
    exit()

# Aplicar detección de bordes

# 1. Laplaciano (detecta cambios rápidos en todas direcciones)
laplacian = cv2.Laplacian(img, cv2.CV_64F)
laplacian = cv2.convertScaleAbs(laplacian)
```

2. Sobel X (detecta bordes verticales)

```
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
```

```
sobelx = cv2.convertScaleAbs(sobelx)
```

3. Sobel Y (detecta bordes horizontales)

```
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)
```

```
sobely = cv2.convertScaleAbs(sobely)
```

4. Canny (algoritmo multietapa, muy preciso)

```
canny = cv2.Canny(img, 150, 250) # puedes ajustar los umbrales
```

Mostrar resultados

```
titulos = ['Imagen Original', 'Laplaciano', 'Sobel X', 'Sobel Y', 'Canny']
```

```
imagenes = [img, laplacian, sobelx, sobely, canny]
```

```
plt.figure(figsize=(14, 10))
```

```
for i in range(5):
```

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

```
    plt.imshow(imagenes[i], cmap='gray')
```

```
    plt.title(titulos[i])
```

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

```
plt.tight_layout()
```

```
plt.show()
```

