

Trabajo 2: Contornos

Visión por computador

Ingeniería Informática Unizar

Curso 2020/2021

| | |
|-----------------|--------|
| Miguel Gómez | 741302 |
| Alejandro Omist | 737791 |

Introducción

En este primer trabajo de la asignatura de Visión por Computador se ha desarrollado una aplicación con OpenCV en c++ que detecta los contornos de una imagen para posteriormente obtener el punto de fuga de dicha imagen.

1. Realizar el cálculo del gradiente horizontal y vertical, y del módulo y orientación del gradiente

- Operadores de Sobel o de Schurr

Para el cálculo del gradiente horizontal y vertical se han probado ambos operadores con sus correspondientes funciones ya implementadas en OpenCV pero se ha optado finalmente por utilizar Sobel ya que los resultados obtenidos han sido consistentemente mejores.

En cuanto a los cálculos del módulo y la orientación se ha empleado la función *cartToPolar* DE OpenCV. Previamente se ha realizado un preprocesamiento de la imagen para eliminar ruido mediante un filtro Gaussiano (función *GaussianBlur*).

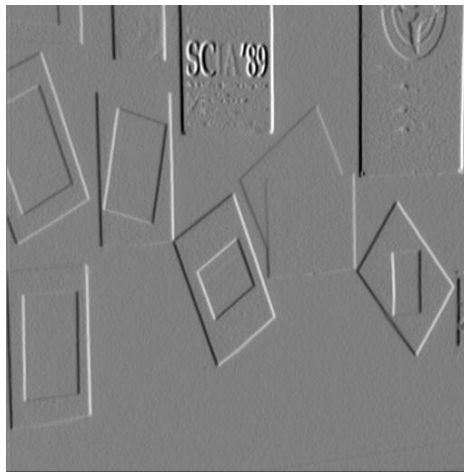


Imagen 1: Gradiente en x (Sobel)

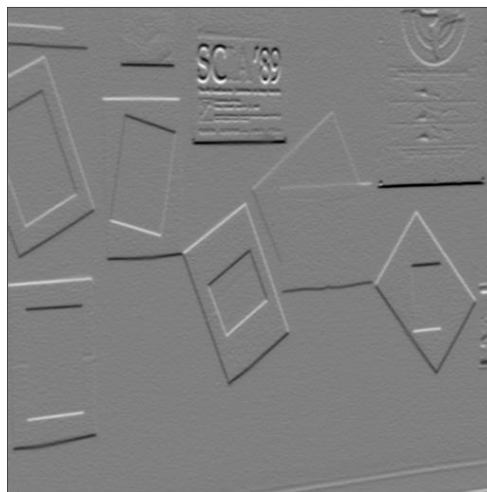


Imagen 2: Gradiente en y (Sobel)



Imagen 3: Módulo del gradiente (Sobel)

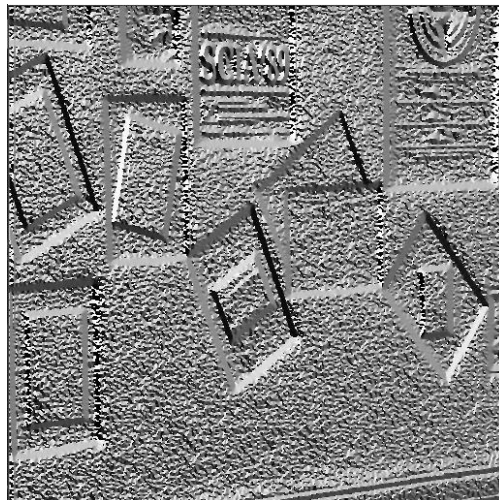


Imagen 4: Orientación del gradiente (Sobel)

- **Operadores de Canny**

Para esta parte se emplean dos matrices, una para la máscara Gaussiana y otra para su derivada. También se calcula el valor de K para posteriormente normalizar los gradientes. Las máscaras son aplicadas sobre la imagen empleando la función *sepFilter2D* de OpenCV. Después a las dos imágenes obtenidas se las divide por K obteniendo de esta forma el gradiente horizontal y el gradiente vertical

Para el cálculo del módulo y de la orientación se han empleado las siguientes fórmulas vistas en clase:

$$\text{Módulo} \quad |\nabla f| = \sqrt{(\nabla_x)^2 + (\nabla_y)^2}$$

$$\text{Orientación} \quad \theta = \text{atan2}(\nabla_y, \nabla_x)$$

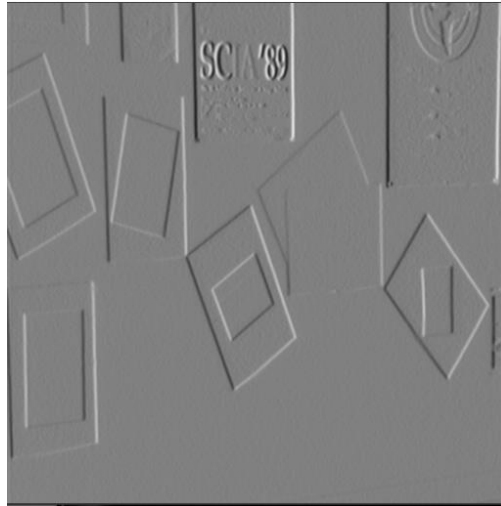


Imagen 5: Gradiente en x (Canny)

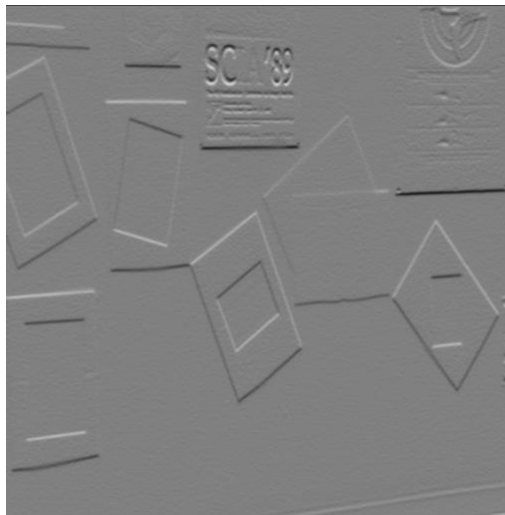


Imagen 6: Gradiente en y (Canny)



Imagen 7: Módulo del gradiente (Canny)

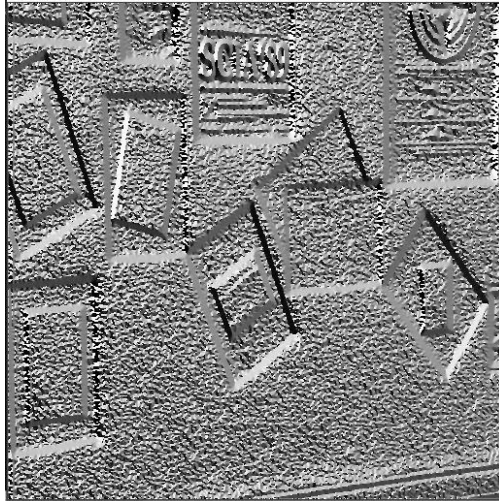


Imagen 8: Orientación del gradiente (Canny)

- **Contornos**

Una vez calculado el módulo y la orientación del gradiente se procede a identificar los contornos de la imagen. Para ello se compara cada uno de los píxeles de la matriz del módulo y se compara con el umbral correspondiente, y en caso de que se supere, se agranda el grosor del contorno dibujando círculos sobre la matriz del módulo.

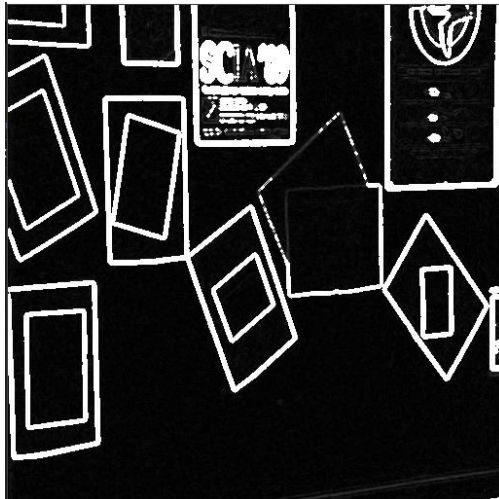


Imagen 9: Contornos (Sobel)

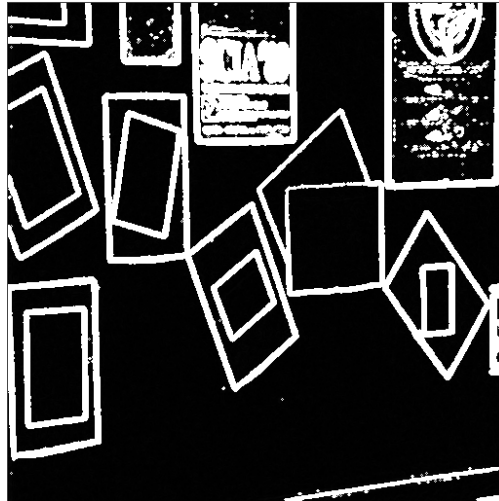


Imagen 10: Contornos (Canny)

Como se puede observar utilizando el operador de Canny se detectan más contornos ya que su umbral es menor.

2. Programar la detección del punto central de un pasillo mediante la transformada de Hough

Una vez que hemos comprobado que los contornos se han detectado correctamente, los utilizamos para localizar el punto de fuga utilizando la transformada de Hough mediante democracia directa. Es decir, cada uno de los puntos de contorno votará la existencia de varias rectas que pasan por dicho punto, estando cada recta definida por la siguiente fórmula:

$$x \cos\theta + y \sin\theta - \rho = 0$$

Siendo ρ la distancia al centro y θ la inclinación.



Imagen 11: Punto de fuga pasillo 1



Imagen 12: Punto de fuga pasillo 2