Práctica 2.
Contornos

Alberto Lardiés, 735976
Devid Dokash, 780131

24 de junio de 2023

Índice

1. Decisiones de diseño generales

El objetivo de este trabajo es desarrollar con OpenCV un programa capaz de detectar contornos y utilizarlos para obtener el punto de fuga de un pasillo:

1. Para el suavizado de las imágenes, se ha usado la función `cv.GaussianBlur(frame,(3,3),0)`, con un kernel de 3x3 debido a que era el kernel con mejores resultados.
2. Para el calculo de los gradientes, modulo y orientación se han implementado tanto el operador Sobel como el operador canny manuales.
3. Para la transformada de Hough, se comprueba que cada píxel sea apto para votar y se calcula la recta de la orientación en coordenadas polares.
4. Se ha implementado una pequeña interfaz que permite subir una imagen o visualizar los efectos en vivo, eligiendo que efectos mostrar, y elegir el operador (Sobel o Canny).

2. Gradientes

En esta primera parte, se han implementado dos operadores. Comenamos por el operador de Sobel:

1. Se suaviza la imagen mediante una matriz convolucional de tamaño 3x3 aplicando un filtrado de blur Gaussiano y así reduciendo el ruido de los píxeles de la imagen.
2. Se hallan los distintos gradientes (X e Y) de la imagen mediante el operador de Sobel que hemos implementado, mediante un kernel de 3x3 para ambos casos.

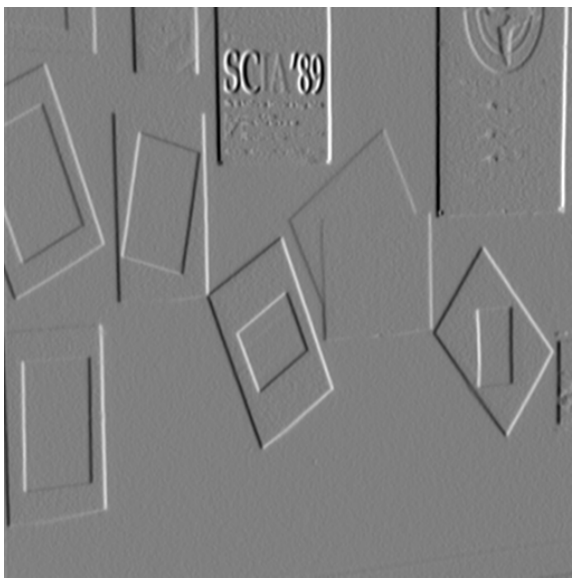

Figura 1: Gradiente en el eje X.

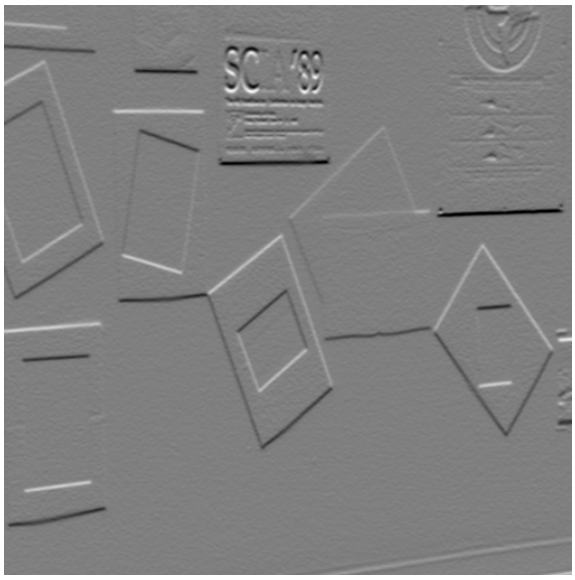

Figura 2: Gradiente en el eje Y.

3. Se halla de manera adicional, el gradiente en ambas direcciones (XY), que es la combinación de las dos.

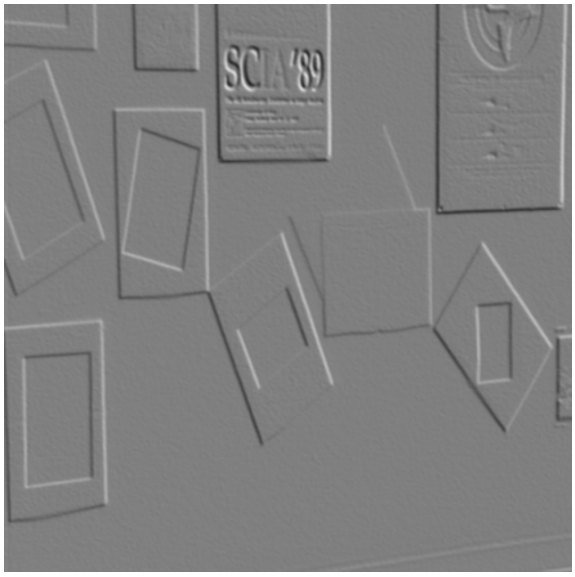

Figura 3: Gradiente en el eje X e Y.

4. Se calcula el modulo mediante los gradientes en X e Y, $modulo = sqrt(G_x^2 + G_y^2)$:

Figura 4: Módulo del gradiente con Sobel

5. Finalmente, se calcula la orientación, en radianes, mediante los gradientes en X e Y, $orientacion = arctan2(G_y, G_x)$:

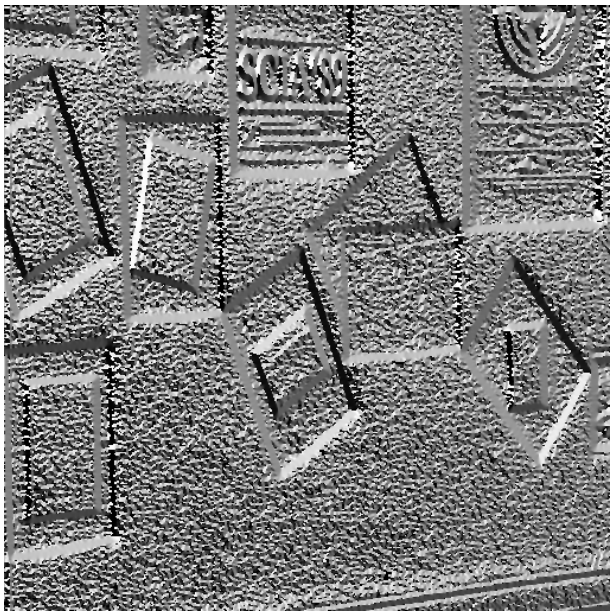

Figura 5: Orientación del gradiente

Una vez implementado Sobel, se implementa Canny:

1. Se calculan la gaussiana y la derivada gaussiana, mediante una función auxiliar que se ha implementado, para sacar las componentes del kernel de Canny.
2. Se hacen algunas cuentas para poner esos kernel en la orientación (vertical u horizontal) que corresponde para cada gradiente.
3. Siguiendo la fórmula de las diapositivas, se convoluciona la imagen primero con los kernel verticales, guardando los gradientes parciales.
4. Luego se convolucionan esos gradientes parciales cada uno con su kernel horizontal correspondiente. Se hacen por separado para reducir el número de cuentas a realizar.

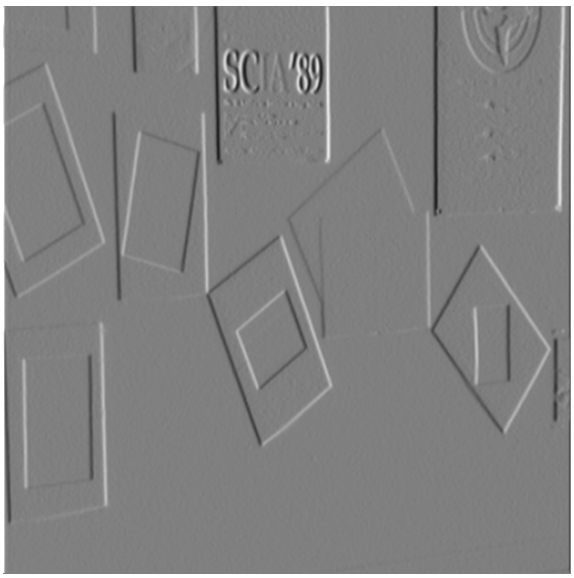

Figura 6: Gradiente en el eje X.

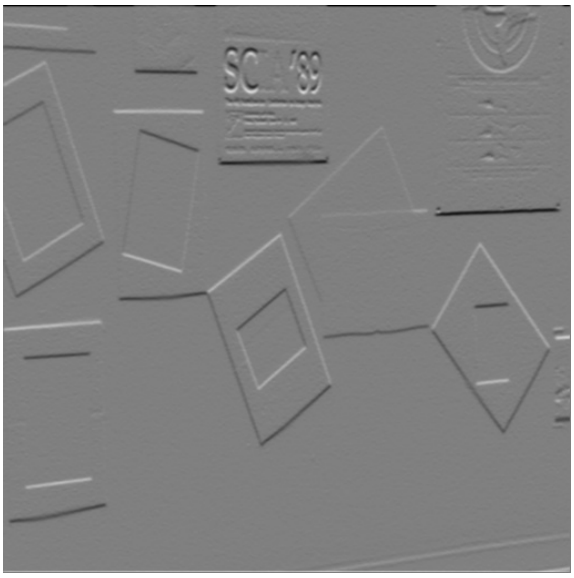

Figura 7: Gradiente en el eje Y.

5. Por último se calculan el gradiente conjunto, el módulo y la orientación de igual manera que en el operador de Sobel.

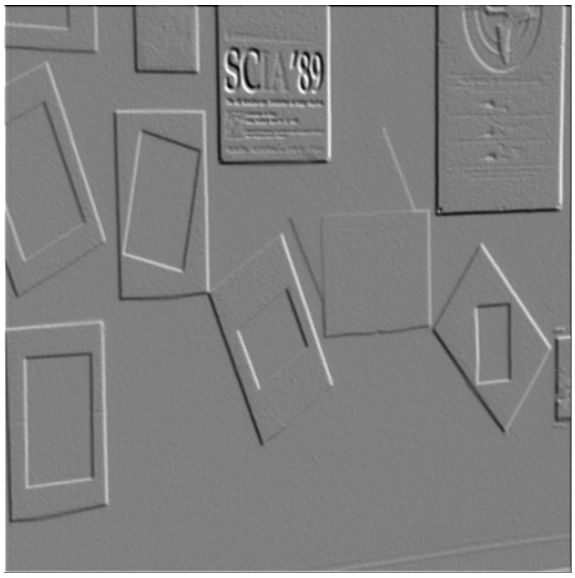

Figura 8: Gradiente en el eje X e Y.

Figura 9: Módulo del gradiente con Canny

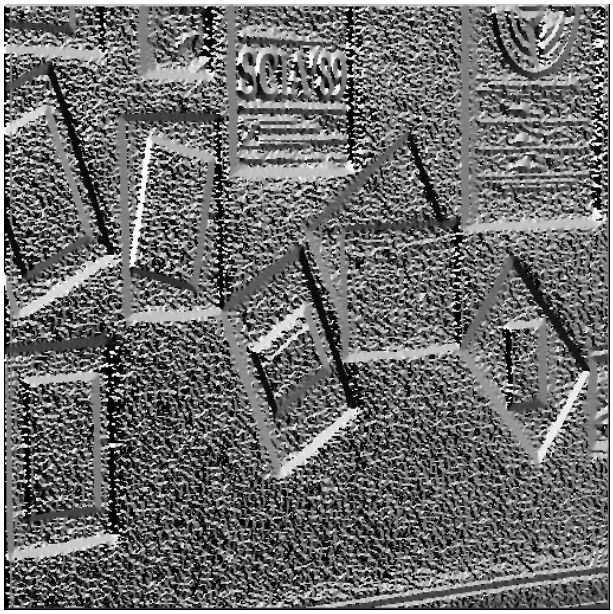

Figura 10: Orientación del gradiente

Observando los valores mínimos y máximos, se puede observar que son bastante dispares:

- En el caso de los gradientes con Sobel son -407 y 374 en X y -366 y 454 en Y, mientras que con Canny son -296 y 378 en X y -553 y 328 en Y.
- Aún transformando las matrices de gradiente, los valores sufrían de desbordamiento, por lo que se ha necesito implementar y aplicar una propia función de normalización de los datos.
- En el caso del módulo también se salen del rango esperado.
- Para la orientación se sigue con la dinámica, aunque en este caso los valores no sobrepasan el máximo para un ángulo, que sería pi o menos pi.

Matriz	**Sobel**		**Canny**	
	Mínimo	Máximo	Mínimo	Máximo
Gradiente en X	-407	374	-296	378
Gradiente en Y	-366	454	-553	328
Módulo	0	454	0	553
Orientación	-pi	pi	-pi	pi

Figura 11: Mínimos y máximos de las diferentes matrices.

3. Punto de fuga

Para el cálculo del punto de fuga se ha implementado una función, $Hough_transform$, que dados el módulo y la orientación del gradiente de una imagen y un umbral de threshold, devuelve el punto de fuga de la imagen. Para ello, para cada píxel, si su módulo de gradiente es mayor que el umbral, y su orientación no es ni vertical ni horizontal, se calcula la recta que pasa por ese punto, de acuerdo a su orientación de gradiente, en coordenadas polares. Si intersecta con la línea del horizonte de la imagen, se añade un voto al punto del horizonte donde intersectan. Al final

devuelve el punto del horizonte que más votos tenga. Para la imagen 'pasillo1.pgm' se ha calculado a mano la altura de la línea del horizonte para representar mejor el punto de fuga.

Después, se dibuja en la imagen original una cruz con centro en el punto de fuga para marcarlo.

Los resultados son los siguientes:

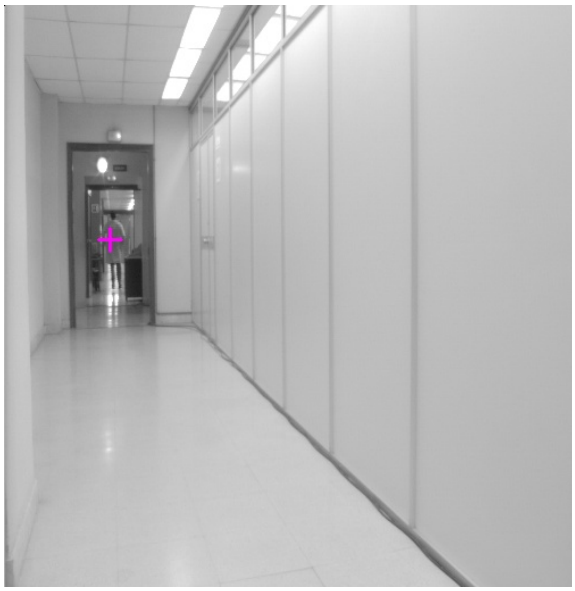

Figura 12: Punto de fuga en pasillo1.pgm.

Figura 13: Punto de fuga en pasillo2.pgm.

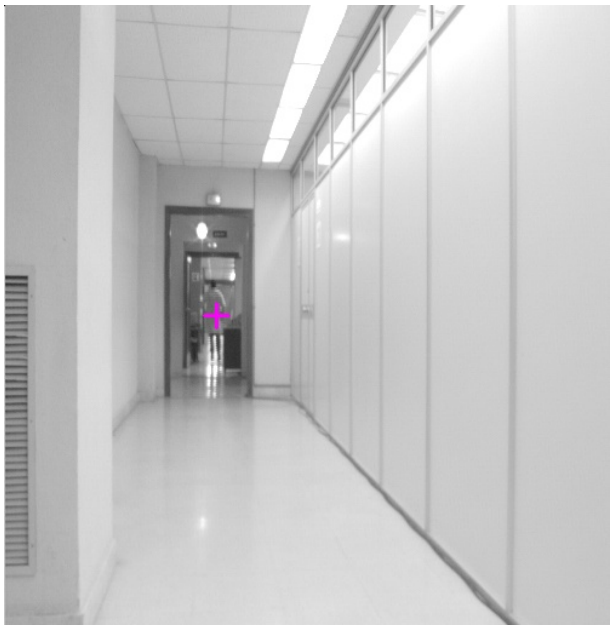

Figura 14: Punto de fuga en pasillo3.pgm.

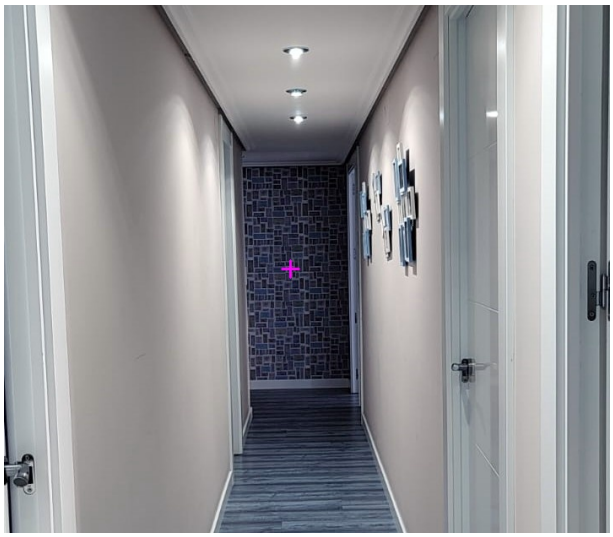

Figura 15: Punto de fuga en pasillo$_casa.jpeg$.@