

Trabajo 2. Contornos

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.

En concreto, se plantean los siguientes objetivos:

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

Para ello pueden usarse:

- Operadores de Sobel o de Scharr. Previamente podría aplicarse un filtrado Gaussiano a la imagen, con sigma variable.
- Operadores de Canny, con sigma variable. Recuerda que **no** debes usar el *Canny Edge Detector* de OpenCV (que hace una serie de operaciones distintas y otras adicionales). Debes implementar tu versión del cálculo del gradiente usando para ello el **operador** de Canny.

Se recomienda prestar atención al tipo de dato adecuado para representar cada resultado, y visualizar todos ellos por pantalla, para comprobar que se calculan correctamente. Para probar el código puede utilizarse la imagen `poster.pgm`, proporcionada con el enunciado, y comparar con los resultados que aparecen en las transparencias de clase.

Razona además sobre los valores máximos y mínimos que se obtienen en cada caso para el gradiente horizontal, vertical, módulo y orientación, relacionándolos con las máscaras utilizadas en cada caso. Comenta asimismo las suposiciones acerca de los sistemas de referencia usados, estableciendo conexiones con las imágenes obtenidas y con las máscaras utilizadas.

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

Para ello, se tomarán como posibles puntos centrales los puntos de la línea del horizonte (fila central de la imagen), con una discretización adecuada. Sólo podrán tener voto los píxeles cuyo gradiente tenga un módulo mayor que un cierto umbral. Para la votación se deberá utilizar la **orientación** del gradiente en dichos puntos. Como imágenes de prueba se utilizarán `pasillo1.pgm` y `pasillo2.pgm`. El programa deberá mostrar la imagen original con una cruz señalando el punto de fuga encontrado.

Ten en cuenta que **no** debes usar `HoughLines()` o `HoughLinesP()` de OpenCV para calcular las líneas en la imagen. En su lugar, debes implementar una versión propia de transformada de Hough, no para extraer rectas, sino para votar por posibles puntos de fuga. Para ello, deberás usar la información extraída en el apartado anterior, incluyendo la orientación y módulo del gradiente.

3. **Opcional:** Desarrollar un programa que procese en vivo las imágenes de la cámara y calcule el punto de fuga, aunque la cámara no esté horizontal. La imagen pasillo3.pgm presenta un ejemplo donde la cámara está ligeramente inclinada.

Se tendrá en cuenta cuántas restricciones acerca de la localización del punto de fuga se relajan con la implementación opcional, por ejemplo:

- 1) Punto de fuga en fila central vs. en cualquier fila de la imagen
- 2) Se admite o no que la cámara esté rotada alrededor del eje del pasillo
- 3) Se admite o no que haya más de un punto de fuga finito

Se valorará además la interfaz gráfica resultante, encaminada a representar sobre las imágenes de forma intuitiva los resultados.

Para ayudarte con esta parte, sobre todo si abor das la tercera opción, puedes acudir a la literatura relacionada. Muchas de las posibles soluciones se basan en el uso de Ransac. Como en este momento aún no sabes cómo funciona (lo veremos un poco más adelante en la asignatura), te proponemos diferentes alternativas. Puedes hacer de momento un procesado exhaustivo, aunque sea más costoso y requiera que lo hagas offline. Alternativamente, puedes usar ideas de muestreo aleatorio, para procesar en vivo las imágenes, aunque ahora mismo se trate de una solución más intuitiva. Si vais a hacer esta parte opcional, acordaos de comentárnoslo en clase / prácticas / tutorías para que os podamos orientar.

Entrega en Moodle:

1. Memoria del trabajo realizado en pdf, explicando cómo habéis implementado los programas, argumentando las decisiones de diseño que hayáis tomado, e incluyendo ejemplos de los resultados.
2. Código completo desarrollado.
3. Opcional: un video donde se muestren en vivo el cálculo del punto de fuga.