
PRÁCTICA 3
DETECCIÓN DE BORDES

Robótica y Visión por Computador
Ing. Telecomunicaciones

Universidad Miguel Hernández

Objetivo de la práctica:

El objetivo de la práctica es realizar un estudio comparativo de diversos algoritmos de detección de bordes existentes en una imagen. Las imágenes utilizadas son en escala de grises de 256 filas por 256 columnas, y se encuentran almacenadas en .jpg.

Realización de la práctica:

1. En primer lugar se debe abrir el workspace creado en la práctica 0 o bien generar uno nuevo tal y como se explicó en dicha práctica.
2. Añadimos un nuevo proyecto al workspace, que llamaremos **practica3**
 - a. Si estamos trabajando sobre el workspace anterior, los proyectos `cv.dsp`, `cvaux.dsp` y `highgui.dsp` ya están añadidos, y lo único que debemos hacer es establecer las dependencias y los settings del nuevo proyecto creado.
 - b. Si hemos creado un proyecto nuevo, se deben añadir los proyectos `cv.dsp`, `cvaux.dsp` y `highgui.dsp` tal y como se explicó en la práctica 0 y luego establecer la configuración adecuada.
 - c. Desde el explorador de Windows, creamos un directorio `\images` dentro de la carpeta `\practica3` y colocamos las imágenes de la práctica.
3. Escribir el código que realice las siguientes acciones:
 - a. Abrir las imágenes `inicial.jpg`, `ruido_5.jpg`
 - b. Declarar una imagen resultado con una profundidad de campo de 16 bits (`res`) y una imagen resultado con una profundidad de campo de 8 bits (`res8`)

Para las **dos** imágenes iniciales, aplicar los diferentes detectores existentes en OpenCV:

- Sobel. Componente horizontal. Probar con diferentes tamaños de kernel y observar los resultados.
 - Sobel. Componente vertical. Probar con diferentes tamaños de kernel y observar los resultados.
- NOTA: Observar que cuando el kernel es mayor que 1x3 o 3x1, el filtrado gaussiano ya está implícito en el uso de la función `cvSobel`.*
- Sobel. Gradiente en x y en y. Observar el efecto del tamaño del kernel.
 - Laplaciana. Observar que la sigma viene determinada por el tamaño del kernel utilizado. Probar para `apertura_size=1` y `apertura_size=3`.
 - Canny. Observar el efecto de los parámetros `Threshold` y `apertura_size` en el algoritmo. En la imagen `ruido_5.jpg` se pide utilizar la función `cvSmooth` de la *practica 2* para filtrar la imagen antes de utilizar el detector de Canny. Anotar los resultados.

c. Mostrar las imágenes obtenidas por pantalla.

NOTA: Para los detectores de Sobel y Laplace, leer la ayuda sobre cómo evitar el overflow que se encuentra en la documentación de la función `cvSobel`.

d. Liberar memoria.

4. Anotar para cada caso anterior la conclusión obtenida.

La siguiente figura muestra como ejemplo las imágenes obtenidas aplicando el detector de Canny con una máscara de tamaño 3x3 a las imágenes dato.

