

## **Trabajo de Visión por Computador**

### **TRABAJO-1: Filtrado y Detección de regiones**

Implementación

**VALORACIÓN TOTAL: 8 puntos**

**Fecha de entrega: 20 de octubre**

#### **Informe a presentar**

Para este trabajo como para los demás proyectos debe presentar un informe escrito con sus valoraciones y decisiones adoptadas en cada uno de los apartados de la implementación. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. ( hacerlo en pdf ). La entrega de código sin memoria explicativa no puntúa.

**Normas de la entrega de Prácticas:** EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PERDIDA DIRECTA DE 1 PUNTO CADA VEZ QUE SE DETECTE UN INCUMPLIMIENTO.

1. El código se debe estructurar como un único fichero main que irá llamando de forma secuencial a distintas funciones, una por cada apartado de la práctica.
2. El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
3. Todos los ficheros juntos se podrán dentro de un fichero zip/rar..
4. SOLO ENTREGAR EL CODIGO FUENTE. ( NO INCLUIR IMÁGENES)
5. Los path que se usen en la lectura de imágenes o cualquier fichero de entrada debe ser siempre "imagenes/nombre\_fichero"
6. Todos los resultados numéricos serán mostrados por pantalla. No escribir nada en el disco.
7. La práctica deberá poder ser ejecutada de principio a fin sin necesidad de ninguna selección de opciones. Para ello se deberá de fijar los parámetros por defecto que se consideren óptimos.
8. Poner puntos de parada para mostrar imágenes o datos por consola y el final de cada apartado

**Forma de entrega:** Subir el zip al Tablón docente de CCIA (decsai.ugr.es).

**Este trabajo de implementación tiene como objetivo principal mostrar cómo usando técnicas de filtrado lineal es posible extraer información relevante presente en una imagen que permite su interpretación.**

**1.- USANDO LAS FUNCIONES DE OPENCV : escribir funciones que implementen los siguientes puntos: ( 1 puntos)**

- A) El cálculo de la convolución de una imagen con una máscara 2D. Usar una Gaussiana 2D (GaussianBlur) y máscaras 1D dadas por `getDerivKernels`). Mostrar ejemplos con distintos tamaños de máscara, valores de sigma y condiciones de contorno. Valorar los resultados.**
- B) Usar la función Laplacian para el cálculo de la convolución 2D con una máscara normalizada de Laplaciana-de-Gaussiana de tamaño variable. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores de sigma: 1 y 3.**

**2.- IMPLEMENTAR funciones para las siguiente tareas (4 puntos)**

- A. (0.5) Una función que genere una representación en pirámide Gaussiana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes y justificar la elección de los parámetros (0.5)**
- B. (1.5) Una función que genere una representación en pirámide Laplaciana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes. (1)**
- C. (2) Construir un espacio de escalas Laplaciano para implementar la búsqueda de regiones usando el siguiente algoritmo:**
  - a. Fijar sigma**
  - b. Repetir para N escalas**
    - i. Filtrar la imagen con la Laplaciana-Gaussiana normalizada en escala
    - ii. Guardar el cuadrado de la respuesta para el actual nivel del espacio de escalas
    - iii. Incrementar el valor de sigma por un coeficiente  $k$ . ( 1.2-1.4)
  - c. Realizar supresión de no-máximos en cada escala**
  - d. Mostrar las regiones encontradas en sus correspondientes escalas. Dibujar círculos con radio proporcional a la escala.**

**3.- Imágenes Híbridas: (SIGGRAPH 2006 paper by Oliva, Torralba, and Schyns). (3 puntos)**

Mezclando adecuadamente una parte de las frecuencias altas de una imagen con una parte de las frecuencias bajas de otra imagen, obtenemos una imagen híbrida que admite distintas interpretaciones a distintas distancias ( ver hybrid images project page).

Para seleccionar la parte de frecuencias altas y bajas que nos quedamos de cada una de las imágenes usaremos el parámetro sigma del núcleo/máscara de alisamiento gaussiano que usaremos. A mayor valor de sigma mayor eliminación de altas frecuencias en la imagen

convolucionada. Para una buena implementación elegir dicho valor de forma separada para cada una de las dos imágenes ( ver las recomendaciones dadas en el paper de Oliva et al.). Recordar que las máscaras 1D siempre deben tener de longitud un número impar.

Implementar una función que genere las imágenes de baja y alta frecuencia a partir de las parejas de imágenes ( solo en la versión de imágenes de gris) . El valor de sigma más adecuado para cada pareja habrá que encontrarlo por experimentación

1. Escribir una función que muestre las tres imágenes ( alta, baja e híbrida) en una misma ventana. (Recordar que las imágenes después de una convolución contienen número flotantes que pueden ser positivos y negativos)
2. Realizar la composición con al menos 3 de las parejas de imágenes
3. Construir pirámides gaussianas de al menos 4 niveles con las imágenes resultado. Explicar el efecto que se observa.

**BONUS :** Solo se tendrán en cuenta los bonus si se ha logrado al menos el 75% de los puntos en la parte obligatoria.

**1.- Implementar con código propio la convolución 2D con cualquier máscara 2D de números reales usando máscaras separables. (2 puntos)**

**2.- Realizar todas las parejas de imágenes híbridas en su formato a color (1 punto) (solo se tendrá en cuenta si la versión de gris es correcta)**

**3.- Realizar una imagen híbrida con al menos una pareja de imágenes de su elección que hayan sido extraídas de imágenes más grandes. Justifique la elección y todos los pasos que realiza ( 1 punto)**