

Visión por Computador

TRABAJO-1: Filtros de Máscaras

Implementación

Valoración Total: 8 puntos

Fecha de entrega: 28 de octubre

Informe a presentar

Para este trabajo como para los demás proyectos se 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 QUE SE HAYAN DADO)
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/.rar a PRADO

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 SOLO FUNCIONES BASICAS DE OPENCV : escribir funciones Python que implementen de forma eficiente el cálculo de los siguientes puntos: (2.5 puntos)

- A) Máscaras discretas de la funciones: Gaussiana, derivada de la Gaussiana y segunda derivada de la Gaussiana, todas ellas en el caso 1D.
- B) El cálculo de la convolución de una imagen con una máscara Gaussiana cuadrada 2D de dimensiones inferiores a las de la imagen. Comparar su funcionamiento con la salida de la función de OpenCV `GaussianBlur` para el mismo tamaño de máscara.
- C) Comparar las máscaras 1D calculadas en los puntos anteriores para el cálculo de las derivadas de una imagen usando alisamiento Gaussiano y las máscaras 1D dadas por la función `getDerivKernels`. Mostrar ejemplos con distintos tamaños de máscara (dibujar las máscaras), valores de sigma. Leer la implementación de OpenCV y valorar los resultados.
- D) Usar las implementaciones de los puntos A y B para calcular las máscaras normalizadas de la Laplaciana de una Gaussiana para un sigma dado. Visualizar las máscaras encontradas y mostrar ejemplos de funcionamiento usando dos tipos de bordes (cero y replicados) y dos valores de sigma: 1 y 3.

2.- IMPLEMENTAR funciones para las siguiente tareas (2.5 puntos)

Usar las funciones implementadas en el punto.1

- A. (1) Una función que genere una representación en pirámide Gaussiana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes replicados y justificar la elección de los parámetros
- B. (1.5) Una función que genere una representación en pirámide Laplaciana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes replicados.

3.- Imágenes Híbridas: (SIGGRAPH 2006 paper by Oliva, Torralba, and Schyns). (2 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. A veces es necesario 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 puede haber una región de posibles valores.

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).

1. Buscar el valor/es de sigma más adecuado/s para cada pareja
2. Escribir una función que normalice los valores de los píxeles al intervalo $[0,1]$ y 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)
3. Realizar la composición con al menos 3 de las parejas de imágenes
4. Construir pirámides gaussianas de al menos 4 niveles con las imágenes híbridas encontradas. Explicar el efecto que se observa a lo largo de la pirámide.

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

1.- 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)

2.- 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)