Trabajo de Visión por Computador

TRABAJO-1

VALORACIÓN TOTAL: 18 puntos (10 teoría y 8 programación)

Filtrado y Muestreo

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 puntua.

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 fín 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
9. NO ESTA PERMITIDO EL USO DE CODIGO C/C++ que no sea absolutamente estándar. Por ejemplo no se permite dimensionar matrices a partir de variables o usar tipos de datos específicos de un compilador.

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

CUESTIONARIO DE TEORÍA: (estará disponible en la web más adelante)

Este trabajo de implementación tiene como objetivo aprender a manejar las operaciones de convolución y correlación, la representación de imágenes mediante una pirámide y analizar algunas propiedades de las frecuencias presentes en las imágenes.

1.- USANDO LAS FUNCIONES DE OPENCV : filter2D, GaussianBlur, Scharr, Sobel, getDerivKernels, getGaussianKernel, sepFilter2D, Laplacian. pyrUp(), pyrDown(), (matploptlib) subplot(), escribir funciones que implementen los siguientes puntos: (5 puntos)

A) **Una función que sea capaz de representar varias imágenes con sus títulos en una misma ventana. Usar esta función en todos los demás apartados**
B) **Una función de convolución con máscara gaussiana de tamaño variable y sigma variable. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
C) **Una función de convolución con núcleo separable de tamaño variable. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
D) **Una función de convolución con núcleo de 1ª derivada de tamaño variable. .Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
E) **Una función de convolución con núcleo de 2ª derivada de tamaño variable. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
F) **Una función de convolución con núcleo Laplaciana-de-Gaussiana de tamaño variable. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
G) **Una función que genere una representación en pirámide Gaussiana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**
H) **Una función que genere una representación en pirámide Laplaciana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando dos tipos de bordes y dos valores distintos de sigma.**

Valorar la influencia del tamaño de la máscara y el valor de sigma sobre la salida, en todos los casos.

2) 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.

1. **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**
2. **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)**
3. **Realizar la composición con ela menos 3 de las parejas de imágenes**

BONUS :

1.- Cálculo del vector máscara Gaussiano: Sea $f(x) = exp(-0.5\ \frac{x^2}{\sigma^2})$ **una función donde** σ **(sigma) representa un parámetro en unidades píxel. Implementar una función que tomando sigma como parámetro de entrada devuelva una máscara de convolución 1D representativa de dicha función. Justificar los pasos dados (Ayuda: comenzar calculando la longitud de la máscara a partir de sigma y recordar que el valor de la suma de los valores del núcleo es 1) (1 punto)**

2.- Implementar una función que calcule la convolución de un vector señal 1D con un vector-máscara de longitud inferior al de la señal usando condiciones de contorno reflejada. La salida será un vector de igual longitud que el vector señal de entrada. (Ayuda: En el caso de que el vector de entrada sea de color, habrán de extraerse cada uno de los tres vectores correspondientes a cada una de las bandas, calcular la convolución sobre cada uno de ellos y volver montar el vector de salida. Usar las funciones split() y merge())) (2 puntos)

3.- Implementar una función que tomando como entrada una imagen y el valor de sigma calcule la convolución de dicha imagen con una máscara Gaussiana 2D. Usar las funciones implementadas en los dos bonus anteriores. (1 punto)

3.- Construir una pirámide Gaussiana de al menos 5 niveles con las imágenes híbridas calculadas en el apartado anterior. Mostrar los distintos niveles de la pirámide en un único canvas e interpretar el resultado. Usar implementaciones propias de todas las funciones usadas (2 punto)

4.- Realizar todas las parejas de imágenes híbridas en su formato a color (1.5 puntos) (solo se tendrá en cuanta si la versión de gris es correcta)