

Procesamiento en el dominio de la frecuencia

1. Descripción

En esta tarea se utilizarán los conceptos de filtrado en los dominios de espacio y frecuencia. Se utilizarán clases de convolución y de transformada discreta de Fourier para aplicar un filtro pasa-bajos a una imagen en escala de grises.

2. Objetivos

Al finalizar la tarea el estudiante podrá:

1. aplicar máscaras de filtrado (o *kernel*) a imágenes por medio de la convolución.
2. filtrar imágenes en el dominio de la frecuencia.
3. medir tiempos de ejecución de algoritmos.

3. Metodología

Para realizar esta tarea cree un programa que utilice la OpenCV y

1. compruebe cuánto se tarda en convolucionar una imagen con una máscara gaussiana para diferentes tamaños de máscara (ver `cv::getGaussianKernel`)

Para medir tiempos utilice las clases

`std::chrono::duration` y `std::chrono::high_resolution_clock`.

de C++11 (no olvide indicar a su compilador que está utilizando C++11 o C++14 con `-std=c++11`). Utilice máscaras de filtros de tamaño impar y las funciones `cv::filter2D` y `cv::sepFilter2D` para analizar las diferencias de tiempo en el espacio cuando se emplean filtros separables..

Para calcular la desviación estándar σ de los filtros utilice la fórmula $\sigma = ((s + 2)/6)$ con s el tamaño del filtro.

2. Compruebe cuánto tarda en ser filtrada una imagen con los mismos filtros utilizado anteriormente, pero en el dominio de la frecuencia.

Las clases que necesita para esto están descritas en detalle [en el sitio de la biblioteca](http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html)¹. Tenga especial cuidado con el relleno de ceros (*padding*) requerido para obtener los resultados

¹http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html

correctos. Asegúrese de que sus resultados en ambos dominios son equivalentes. Es normal obtener pequeñas divergencias por errores de precisión numérica, sin embargo, por selección errónea del tipo de borde, estos errores pueden crecer considerablemente con el tamaño de los filtros.

Recuerde que para medir tiempo se utilizan varias corridas para calcular el valor medio, donde la medición de tiempo es una sola para n aplicaciones del algoritmo a medir. También puede utilizar algún perfilador para esta tarea (por ejemplo, **valgrind**).

3. Grafique (utilizando por ejemplo, gnuplot) los tiempos de ejecución de convolución y en el dominio de la frecuencia en función del tamaño de la máscara y determine a partir de qué tamaño de máscara es más eficiente convolucionar en el dominio de la frecuencia.

4. Entregables

1. Los archivos fuente. Asegúrese que el nombre del ejecutable y el archivo fuente sean **tarea05** y **tarea05.cpp** respectivamente. Entregue además el archivo **CMakeLists.txt** o **Makefile** que utiliza para configurar la compilación de su tarea (se recomienda usar cmake).
2. La gráfica de tiempos en formato **pdf**.
3. Fecha de entrega Jueves 15 de Noviembre.