Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Profesor: Dr. Pablo Alvarado Moya
CE-5201 Procesamiento y Análisis de Imágenes Digitales
I Semestre 2017

Tarea 3

I Descripción

En esta tarea se utilizarán los conceptos de filtrado en el tiempo y la 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.

II Objetivos

Al finalizar la tarea el estudiante podrá:

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

III Metodología

Para realizar esta tarea usted debe crear un programa que

1. Compruebe cuánto se tarda en convolucionar una imagen con una máscara gaussiana, para diferentes tamaños de máscara y de imagen.

   Para medir tiempos utilice la clase `lti::timer`, `std::chrono::high_resolution_clock` o su equivalente en `boost`. Utilice máscaras de filtros cuadradas de tamaño de lado impar del tipo `lti::gaussKernel2D<float>` (separable) y `lti::octagonalKernel<float>` (no separable), y la clase `lti::convolution`, o sus equivalentes en la biblioteca OpenCV.

   Para calcular la varianza σ^2 de los filtros de convolución puede utilizar la fórmula $\sigma^2 = \left(\frac{s+2}{6}\right)^2$ donde s es el tamaño del filtro.

   Observe que la medición de una única "corrida" para un tamaño de máscara y de imagen *no* es suficiente para obtener un dato de medición válido. Deberá por tanto medir la duración total de n corridas, y luego obtener el promedio por corrida dividiendo dicho total entre n.

   Cree una tabla para el filtro separable y otra para el filtro no separable. Cada tabla debe tener en una dirección el tamaño de la imagen filtrada (en número total de píxeles) y en el otro el tamaño de kernel (también en número total de píxeles). Los tamaños de la imagen deben variar entre 64×48 píxeles y 1920×1080 píxeles, con al menos 10 valores intermedios. Los tamaños del filtro deben ser cuadrados, impares

y variar entre 3×3 y 1023×1023, con al menos 10 valores en ese rango. De este modo, la medición requerirá llenar dos tablas con al menos 100 entradas, donde para cada entrada usted deberá filtrar la imagen varias veces para poder promediar el tiempo. Nótese que para tamaños de imagen y máscara de filtro pequeños, es necesario promediar con más corridas que para filtros e imágenes grandes. Tome esto en cuenta en el diseño de su medidor de tiempo.

Observe que el tiempo requerido para estas mediciones obligará a dejar el proceso corriendo por varias horas.

Realice dos gráficas en octave con estos resultados (use por ejemplo `mesh`, para graficar los resultados). Para ello, en la primera parte puede crear un archivo con los datos, que usted puede cargar desde octave.

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

   Para realizar este punto debe buscar formas eficientes para:

   2.1. calcular las respuestas al impulso de los filtros a utilizar.
   2.2. completar el tamaño de la respuesta al impulso con ceros (*padding*), para ajustar el tamaño de tal modo que al final no ocurra aliasing. Para ello puede usar por ejemplo la clase `lti::boundaryExpansion`
   2.3. calcular la respuesta en frecuencia del filtro, calculando la DFT (utilizando la FFT) de la respuesta al impulso del filtro aumentada con ceros. Estos filtros puede pre-calcularlos
   2.4. calcular la DFT (utilizando la FFT) de la imagen de entrada, a la que debe haber completado con ceros como corresponda (*padding*)
   2.5. multiplicar las magnitudes de la DFT de la imagen con la de la respuesta en frecuencia del filtro.
   2.6. calcular la transformada inversa el resultado.
   2.7. eliminar el borde ampliado

   Las clases que necesita para hacer esto son `lti::fft`, `lti::ifft`, y el método `emultiply` de la clase `lti::channel`. Puede usar los métodos equivalentes en la OpenCV.

   En principio puede excluir la conversión del kernel de la medición temporal, pero puede hacer ambas mediciones por completitud. Realice las mismas mediciones de tiempo que en el caso de la convolución, pero en este caso solo considerando el filtro 2D (pues el caso separable se maneja de forma idéntica). Genere nuevas gráficas donde superponga los resultados de medición de tiempo para los filtrados en la frecuencia y en el espacio.

3. Verifique que los resultados del filtrado tanto en la frecuencia como en el espacio sean equivalentes. Proponga una forma de comparar los resultados del filtrado en los dos dominios y justifique posibles divergencias entre ellos.

IV Entregables

1. Un archivo `pdf` con las gráficas generadas en GNUPlot y con la descripción de su diseño experimental de medición, y los resultados de divergencia de los procesos convolutivos en el espacio y en la frecuencia.
2. Los archivos fuente. Asegúrese de que el nombre del ejecutable y archivo fuente sean `tarea03` y `tarea03.cpp` respectivamente.
3. Una demostración del programa el día de entrega.