

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
HIGH PERFORMANCE COMPUTING
PARCIAL 2

FECHA ENTREGA: OCTUBRE 28 DE 2015

1. Crear un programa que cargue una imagen desde un archivo, la lleve a escala de grises y realice un filtro de Sobel sobre esta imagen.
2. Se deben crear dos imágenes de salida:
 - a. Una es la imagen donde se resaltan los bordes de la imagen inicial una vez se le ha pasado el filtro de Sobel secuencial.
 - b. La otra es la imagen donde se resaltan los bordes una vez se le ha pasado el filtro de Sobel en paralelo.
3. Se deben crear tres programas distintos, que muestren la implementación de la convolución usando sólo memoria global, usando memoria constante y usando memoria compartida.
4. Se deben tomar tiempos de ejecución de cada algoritmo con imágenes de diferentes tamaños, las imágenes a usar estarán disponibles en el servidor, en la carpeta **“./inputs”** (img1.jpg, img2.jpg, img3.jpg, img4.jpg, img5.jpg y img6.jpg).
5. Se deberán realizar **GRÁFICAS** de tiempos de ejecución de cada algoritmo contra las diferentes imágenes.
6. Los tiempos de ejecución de cada algoritmo deberán ser tomados al menos 20 veces y posteriormente crear un promedio de estos tiempos (para cada imagen). El tiempo promedio que se obtenga será el utilizado para realizar las gráficas.
7. Realizar gráficas de aceleración (Speedup) de cada algoritmo.
8. La versión **“secuencial”** del algoritmo deberá ser una ya creada por alguna librería existente.
9. Se deben comparar en una gráfica de barras las diferentes aceleraciones obtenidas para cada versión implementada. (Memoria Global, Memoria Constante, Memoria Compartida)
10. Realizar un capítulo de **CONCLUSIONES**. Donde expliquen los resultados obtenidos en cada gráfica.
11. El código de los puntos y el trabajo en general (tablas, gráficas, entre otros) debe quedar en un repositorio al cual el profesor deberá poder ingresar para realizar las pruebas del caso; el documento resultado del proceso deberá estar en formato PDF y con una **buena estructura :-)**.

NOTAS:

- Todas las gráficas y tablas que construyan deberán tener los títulos, unidades y leyendas necesarias para que sean claras a los ojos de cualquier lector.
- Las conclusiones deberán mostrar de manera clara explicaciones de cada gráfica y justificaciones del por qué se obtuvieron los resultados planteados.

- Se sugiere utilizar opencv para realizar la carga de la imagen, el algoritmo secuencial y la posterior escritura de la imagen a un archivo.
- Las imágenes resultado de cada ejecución se podrán ir descargando desde el sitio web.
- Se solicitará que por favor siempre que lancen los algoritmos utilicen el mismo nombre para sus imágenes. Estas deben escribirse en la carpeta “./outputs/codigo.png”. Esto para evitar que se supere la capacidad de nuestro disco (Recuerden que es una plataforma de prueba).
- Si presentan problemas en la ejecución por favor hacerlo saber de manera cordial a través del apartado para reporte de bugs en la plataforma o a través de un correo electrónico al profesor.
- La fecha de entrega de este parcial será inaplazable. Para este día deberán estar en el correo del profesor las notificaciones correspondientes de cada uno de los repositorios. No se permitirán cambios en el código una vez se cumpla la fecha de entrega.
- En el repositorio [hpccourse](#) se encuentran los códigos implementados aún sin depurar. El código de la versión de Memoria Compartida fue tomado de [aquí](#) las otras dos versiones fueron adaptadas desde las versiones en una dimensión del libro guía y presentan algunos **fallos :-)**
- Para iniciar la implementación pueden crear los programas secuenciales usando OpenCV de manera local.
- Información de ayuda
 - [nvision08](#)
 - Programming Massively Parallel Processor Book
 - [imageConvolution](#)
 - [OpenCV](#)