**HIGH PERFORMANCE COMPUTING**

**PARCIAL 2**

**FILTRO DE SOBEL SECUENCIAL Y PARALELO CON CUDA C Y OPENCV**

**Presentado por:**

**ALEJANDRO MARÍN LÓPEZ**

**Presentado a:**

**JHON HAIBER OSORIO RIOS**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**

**OCTUBRE 20**

**2017**

**INTRODUCCIÓN**

Para el siguiente proyecto se pretende mostrar el cambio que pueden llegar a tener ciertas imágenes mediante el uso del filtro de Sobel.

Dichas imágenes constan de diferentes tamaños siendo aplicadas sobre el método secuencial y paralelo utilizando para ello CUDA C junto a implementaciones de uso de memoria sobre la GPU.

**REQUERIMIENTOS BÁSICOS**

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:
3. Una es la imagen donde se resaltan los bordes de la imagen inicial una vez se le ha pasado el filtro de Sobel secuencial.
4. La otra es la imagen donde se resaltan los bordes una vez se le ha pasado el filtro de Sobel en paralelo.
5. Se deben crear tres programas distintos, que muestran la implementación de la convolución usando sólo memoria global, usando memoria constante y usando memoria compartida.
6. Se deben tomar tiempos de ejecución de cada algoritmo con imágenes de diferentes tamaños, las imágenes a usar se enviarán por correo electrónico.
7. Se deberán realizar GRÁFICAS de tiempos de ejecución de cada algoritmo contra las diferentes imágenes.
8. 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 y cada algoritmo). El tiempo promedio que se obtenga será el utilizado para realizar las gráficas.
9. OJO: Cada algoritmo deberá ejecutarse cómo mínimo con cada imagen 20 veces y obtener tiempos promedios. NOTA: Pilas, no dejar todo para el final, seguramente van a ocupar el clúster todos.
10. Realizar gráficas de aceleración (Speedup) de cada algoritmo.
11. La versión “secuencial” del algoritmo deberá ser una ya creada por alguna librería existente.
12. Se deben comparar en una gráfica de barras las diferentes aceleraciones obtenidas para cada versión implementada. (Memoria Global, Memoria Constante, Memoria Compartida).