Reporte técnico

Adrián Biller Alcántara
A01018940
Septiembre 13

Resumen

El siguiente documento es una descripción del proyecto realizado como tarea de blurring de imágenes utilizando diferentes herramientas de cómputo paralelo

1. Introducción

Dentro de los ejemplos vistos en cuanto a programación paralela, existen múltiples métodos, todos con diferentes características. El principal método en el que cualquier persona programa es desde la aproximación lineal. Sin embargo, como visto en clase, es necesario optimizar ciertos procesos utilizando nuevas herramientas como lo es el cómputo paralelo. En este proyecto se utilizaron dos tipos diferentes de computación paralela: threads con openMP y threads con el uso de una tarjeta de video y cuda toolkit.

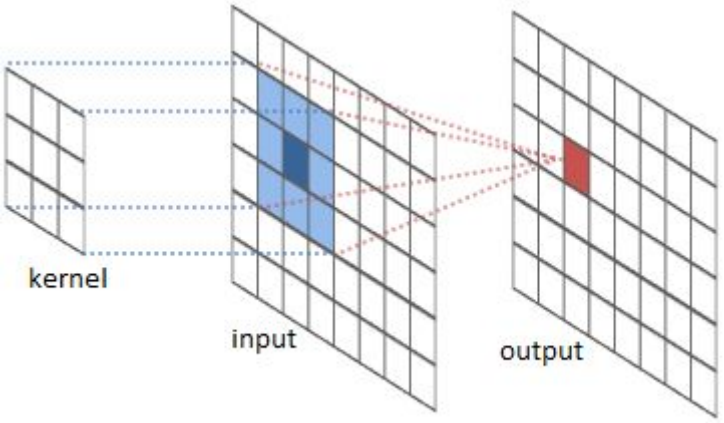
Basándose en ambos métodos se hará una comparación del performance entre ellos para concluir que algoritmo es más eficiente para esta tarea e incluso los detalles de cuál es la mejor configuración.

2. Desarrollo

En esta práctica se desarrollaron múltiples códigos para realizar el efecto borroso en una imagen utilizando las librerías de opencv para la manipulación de imágenes y cuda y openmp para la paralelización de procesos.

En este caso para crear el efecto borroso en una imagen se utilizó el algoritmo de convolución. este consiste que por cada pixel de la imagen a modificar se debe tomar un radio de píxeles alrededor del que se modificará para calcular el promedio y asignarlo al píxel inicial.

En la siguiente imagen se puede mostrar el proceso de cada asignación de color de pixel.



3. Ejemplo de referencias y figuras

En la siguiente tabla se pueden observar los tiempos de ejecución de la multiplicación de matrices dependiendo de la herramienta y en donde se están ejecutando los procesos.

Tiempo de ejecución en milisegundos

	Imagen pequeña (640 x 400)	**Imagen mediana (1620 x 1075)**	**Imagen grande (2560 x 1440)**
CPU Intel core i5 Coffee Lake 2.5GHz	**79.01**	**660.6157**	**1271.159**
CPU Threads Intel core i5 Coffee Lake 2.5GHz	**3.200**	**22.58**	**39.6335**
GPU Servidor con GeForce GTX 670	**0.035**	**0.04156**	**0.042631**

Como se puede observar en esta tabla, el uso de gpu de nuevo para el procesamiento de imagen El speedup según la fórmula $speedup = \frac{Time\ Sequential}{Time\ Parallel}$

y utilizando los números obtenidos salen los siguientes resultados:

	Imagen pequeña (640 x 400)	**Imagen mediana (1620 x 1075)**	**Imagen grande (2560 x 1440)**
CPU Threads Intel core i5 Coffee Lake 2.5GHz	**s =24.690625**	**s = 29.25667404**	**s =32.07284**
GPU Servidor con GeForce GTX 670	**s = 2257.428571**	**s = 15895.46920**	**s =29817.7148**

Aún utilizando el servidor a horas donde se encontraba mucha actividad se puede observar que el speedup que tiene utilizando gpu es considerablemente grande.

4. Referencias

Cheng, John, et al. *Professional CUDA C Programming*. John Wiley & Sons, 2014.