Universidad del Valle de Guatemala
Facultad de Ingeniería
Práctica Profesional
Sección 10

Informe Final

Proyecto 3- CUDA

Presentado por:
Cristina Bautista (161260)
Andrea Paniagua (18733)
Diego Alvarez (14104)

Guatemala, junio 2022

Índice

Introducción

CUDA que en ingles significa "*Compute Unified Device Architecture"* es una plataforma de compuatcion paralela y un modelo de programación que da lugar a incrementos en el rendimiento de computación al aprovechar la potencia de la unidad de procesamiento grafica o GPU (Aller, 2018).

Los núcleos CUDA son unos procesadores paralelo**s** que se encargan de procesar todos los datos que entran y salen de la GPU, realizando cálculos gráficos cuyo resultado los ve el usuario final. Se encuentran dentro de la GPU.

Esta estructura permite a los núcleos trabajar juntos para completar una misma tarea. Así se convierten en otro recurso para que ayuden a CPU a la hora de manejar al mismo tiempo.

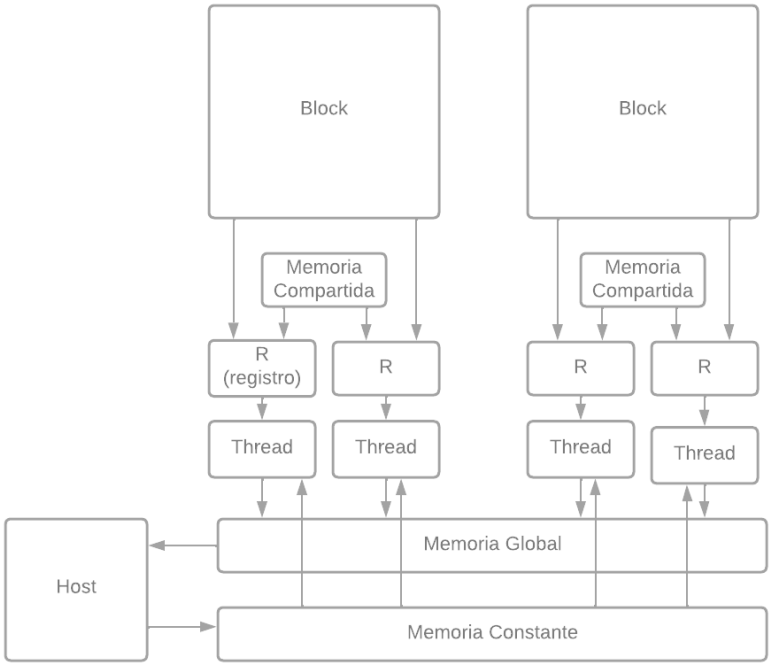
La transformada de Hough es una herramienta que permite detectar curvas en una imagen. Es una técnica muy robusta frente al ruido y a la existencia de huecos en la frontera del objeto. A la hora de aplicar la transformada de Hough a una imagen es necesario obtener primero una imagen binaria de los píxeles que forman parte de la frontera del objeto (Universidad de Jaen, 2021).

Bitácora de Pruebas

Prueba No.	Global	Constante
1	6.00976	6.137824
2	6.015456	6.147968
3	6.011744	6.102368
4	6.032608	6.117952
5	6.040576	6.20096
6	6.037184	6.079008
7	6.06912	6.088864
8	6.049856	6.080704
9	6.102016	6.117344
10	6.116992	6.076448
Promedio	6.0485312	6.114944

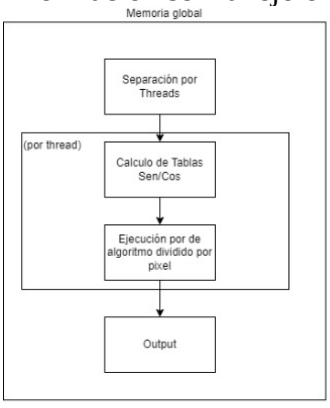
Memoria Constante

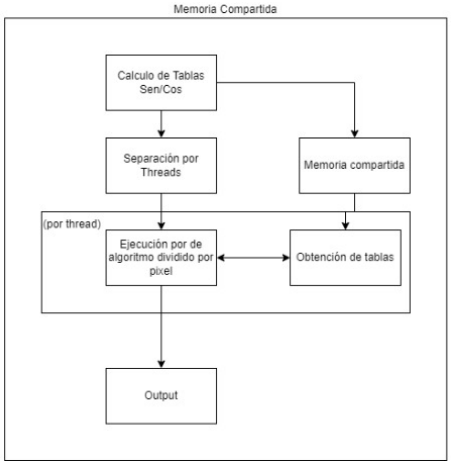
Antes de la memoria constante se tenían dos buckets; para senos y cosenos, pero por thread se tenia que calcular al inicio. Con memoria constante se calculan las buckets y así están disponibles para los threads. Al no calcularlo por thread se ahorran pasos extras.

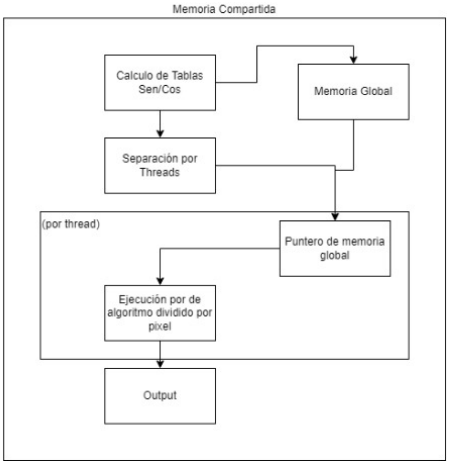


Memoria Compartida

Se trabajo sobre la entrega de la memoria constante. La clausula de cada bloque se debe de ver como matriz que acumula memoria, causando que la matriz local se agrega a la matriz global al final de cada ejecución de cada bloque. Se definió una variable para utilizar los IDs de los hilos del bloque en el kernel y un acumulador local con degreeBins*rBins elementos. Toda su información se manejo en kernel.







Conclusiones

- La memoria constante acelera mucho el tiempo de ejecución para procesos que dependen todos de una misma fuente que no cambia, como era esperado.
- La memoria compartida no acelera el tiempo de ejecución aunque se base en la memoria constante.
- La memoria constante permite hacer el recuento de los threads y tener que hacer ese recuento manualmente.
-

Recomendaciones

- Se recomienda utilizar distintos enfoques para comprender los conceptos de memoria.
- Actividades sobre el uso de cada paso de la memoria compartida o global.
- Realizar una comparativa entre ambas memorias para comprender las diferencias y casos de uso de cada una.

Referencias

- Aller; A. (2018) . "**Qué son los Nvidia CUDA Cores y cuál es su importancia".** Profesional Review. Extraído de: https://www.profesionalreview.com/2018/10/09/que-son-nvidia-cuda-core/
- Universidad de Jaen**. 'Segmentación. Transformada de Hough'. Extraído de: http://www4.ujaen.es/~satorres/practicas/practica4_vc.pdf**
- Usera, J. (2018). 'Tarjetas GráficasTutoriales Núcleos CUDA: cómo funcionan en las tarjetas gráficas de NVIDIA'. Extraído de: https://hardzone.es/2018/05/06/nucleos-cuda-tarjetas-graficas-nvidia/