



Laboratorio 2 – Intercalado Coalescente de Columnas

Computación de Alto Desempeño

2019-1

Número de Equipo: **2**

Modelo de Tarjeta: NVIDIA GTX 770

Compute Capability: 3.0

Integrantes:

Nombre	Rol USM
Benjamin Seider	201573541-7
Cristian Navarrete	201573549-2
Gonzalo Fernández	201673557-7

1. Preguntas

1. Kernel1 : [Pregunta] ¿Encuentra algún problema en su implementación? De ser su respuesta afirmativa, antes de continuar, explique como se le ocurriría solucionarlo.

No existiría coalescencia cuanto X sea muy grande ya que la palabra que tiene los píxeles de la derecha, estará a gran distancia de los de la izquierda, para solucionarlo, se podrían en vez de almacenar las filas concatenadas, almacenar las columnas concatenadas.

2. Kernel2 : [Pregunta] Solo si su respuesta a la [Pregunta] anterior fue afirmativa, indique si esta implementación solucionó el error encontrado.

No lo soluciona.

3. Kernel2: ¿Cuál es el problema de esta implementación? ¿Ocurre en todos los casos? Indique que condiciones se tienen que dar para que se presente este problema de optimización.

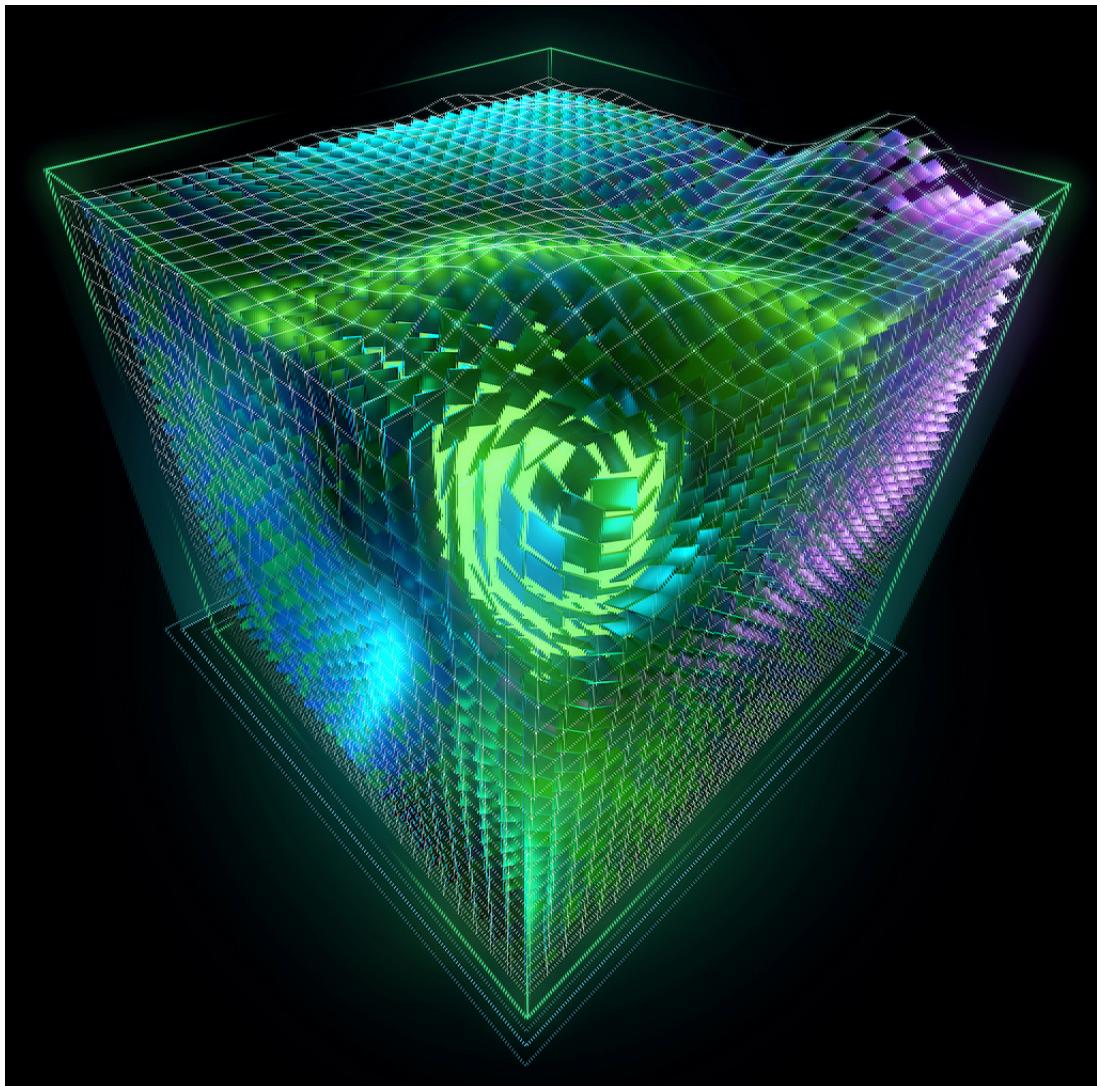
Que con X muy grande no es posible mantener coalescencia, no, solo ocurre cuando $X > 64$.

4. Kernel3: [Pregunta] ¿Se arregló el problema que presentaba el caso anterior? ¿A qué se debe esto?

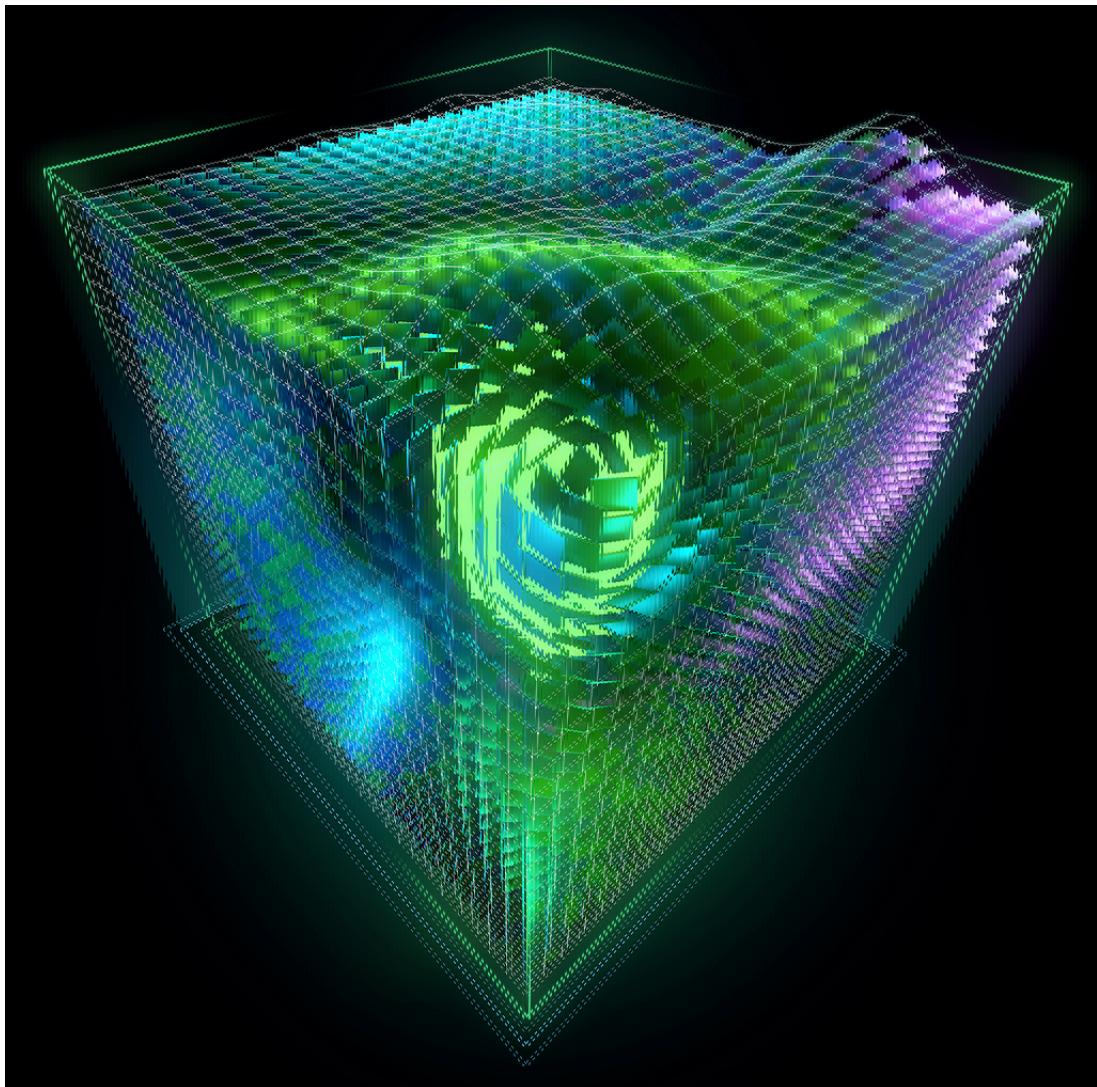
Si, ya que ahora tenemos en memoria concatenadas las columnas, por tanto las hebras siempre leerán y escribirán espacios de memoria conjuntos.

2. Imágenes Obtenidas

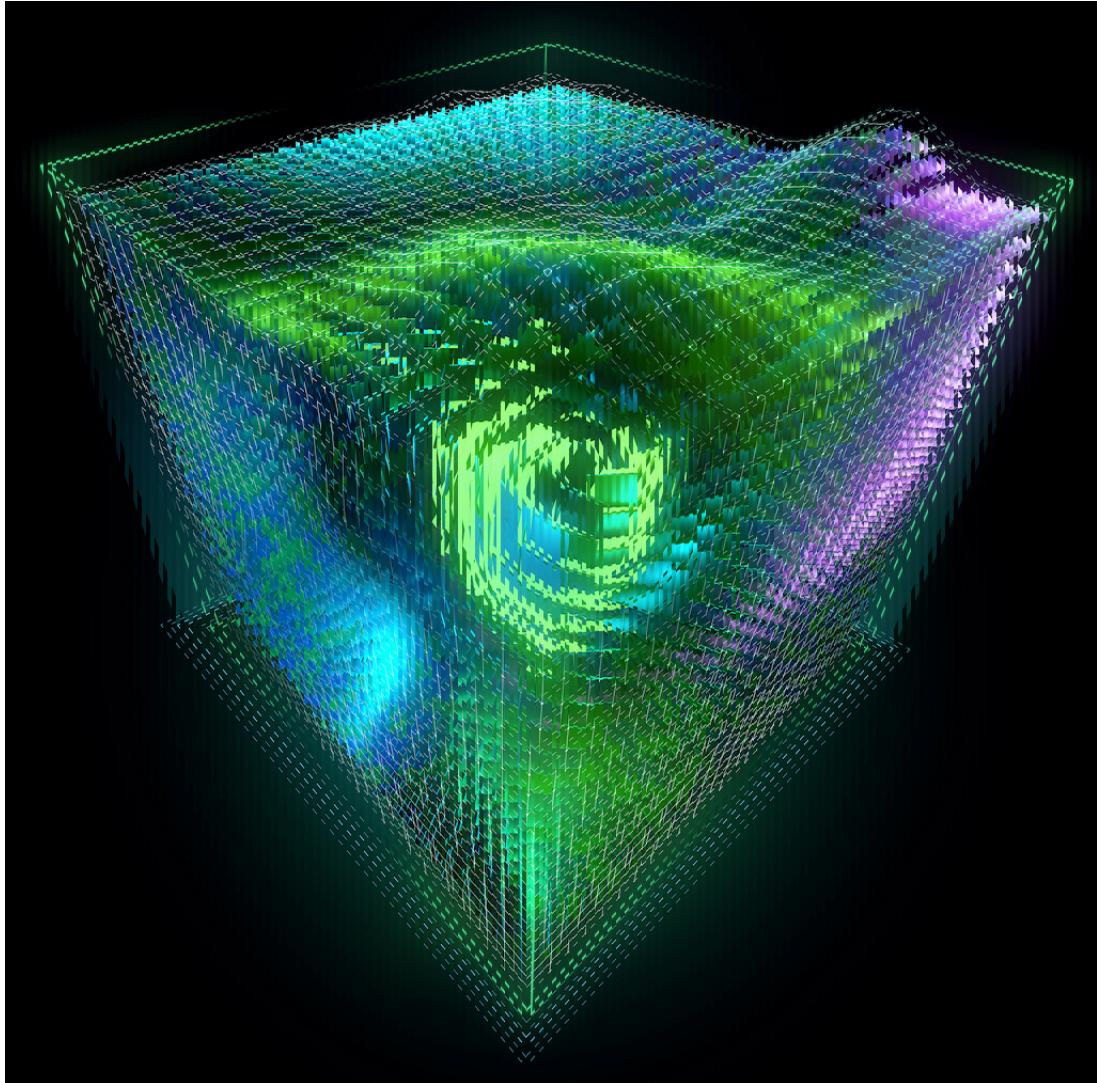
2.1. Imagen 1



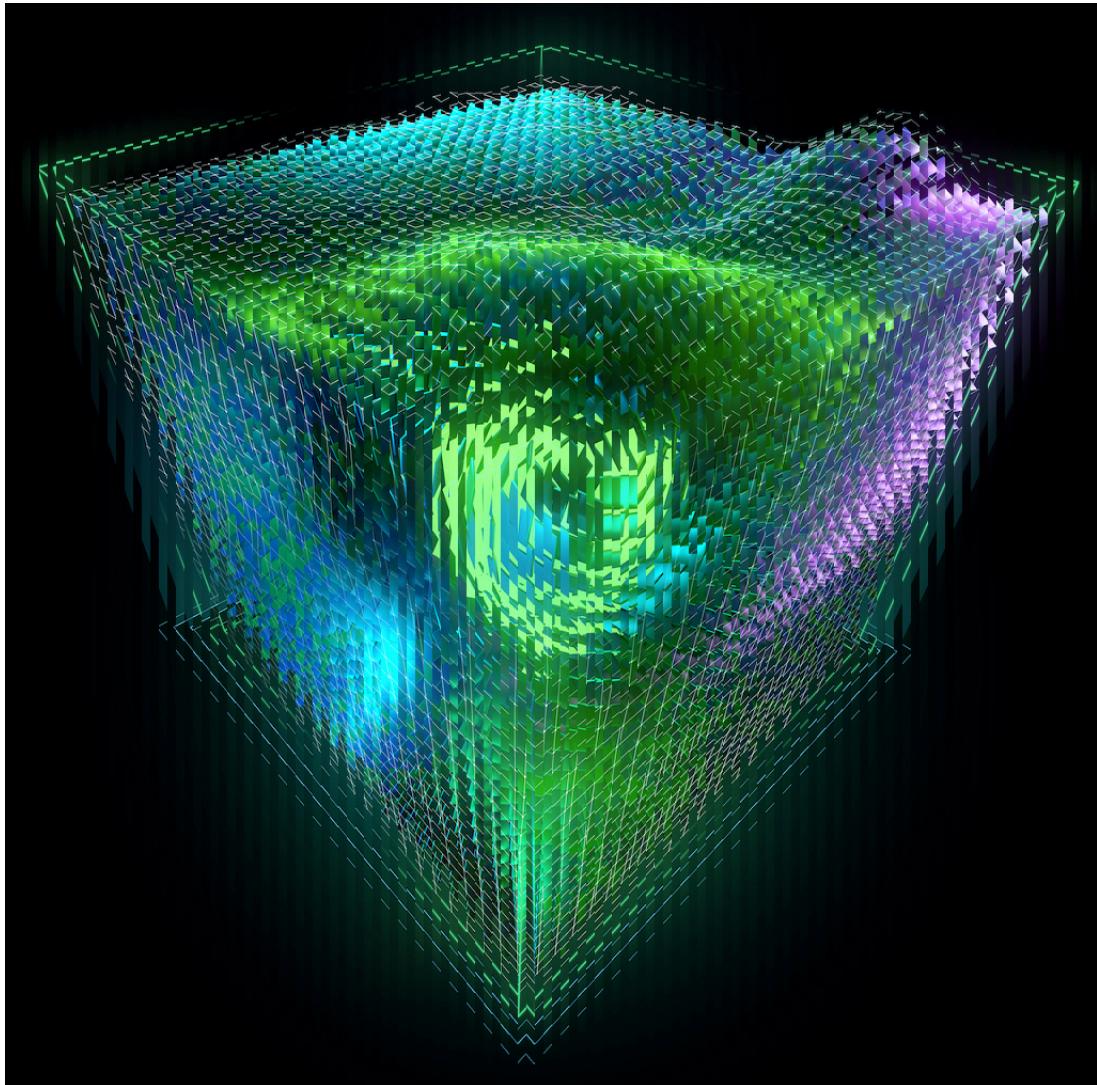
2.2. Imagen 2



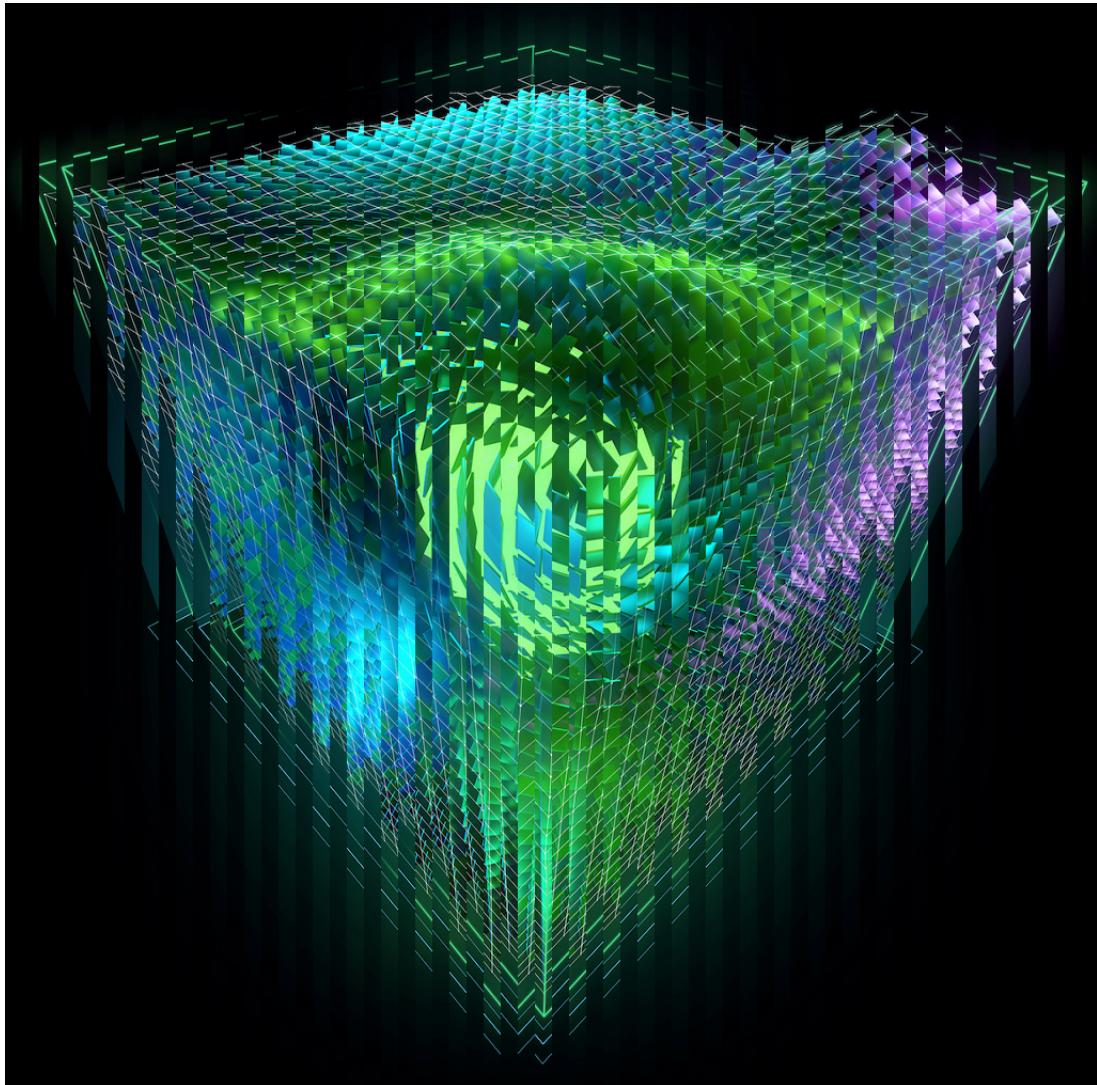
2.3. Imagen 3



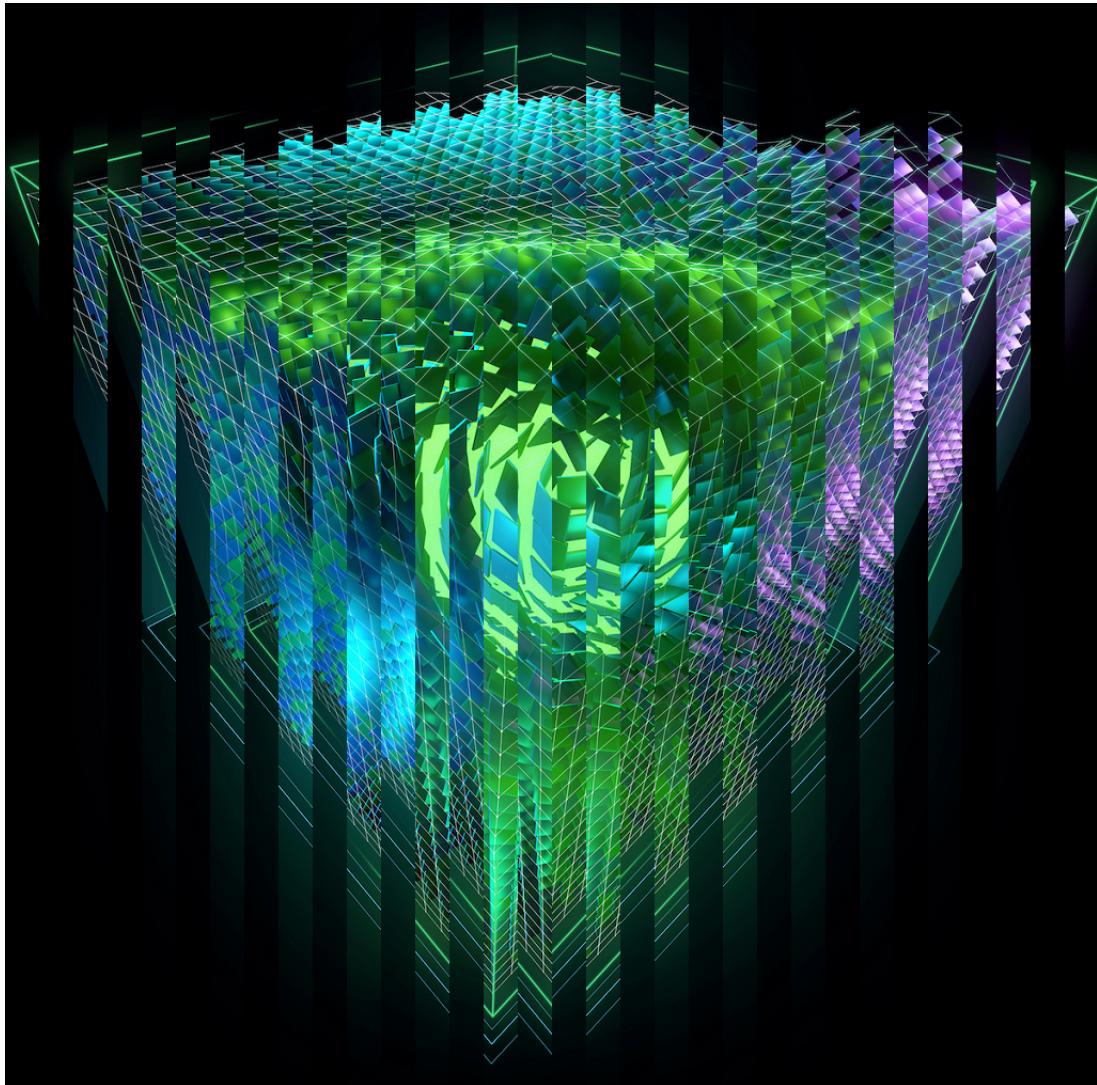
2.4. Imagen 4



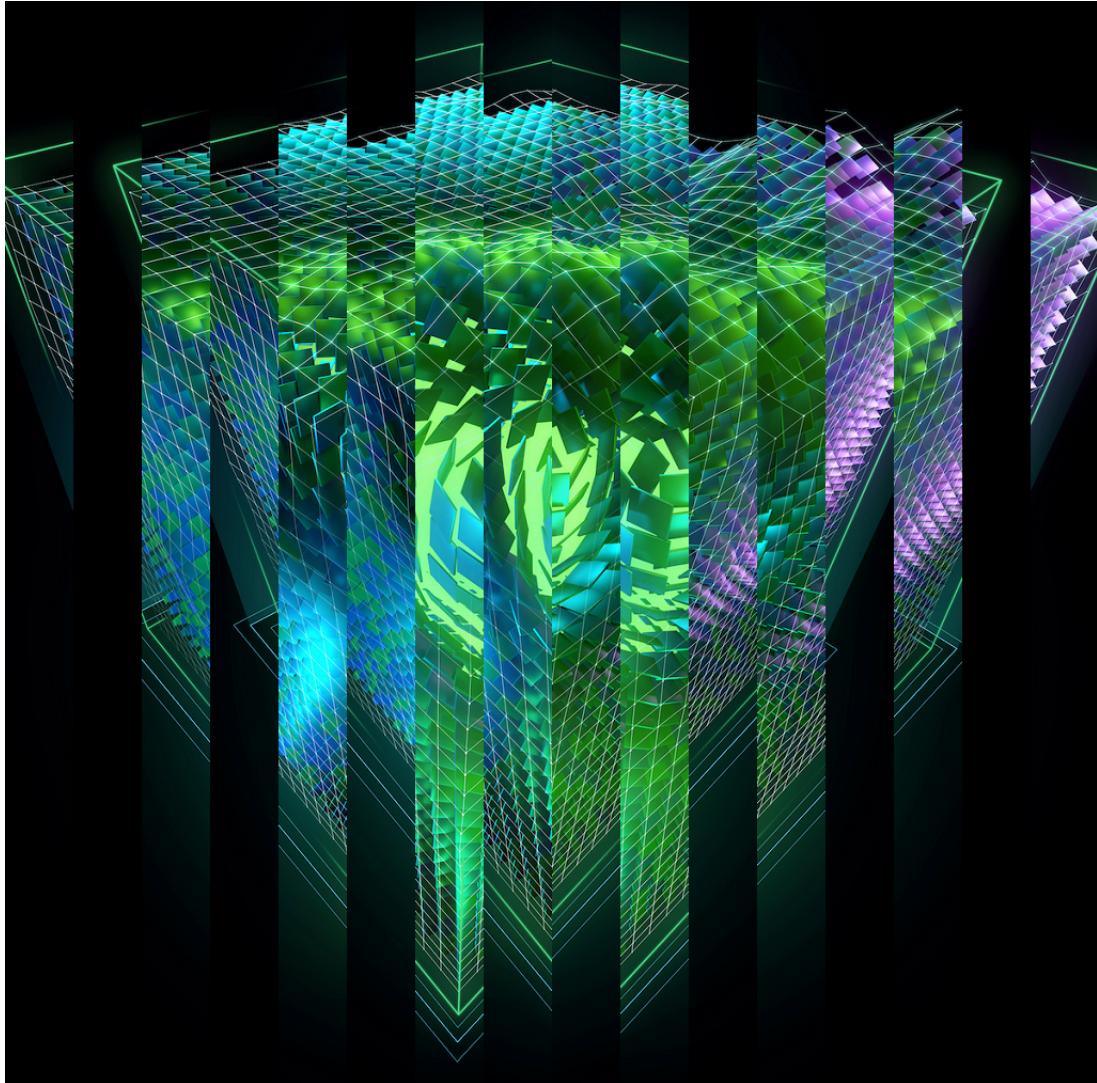
2.5. Imagen 5



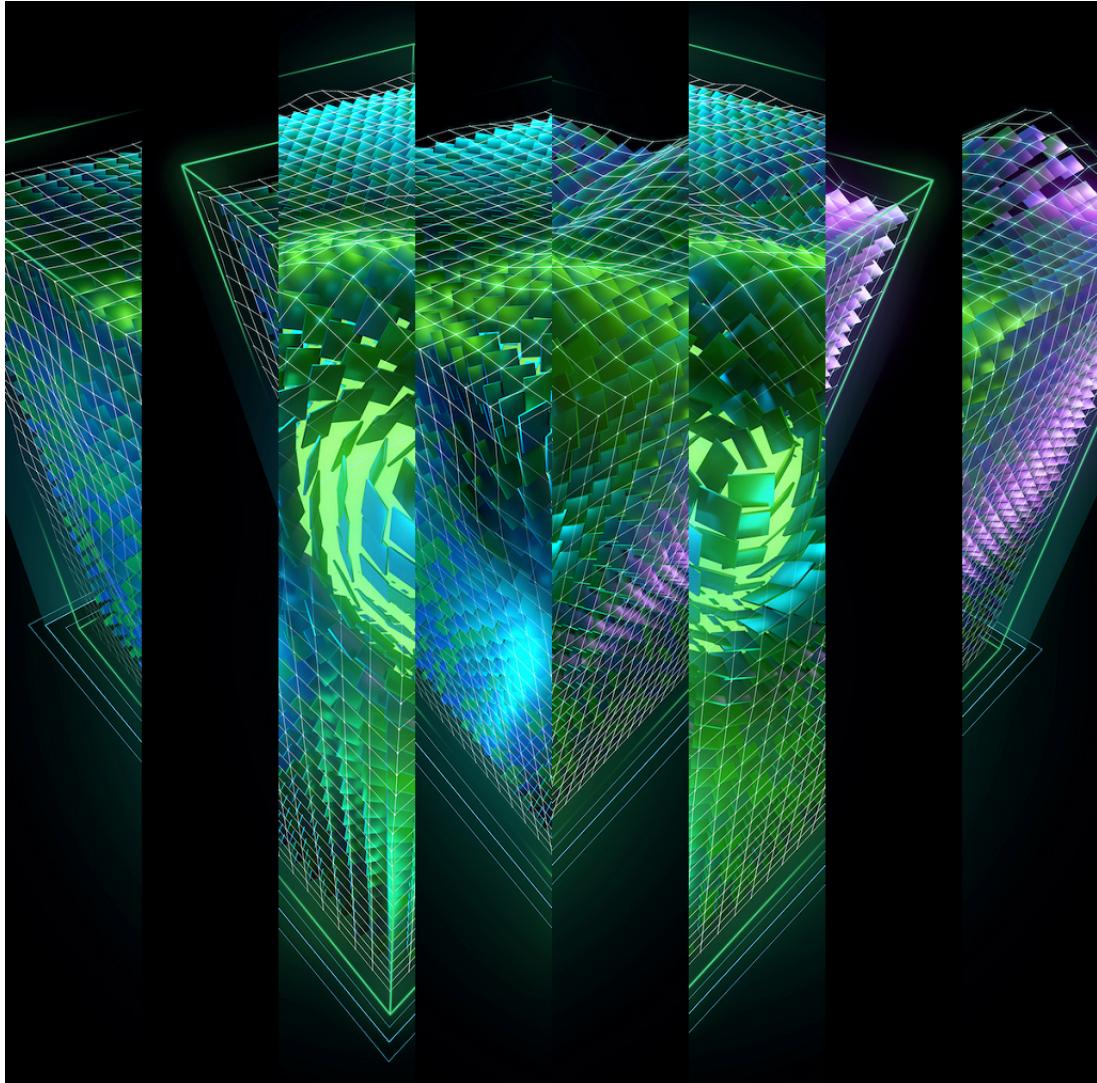
2.6. Imagen 6



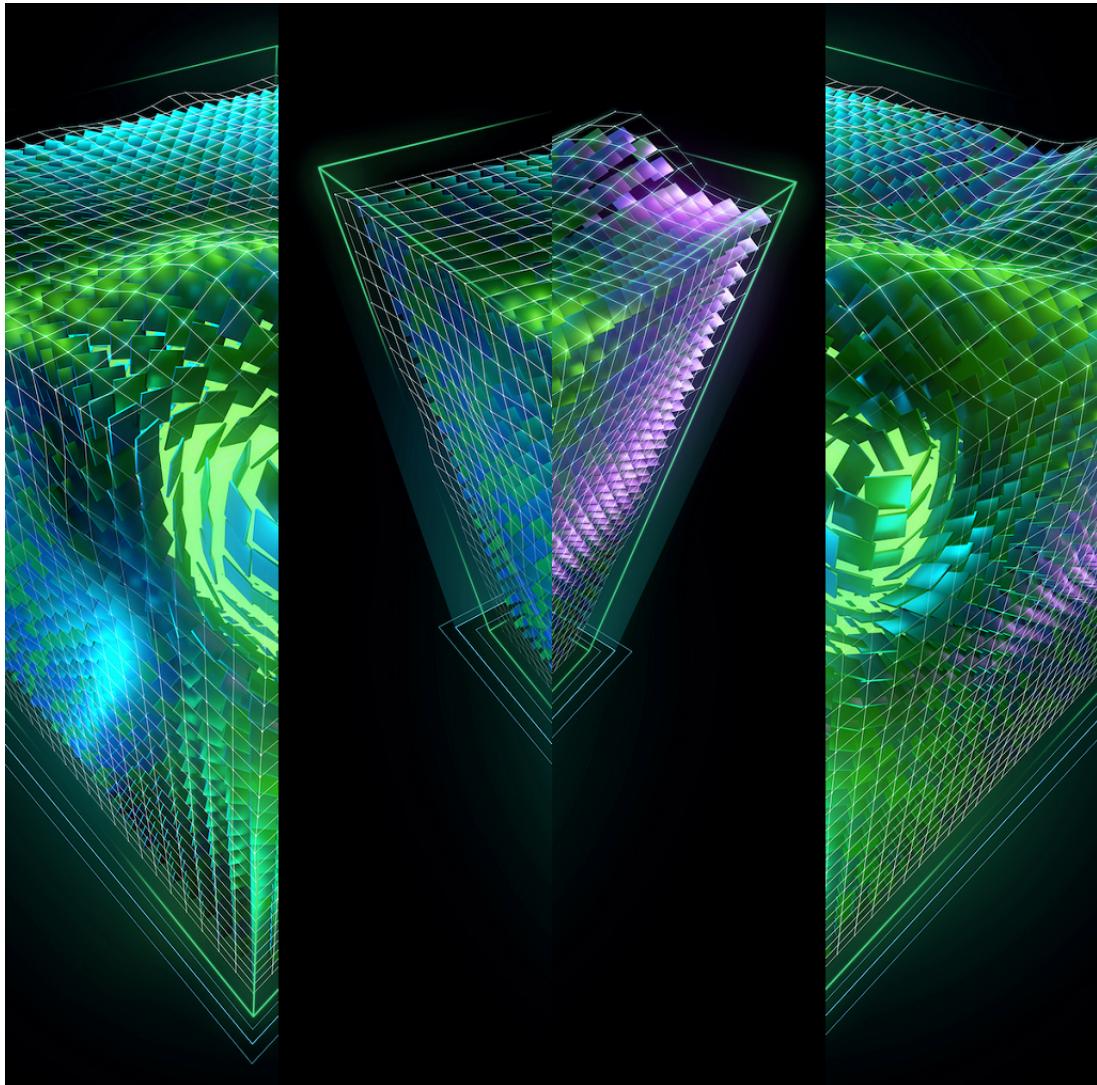
2.7. Imagen 7



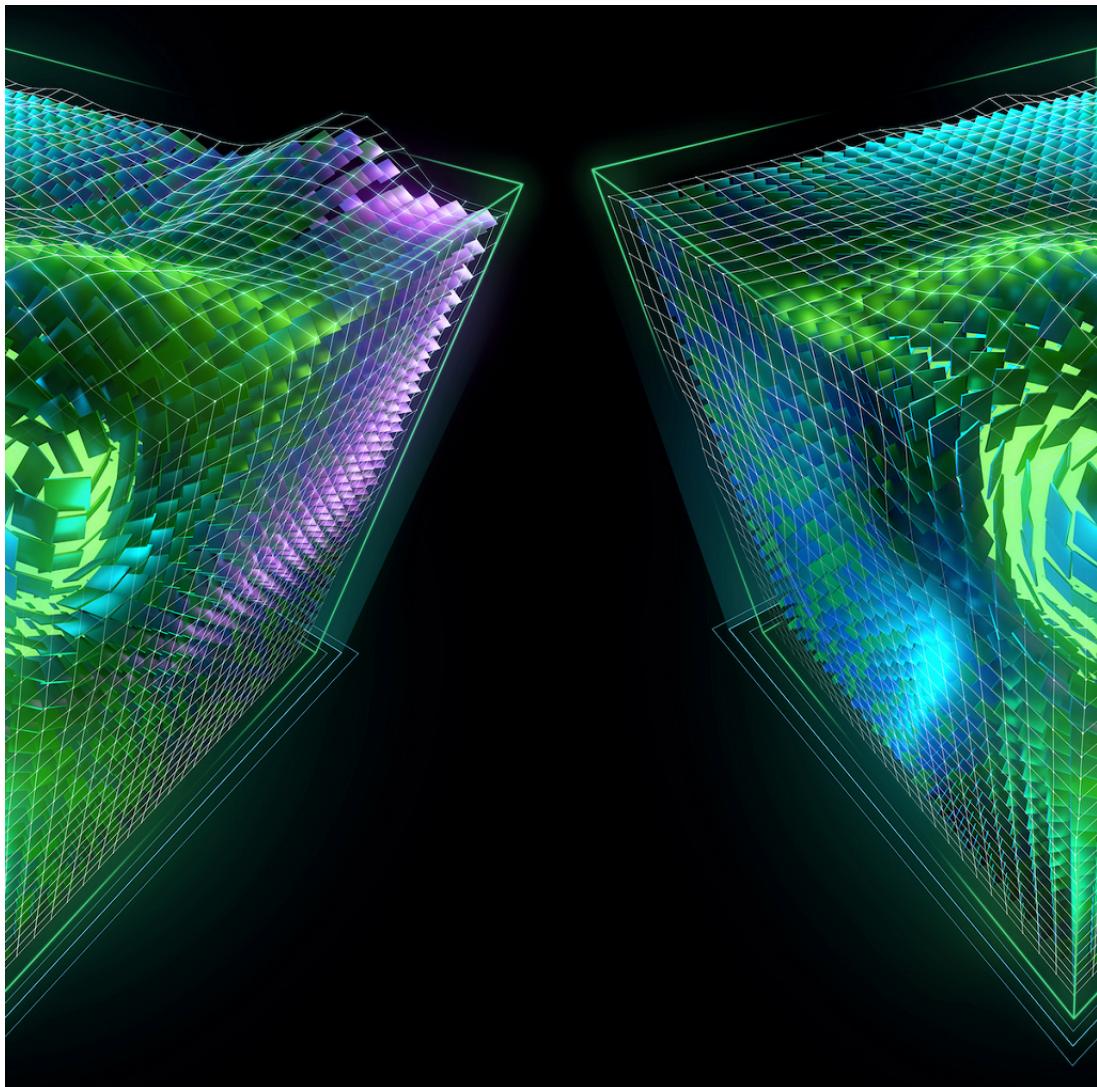
2.8. Imagen 8



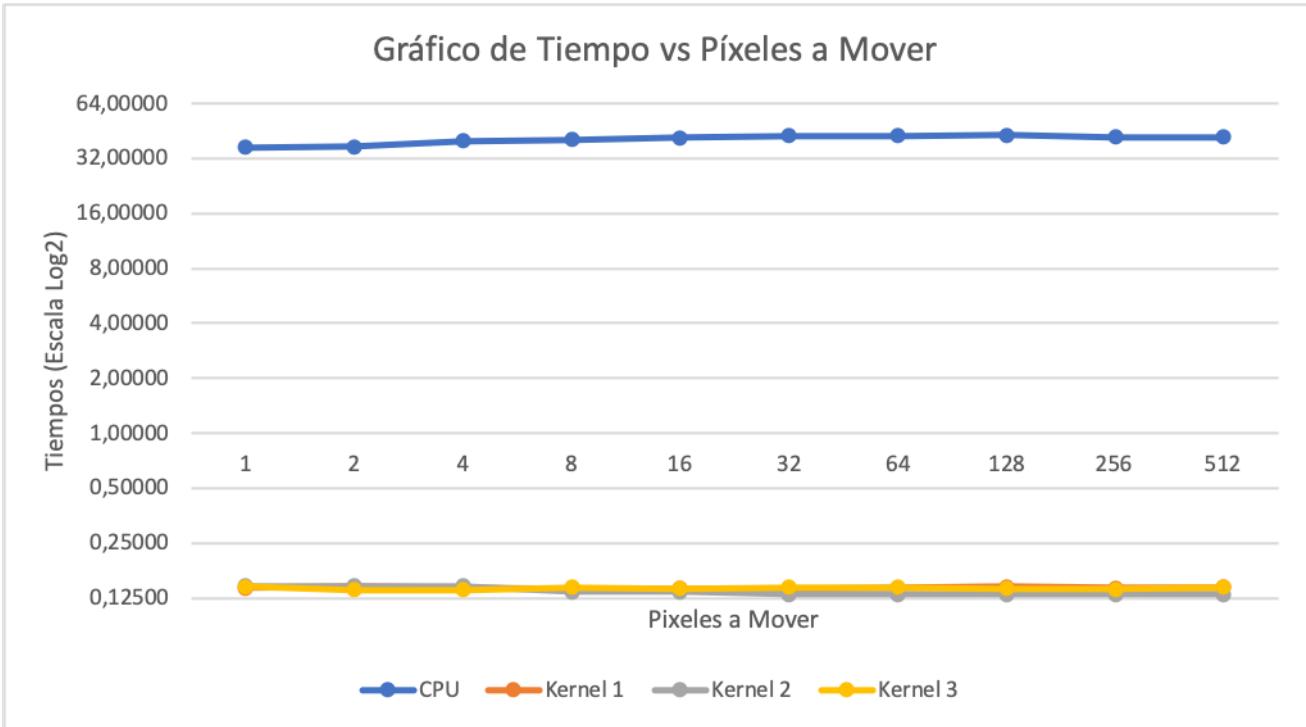
2.9. Imagen 9



2.10. Imagen 10



3. Gráfico de Tiempo vs Píxeles a Mover



-	CPU [ms]	Kernel 1 [ms]	Kernel 2 [ms]	Kernel 3 [ms]
1	36,92900	0,14153	0,14512	0,14413
2	37,27800	0,14441	0,14518	0,13847
4	39,88500	0,14201	0,14499	0,13882
8	40,76400	0,14022	0,13520	0,14307
16	41,63200	0,14102	0,13539	0,14080
32	42,53900	0,14070	0,13107	0,14272
64	42,50600	0,14214	0,13104	0,14343
128	43,16300	0,14393	0,13126	0,14035
256	41,84400	0,14147	0,13072	0,13917
512	41,93500	0,14310	0,13136	0,14320

4. Bandwidth

Kernel 1 [GB/s]	Kernel 2 [GB/s]	Kernel 3 [GB/s]
175,759	165,851	173,897
175,816	166,145	174,840
175,187	166,261	174,727
175,870	139,337	175,733
175,823	139,496	173,917
174,074	144,113	175,156
174,562	144,377	173,850
175,545	144,298	175,316
174,436	144,280	174,779
176,927	144,557	178,520

[PREGUNTA] Analice los datos obtenidos para tiempos y uso de Bandwidth. ¿Qué puede concluir? ¿Se observa realmente una diferencia significativa? ¿A qué cree usted que se debe esto? Comente al respecto.

Se observa una reducción en el Bandwidth entre el Kernel 1 y el Kernel 2, esto se puede deber a que el Kernel 1 resuelve mejor la colasencia.

5. Conclusiones

En este Laboratorio, hemos aprendido a utilizar el Nvidia Visual Profiler para determinar tiempos de uso de Kernel y Bandwidth utilizado, también desarrollamos el concepto de Coalescencia. A diferencia del laboratorio pasado, la GPU fue siempre más rápida que el CPU, esto se debe a que la carga podía ser distribuida de manera equitativa entre las hebras, respecto a los supuestos, el programa asume que las imágenes entregadas pueden ser divididas sin problemas por 1,2,4,8,16,32,64,128,256,512.