

Taller Práctico 4:

Procesamiento de video en algoritmos CUDA con GPU, detectando trayecto de asteroides

Por: Guillermo Andres De Mendoza Corrales
Universidad Sergio Arboleda, Noviembre 2025

1. Objetivo del Laboratorio

El objetivo principal de este taller es desarrollar un algoritmo de procesamiento de video en tiempo casi real capaz de **detectar automáticamente la presencia de un asteroide** y **trazar su trayectoria** dentro de un video de fondo estelar.

2. Requisitos del Algoritmo

El código final de los estudiantes (escrito en Python, utilizando librerías aceleradas por CUDA) debe cumplir con los siguientes requisitos:

1. **Entrada:** Aceptar como entrada un archivo de video (.mp4) simulando la observación del espacio.
2. **Procesamiento:** Utilizar la aceleración por **CUDA (Numba/CuPy)** para realizar las operaciones de detección de forma masivamente paralela y eficiente.
3. **Salida:** Producir dos resultados clave:
 - **Detección:** Una declaración clara que indique si "**Existe Asteroide**" o "**No Existe Asteroide**" en el video.
 - **Trazado:** Una **gráfica (usando Matplotlib)** que muestre la trayectoria detectada del asteroide sobre el fondo estelar. Si no se detecta asteroide, la gráfica simplemente mostrará el fondo

3. Estrategia de Detección Acelerada (CUDA)

Para detectar un asteroide (un objeto en movimiento) contra un fondo estático (estrellas), se utilizará la técnica de **Sustracción de Fondo (Background Subtraction)**. Esta técnica es altamente paralizable y, por lo tanto, ideal para la aceleración con CUDA.

3.1. Creación de la Imagen Comparativa (Fondo Estático)

La primera tarea es obtener una imagen de referencia que contenga **solo las estrellas y el ruido**, pero **sin el asteroide**.

- **Método Sugerido: Apilamiento de Mediana (Median Stacking).**
 - Los estudiantes deben cargar **todos los cuadros** del video como arreglos (arrays).
 - En la **GPU (utilizando CuPy o un kernel de Numba)**, se debe calcular el **valor de la mediana** para cada píxel a lo largo de la dimensión del tiempo.
 - **Resultado:** Una **Imagen de Fondo "I_fondo"** donde el asteroide (que solo está presente en un subconjunto de cuadros) habrá desaparecido.

3.2. Detección de Cambios (Sustracción Acelerada)

Una vez que se tiene la imagen de fondo, el asteroide se aísla mediante una simple resta. Esta es la parte más intensiva y debe acelerarse con CUDA.

- **Proceso:** Para **cada cuadro individual** del video “I_cuadro”, se debe restar la Imagen de Fondo “I_fondo”.
 - Fórmula (ejecutada por un kernel CUDA/función Numba en cada píxel):

$$I_{cambio} = I_{cuadro} - I_{fondo}$$

- **Resultado:** La imagen “I_cambio” contendrá **solo el asteroide** como un punto brillante (positivo) si estaba presente en ese cuadro. El resto del cielo (estrellas) será cercano a cero.

3.3. Umbralización y Generación de la Matriz de Píxeles

Para confirmar la presencia y la posición del asteroide, se aplicará un umbral de brillo a “I_cambio”.

- **Umbralización:** Los estudiantes deberán aplicar un **umbral** (elegido experimentalmente) a I_cambio. Cualquier píxel con un valor mayor al **umbral** se considera parte del asteroide.
 - **Resultado:** Una **matriz binaria/booleana** (también generada en la GPU) donde los valores **1** (True) indican la posición del asteroide y los valores **0** (False) el fondo.
- **Salida de Matriz:** Los estudiantes deben generar una tabla o matriz (en la CPU después de la transferencia de datos) que contenga la siguiente información solo para los cuadros donde se detectó un cambio significativo:

Cuadro (Tiempo)	Posición X (píxel)	Posición Y (píxel)
t_1	x_1	y_1
t_2	x_2	y_2
...














3.4 Resultados

Con los datos de posición (x, y) en función del tiempo (t), los estudiantes procederán al análisis final.

1. **Veredicto de Detección:** Si la tabla de posiciones contiene **puntos** se declara: **"Existe Asteroide"**. De lo contrario: **"No Existe Asteroide"**.
2. **Trazado de la Trayectoria:** Utilizando la tabla de datos, se generará un gráfico de dispersión con las librerías gráficas en python en el eje horizontal y en el eje vertical. Esto revelará el camino lineal recorrido por el asteroide.

4. Videos de prueba

Link: https://drive.google.com/drive/folders/1sf36FSKKJUO3cf49AlJMODJZkusxPjp4?usp=drive_link

Nombre 	Propietario	Fecha de modificación
 video0 - no asteroid	 yo	14:24 yo
 video1	 yo	14:02 yo
 video2	 yo	14:12 yo
 video3	 yo	14:12 yo
 video4	 yo	14:19 yo
 video5	 yo	14:19 yo

5. Elementos a evaluar con nota académica

- Algoritmo ejecuta correctamente
- Código limpio
- Informe de laboratorio

6. Algoritmos a desarrollar

Los algoritmos a desarrollar son dos:

- paralelo CUDA transformando video a escala de grises

Entradas:

- Video con pixeles en color

Salida:

- Indicador booleano (si hay asteroide o no)
- Gráfica de trayectoria cuando se detecta asteroide

7. Análisis de Resultados y Elaboración del Informe de Laboratorio

El análisis es la parte más importante. Debes documentar los resultados en formato de informe técnico.

Presenta los siguientes puntos en un documento estilo informe de laboratorio:

1. **Título y Objetivos:** (Ya definidos).
2. **Marco Teórico:** Brevemente, explica el procesamiento de video.
3. **Metodología:**
 - **Configuración del Hardware**

- **Configuración del Software:** Versiones de Python y librerías
 - **Explicación del algoritmo desarrollado.**
4. **Resultados:** Presenta tus hallazgos de los algoritmos,
 5. **Análisis de Rendimiento:** Compara los resultados obtenidos.
 6. **Conclusiones:** Resume tus aprendizajes sobre la aplicación práctica del procesamiento paralelo.