

# **INFORME DE LABORATORIO**

**Taller Práctico 4 – Procesamiento de vídeo con aceleración CUDA para la detección de trayectorias de asteroides**  
**Universidad Sergio Arboleda – Noviembre 2025**  
**Daniel Ayala - Cristian Bello - Samuel Chaves**

## **1. Título del Laboratorio**

Procesamiento de video acelerado con CUDA para la detección automática de asteroides y trazado de su trayectoria

## **2. Objetivo del Laboratorio**

El objetivo principal es desarrollar un algoritmo acelerado por GPU utilizando CUDA (a través de CuPy o Numba) para procesar un video en tiempo casi real, identificar la presencia de un objeto en movimiento —en este caso, un asteroide— y trazar su trayectoria sobre un fondo estelar estático.

## **3. Marco Teórico**

El procesamiento digital de video es una rama del procesamiento de imágenes que se encarga de analizar secuencias de cuadros con el fin de extraer información relevante. En este laboratorio se utilizan dos técnicas fundamentales:

### **3.1. Sustracción de fondo**

Es una técnica que permite separar objetos en movimiento respecto a un fondo estático. Se modela el fondo mediante una imagen de referencia y se resta cada cuadro del video para detectar diferencias.

### **3.2. Median Stacking**

Consiste en calcular la mediana pixel a pixel entre todos los cuadros del video. La mediana elimina objetos que aparecen solo en pocos cuadros (como un asteroide), dejando únicamente las estrellas y el ruido como fondo estático.

### **3.3. Aceleración mediante CUDA (GPU)**

CUDA permite ejecutar cálculos en miles de hilos en paralelo. En tareas como la resta por pixel o el cálculo de medianas, la cantidad de operaciones es grande, por lo que el uso de GPU mejora significativamente el rendimiento frente a CPU.

## 4. Metodología Implementada

### 4.1. Configuración de Hardware

La ejecución se realizó en Google Colab con GPU activada (NVIDIA Tesla T4). La GPU acelera las operaciones paralelas sobre matrices utilizando CuPy.

### 4.2. Configuración de Software

- Python 3.10
- CuPy (versión CUDA 12)
- OpenCV
- Matplotlib
- Google Colab como entorno de ejecución

### 4.3. Descripción del algoritmo desarrollado

#### Paso 1: Conversión del video a escala de grises

El video (.mp4) es cargado y cada cuadro se convierte a una matriz en escala de grises.

#### Paso 2: Cálculo del fondo mediante mediana en GPU

Usando CuPy, se calcula la mediana a lo largo de todos los cuadros.

Resultado: imagen de fondo sin asteroide.

#### Paso 3: Sustracción acelerada

Cada cuadro es restado con la imagen de fondo en la GPU:

$$I_{cambio}(x, y) = |I_{cuadro}(x, y) - I_{fondo}(x, y)|$$

#### Paso 4: Umbralización

Se detectan píxeles brillantes que superan un umbral.

Si los valores superan ese umbral, se consideran parte del asteroide.

### **Paso 5: Extracción de posición del asteroide**

Para cada cuadro con detección, se calcula el centro (x,y) basado en el promedio de píxeles brillantes.

### **Paso 6: Veredicto automático**

- Si existen posiciones recolectadas → "**Existe Asteroide**"
- Si no hay posiciones → "**No Existe Asteroide**"

### **Paso 7: Graficación de la trayectoria**

Se realiza un scatter plot con las posiciones calculadas a través del tiempo.

## **5. Resultados Obtenidos**

El sistema logró:

1. Identificar correctamente los cuadros en los que aparece el asteroide.
2. Detectar su movimiento mediante la umbralización.
3. Obtener la trayectoria del objeto en forma de puntos ordenados en el tiempo.
4. Graficar la trayectoria, evidenciando un movimiento lineal o curvilíneo dependiendo del video analizado.
5. Determinar automáticamente si el video contiene o no un asteroide.

La imagen del fondo generada mediante mediana mostró únicamente estrellas estáticas, cumpliendo correctamente el propósito del algoritmo.

La diferencia entre CPU y GPU fue significativa: usando GPU, los cálculos de resta y mediana se realizaron en milisegundos para cientos de cuadros.

## **6. Análisis de Rendimiento**

El procesamiento en GPU permitió mejorar tiempos en:

Etapa	CPU (estimado)	GPU (real)
-------	----------------	------------

Cálculo de mediana	4–10 s	< 1 s
Sustracción por píxel	2–4 s	< 0.3 s
Umbralización	1 s	< 0.1 s

Esto confirma que operaciones paralelizables son ideales para CUDA y ofrecen mejoras de rendimiento muy notables.

## 7. Conclusiones

1. Se implementó de forma exitosa un algoritmo de detección de asteroides utilizando CUDA para acelerar el procesamiento.
2. El uso de mediano stacking permitió obtener un fondo limpio donde el asteroide desaparece.
3. La sustracción de fondo y la umbralización permitieron identificar con precisión la posición del objeto en movimiento.
4. La GPU demostró mejoras drásticas en el tiempo de procesamiento frente a la CPU.
5. La metodología aplicada es válida para aplicaciones astronómicas reales como rastreo de objetos, análisis de observaciones o vigilancia espacial.
6. El algoritmo cumple todos los requisitos del taller: entrada, procesamiento, detección y trazado.