

Paralelización en GPU para la maximización de la ganancia espectral

Manual de Usuario

Tabla de contenido

1	Descripción.....	3
2	Requisitos.....	3
3	Iniciar aplicación	3
4	Realización de pruebas	4
5	Resultados.....	5

Tabla de Ilustraciones

Ilustración 1. Imagen multiespectral.....	4
Ilustración 2. Imagen pancromática	4
Ilustración 3. Imagen de salida	5
Ilustración 4. Comparación de CPU y GPU.....	5

1 Descripción

Esta aplicación permite la ejecución de la transformada Parallel Fusimagery a la cual se le pasan cuatro parámetros siendo los dos primeros las imágenes multiespectral y pancromática, las cuales por medio de un tratamiento digital de imagen se unirán para obtener una imagen satelital de mayor calidad, el tercer parámetro es el nivel y el cuarto el nombre de la imagen de salida. Esta aplicación ya ha sido implementada de forma secuencial, por ello en esta versión se realiza la paralelización de la transformada con el objetivo de optimizar los tiempos de respuesta.

2 Requisitos

Para el funcionamiento de este software, es necesario el siguiente hardware:

1. Tarjeta gráfica: NVIDIA Tesla K80

También es necesario el siguiente software:

1. Python 2.7: <https://www.python.org/download/releases/2.7/>
2. Numpy 1.16.0: <https://pypi.org/project/numpy/>
3. Pillow 5.3.0: <https://pillow.readthedocs.io/en/latest/installation.html#basic-installation>
4. Matplotlib 3.0.2: <https://matplotlib.org/users/installing.html>
5. Pycuda 2018.1.1: <https://pypi.org/project/pycuda/>
6. Skcuda 0.5.2: <https://scikit-cuda.readthedocs.io/en/latest/install.html>

3 Iniciar aplicación

Para iniciar la aplicación, ubique su carpeta a través de la consola. Ahora, utilice el siguiente comando para realizar la ejecución de la aplicación en CPU:

```
python fusion_cpu.py parametro1 parametro2 parametro3 salida
```

Y el siguiente comando para realizar la ejecución de la aplicación en GPU:

```
python fusion_gpu.py parametro1 parametro2 parametro3 salida
```

Como se puede evidenciar tienen la misma cantidad de parámetros, a continuación se hace la descripción de cada uno de ellos

- parametro1: String con la ruta absoluta de la imagen multiespectral a utilizar de tamaño $A \times A$ donde A es un número diádico, es decir, 2^n .
- parametro2: String con la ruta absoluta de la imagen pancromática a utilizar de igual tamaño que la multiespectral.
- parametro3: Entero con el nivel, es decir, la cantidad de veces que se aplicará la transformada.
- salida: Nombre de la imagen de salida después de realizar la ejecución de la aplicación.

4 Realización de pruebas

Actualmente existe una instancia de este software en un servidor de prueba del Centro de Cómputo del Alto Desempeño de la Universidad Distrital (CECAD).

Al realizar la ejecución de la aplicación se ejecutará un comando como el siguiente

```
python          fusion_gpu.py          '/home/nvera/andres/images/rgb81r.tif'  
'/home/nvera/andres/images/pan8r.tif' 3 salidagpu
```

En este comando se pueden evidenciar los parámetros anteriormente descritos haciendo uso del script en GPU. A continuación se muestran las imágenes multiespectral (Ilustración 1) y pancromática (Ilustración 2).

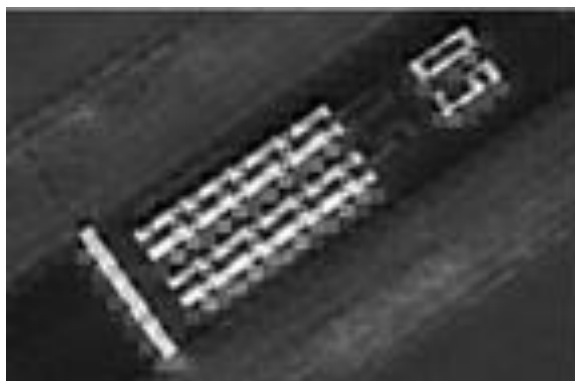


Ilustración 1. Imagen multiespectral

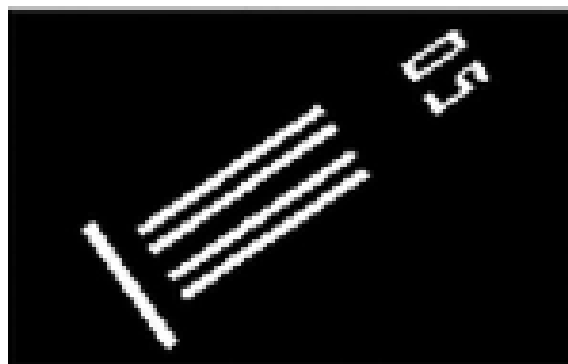


Ilustración 2. Imagen pancromática

Realizando la transformada con los 3 niveles indicados se obtiene la salida con la resolución espectral de la imagen multiespectral original y la resolución espacial de la imagen pancromática original, lo que da como resultados la siguiente imagen (Ilustración 3):

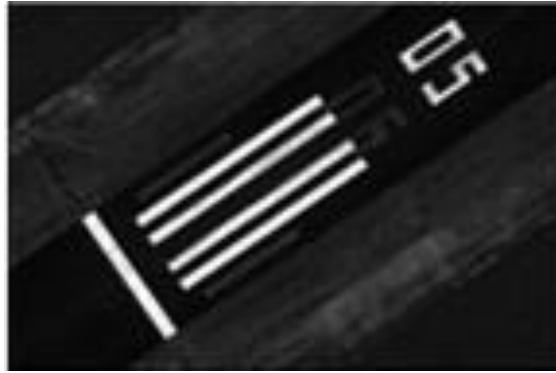


Ilustración 3. Imagen de salida

5 Resultados

Finalmente al realizar la prueba con imágenes de distintos tamaños se obtiene que al comparar la gráfica de los resultados en CPU contra los de GPU se obtiene lo siguiente:

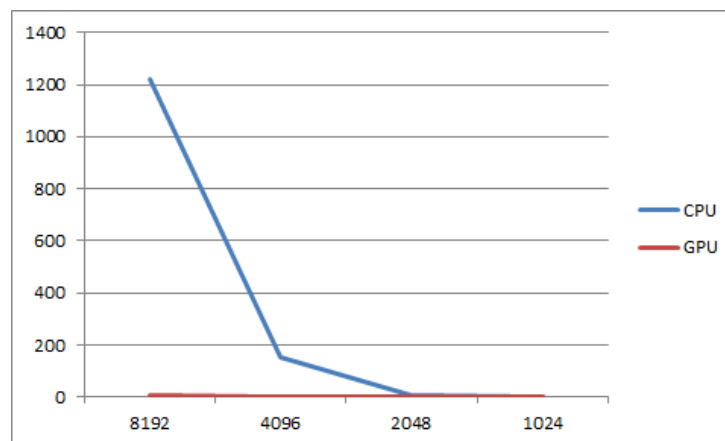


Ilustración 4. Comparación de CPU y GPU

En donde se evidencia que el tiempo de CPU y GPU aumenta a medida que aumenta el tamaño, pero el aumento de la CPU es mucho mayor que el de GPU por lo que se concluye que al realizar la implementación de este algoritmo en GPU se optimizan los tiempos de respuesta, para entender el comportamiento evidenciado en la gráfica anterior, a continuación, se muestra la tabla de datos usados:

Tabla 1. Datos comparación CPU y GPU

Tamaño	CPU	GPU
8192	1217,9416	6,7042
4096	154,1244	1,6650
2048	3,6431	0,3155
1024	0,6846	0,1043