

Paralelización en GPU de la fusión de imágenes satelitales mediante el uso de la transformada Wavelet Fast Haar

Manual de Usuario

Tabla de contenido

1	Descripción	3
2	Requisitos	4
3	Iniciar aplicación	4
4	Realización de pruebas	5
5	Medición de desempeño	6

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 hace uso de la computación heterogénea Multi-Core (CPU) / Many-Core (GPU) para acelerar el proceso de fusión de imágenes multispectrales y pancromáticas mediante el uso de la transformada Wavelet Fast Haar.

La fusión permite la combinación y utilización de datos procedentes de fuentes diferentes. La idea es obtener información de “mayor calidad” que la original, la cual dependerá de la aplicación. La fusión de imágenes es una respuesta a la frecuente necesidad de tener en una sola imagen datos de alta resolución espectral y espacial a partir de imágenes de diferente resolución espacial y/o diferentes sensores remotos. La fusión permite obtener información detallada sobre el medio ambiente urbano y rural, útil para una aplicación específica en diferentes estudios geográficos. La transformada discreta de Wavelet (TDW), es una transformación lineal que tiene una gran utilidad en el área de procesamiento de señales, para este caso señales bidimensionales. Una de sus principales aplicaciones consiste en separar conjunto de datos en componentes de distinta frecuencia espacial, representados en escalas comunes.

El algoritmo (TDW) es utilizado dentro de las estrategias de fusión de imágenes de satélite debido a la alta calidad espectral que caracteriza a las imágenes fusionadas mediante este método

La implementación de la transformada Wavelet, se realizó de la siguiente forma. La implementación de la fusión de imágenes usando la transformada Wavelet se realiza mediante la descomposición de los componentes Intensidad (I), y la imagen pancromática (PAN):

- Registrar una composición a color RGB (verdadero color) de la imagen MULT con la PAN, usando el mismo tamaño de píxel de esta última. Transformar la imagen RGB en componentes IHS (Intensidad, matiz y saturación) de la imagen MULTI.
- Aplicar el concepto de Transformada Wavelet al componente I, iterativamente hasta el segundo nivel descomposición, obteniendo de esta manera los siguientes coeficientes de aproximación y detalle. cA2i coeficientes de aproximación que contienen la información espectral de la componente I, cV2i, cH2i, cD2i, cV1i, cH1i, cD1i, coeficientes de detalle donde se almacena la información espacial de I.
- Aplicar el concepto de la Transformada Wavelet a la imagen PAN hasta el segundo nivel descomposición obteniendo de esta manera los coeficientes de aproximación y detalle. cA2p coeficientes de aproximación que contiene la información espectral de la PAN, cV2p, cH2p, cD2p, cV1p, cH1p y cD1p, coeficientes de detalle donde se almacena la información espacial de la imagen PAN.
- Generar una nueva matriz concatenando los coeficientes cA2i (que almacena la información espectral de la componente I) y los coeficientes de detalle espacial de segundo nivel de la imagen PAN, cV2p, cH2p, cD2p, cV1p, cH1p y cD1p, (que almacena la información espacial de la imagen PAN). Aplicar la transformada inversa de la Transformada Wavelet a la matriz obtenida en el paso anterior para obtener la nueva componente intensidad (N-INT).
- Generar una nueva composición IHS (N-IHS), uniendo la N-INT (nuevo componente intensidad) junto con las componentes originales de matiz y saturación (obtenidas en el primer paso). Realizar la transformación IHS a RGB, usando la nueva

composición N-IHS. De esta manera se obtiene la nueva imagen multiespectral (nueva rgb, N-MULT), que mantiene en menor valor la resolución espectral ganando así la resolución espacial.

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 paramatro1 parametro2 parametro3 salida
```

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

```
python fusion_gpu.py paramatro1 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 Medición de desempeño

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:

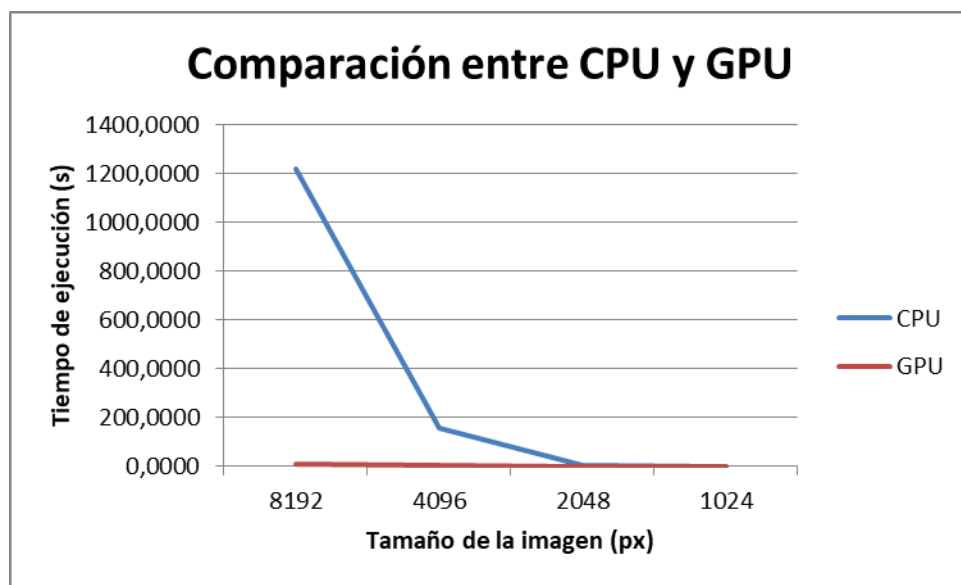


Ilustración 4. Comparación de CPU y GPU

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