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

## Manual de Instalación

### Tabla de contenido

# 1 Descripción

Esta aplicación hace uso de la computación heterogénea Multi-Core (CPU) / Many-Core (GPU) para acelerar el proceso de fusión de imágenes multiespectrales 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

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 Instalación

Para llevar a cabo la instalación, tenga en cuenta que es necesario que ya tenga instalado Python versión 2.7, de lo contrario, no será posible llevar a cabo los siguientes pasos. Cuando se instala Python, tendrá instalado PyPI (Python Package Index) por defecto, que es el que permite instalar paquetes de Python, con el comando *pip*.

## 3.1 Instalar la tarjeta gráfica

Primero se debe verificar si el sistema operativo ya reconoce la tarjeta gráfica, para eso se digita el siguiente comando y se verifica que la tarjeta gráfica NVIDIA Tesla K80 se encuentre en la lista:

```
lisci | grep -i nvidia
```

En caso de que no aparezca se digita el siguiente comando para actualizar los controladores conectados al computador

```
sudo update-pciids
```

Comprobando que la tarjeta gráfica ya se encuentra instalada se procede a realizar la instalación de los drivers por medio de los siguientes comandos:

```
sudo apt-add-repository ppa:xorg-edgers/ppa
sudo apt-add-repository ppa:ubuntu-x-swat/x-updates
sudo apt-get update
sudo apt-get install nvidia-current nvidia-settings
```

## 3.2 Instalar CUDA y PyCUDA

Para correr la aplicación es necesario instalar CUDA y PyCUDA, el primero se instala por medio de los siguientes dos comandos:

```
sudo apt-get install cuda
sudo apt-get install nvidia-cuda-toolkit
```

Con esto instalado se procede a realizar la configuración de dos variables de entorno, la primera PATH y la segunda CUDA_ROOT de la siguiente forma:

```
export PATH=/usr/local/cuda/bin:$PATH
export CUDA_ROOT=/usr/local/cuda
```

Con estas dos variables de entorno se podrá ejecutar el siguiente comando para instalar PyCUDA:

```
pip install pycuda
```

## 3.3 Clonar repositorio

Como primer paso, clone el repositorio de GitHub en su equipo.

```
git clone https://github.com/Parall-UD/Parallel-Fusimagery.git
```

## 3.4 Instalar librerías de Python

Con el comando *pip* debe instalar las librerías de python, de la siguiente forma:

```
pip install numpy
```

```
pip install pillow
```

```
pip install matplotlib
```

```
pip install pycuda
```

```
pip install scikit-cuda
```

## 3.5 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 AxA 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.