

# Librería para la fusión de imágenes satelitales sobre arquitecturas homogéneas (CPU) y heterogéneas (CPU/GPU) (Sallfus)

## Manual de Instalación

### Tabla de contenido

1	Descripción.....	2
2	Requisitos.....	2
3	Instalación .....	3
3.1	Instalar la tarjeta gráfica.....	3
3.2	Instalar CUDA y PyCUDA.....	3
3.3	Instalar dependencias de Python.....	4
3.4	Instalación de la librería Sallfus .....	4
3.5	Ejemplo .....	4

# 1 Descripción

La fusión de imágenes satelitales es un proceso digital que permite reunir en una imagen procesada la riqueza espectral de una imagen multiespectral y la resolución espacial de una imagen pancromática. Para llevar esto a cabo se cuenta con algunas transformaciones ya establecidas previamente, como lo son la Transformada de Brovey, Transformada Multiplicativa, Transformada por análisis de componentes principales y la Transformada Á Trous, entre otras.

Cada una de las distintas técnicas empleadas en la fusión de imágenes satelitales, presentan un conjunto distinto y variado de paso a seguir. No obstante, presentan dos similitudes. La primera de ellas, hace referencia a su propósito, es decir, estas técnicas buscan enriquecer espacialmente una imagen que contenga información espectral. El segundo aspecto, está relacionado con el tipo de operaciones que se realizan, puesto que, en la mayoría de casos son operaciones matriciales, donde a medida que aumenta el tamaño de dicha matriz se eleva el costo computacional.

Al implementar los algoritmos de fusión de imágenes satelitales en forma serial, es decir, realizando su ejecución exclusivamente en CPU, se presentan tiempos elevados al utilizar imágenes de dimensiones superiores, es por esto que este proyecto busca realizar la implementación de las transformadas mencionadas anteriormente mediante procesamiento heterogéneo CPU/GPU con el fin de optimizar los tiempos de ejecución para este algoritmo. Así mismo, se tiene como objetivo proporcionar herramientas para la comparación en términos de tiempos de ejecución y evaluación de la calidad de la imagen obtenida.

## 2 Requisitos

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

1. Tarjeta gráfica NVIDIA, solo si desea realizar la fusión en GPU

Librerías necesarias para ejecutar en CPU:

1. Python 3.6: <https://www.python.org/download/releases/3.6/>
2. Numpy 1.14.5: <https://pypi.org/project/numpy/>
3. Scikit-image 0.14.2: <https://pypi.org/project/scikit-image/>
4. Scipy 1.4.1: <https://pypi.org/project/scipy/>

Librerías necesarias para ejecutar en GPU:

1. Python 3.6: <https://www.python.org/download/releases/3.6/>
2. Numpy 1.14.5: <https://pypi.org/project/numpy/>
3. Pycuda 2017.1.1: <https://pypi.org/project/pycuda/>
4. Skcuda 0.5.2: <https://scikit-cuda.readthedocs.io/en/latest/install.html>
5. Cupy 7.2.0: <https://pypi.org/project/cupy/>
6. Scikit-image 0.14.2: <https://pypi.org/project/scikit-image/>

## 3 Instalación

Para llevar a cabo la instalación, tenga en cuenta que es necesario que tener instalado Python versión 3.6, 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

Esta sección, solo es necesaria realizarla, si usted desea realizar la fusión de imágenes satelitales mediante la GPU. 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 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

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
```

### 3.3 Instalar dependencias de Python

Para llevar a utilizar esta librería en una arquitectura serial haciendo uso de la CPU, se hace necesario mediante el comando *pip* instalar los siguientes módulos de python, de la siguiente forma:

```
pip install numpy == 1.14.5
```

```
pip install scikitimage == 0.14.2
```

```
pip install scipy == 1.4.1
```

Adicionalmente a esto, si el usuario desea hacer uso de la librería en una arquitectura heterogénea mediante la CPU/GPU, se hace necesario mediante el comando *pip* instalar los siguientes módulos de python, de la siguiente forma:

```
pip install pycuda == 2017.1.1
```

```
pip install scikit-cuda == 0.5.2
```

```
pip install cupy == 7.2.0
```

### 3.4 Instalación de la librería Sallfus

Esta librería se encuentra disponible en Python Package Index (PyPI) el cual es un repositorio de software para el lenguaje de programación Python. Por lo tanto, para instalar la librería se debe utilizar el siguiente comando:

```
pip install sallfus
```

Además de esto, se puede encontrar el código fuente de la librería en el siguiente repositorio de Github.

```
https://github.com/Parall-UD/sallfus
```

### 3.5 Ejemplo

Una vez instalada la librería Sallfus y sus distintas dependencias, se debe abrir la terminal de Python y ejecutar el siguiente código, con el propósito de comprobar su correcto funcionamiento.

```

>>> from sallfus.fusion import broveyCPU
>>> import skimage.io
>>> m = skimage.io.imread('multispectral.tif', plugin='tifffile')
>>> p = skimage.io.imread('panchromatic.tif', plugin='tifffile')
>>> results = broveyCPU.fusion_images(m,p)
>>> results["image"]
>>>
[[[18 29 19]
  [19 30 20]
  [42 68 46]
  ...
  [10 16 12]
  [22 35 27]
  [25 39 30]]
 [[28 55 34]
  [30 60 36]
  [28 55 34]
  ...
  [17 26 19]
  [17 26 19]
  [18 27 20]]]
>>> results["time"]
>>> 81.40903687477112

```