

Geoprocessing using GDAL library

[task-2](#)

[Saail](#) #1 October 14, 2022, 6:18am

Geoprocessing using GDAL library

The **Geospatial Data Abstraction Library (GDAL)** is a library for reading and writing raster and vector geospatial data formats.

Various geospatial software(like QGIS, ARCGIS etc.) uses GDAL library in its back-end to do geoprocessing tasks.

- Note: There is no Python specific reference documentation, but the [GDAL API Tutorial](#) includes Python examples. So we will be using python api as well as GDAL commands in python script. You can refer [here](#) for GDAL official documentation.

Some of the important commands and snippets that could be useful are:

1. For finding information of satellite image or aerial image.

```
!gdalinfo satelllite.tif
```

After running this command in your terminal you will see all the details of the georeferenced image like Coordinate system, projection system used, number of bands, size of image, corner longitude and latitude co-ordinates.

2. Changing Co-ordinate Reference System

```
!gdalwarp test.tif crs_updated.tif -t_srs "+proj=longlat  
+ellps=WGS84"
```

The above command is used to change the co-ordinate reference system and save the updated image file.

You can run this same command in python script using python subprocess module.

```
import subprocess
```

```
from_SRS = "EPSG:4326"
```

```
to_SRS = "EPSG:4326"
```

```
src='crs_updated.tif'
```

```
dest= 'updated.tif'
```

```
cmd_list = ["gdalwarp","-r", "bilinear", "-s_srs", from_SRS, "-t_sr
```

```
subprocess.run(cmd_list)
```

3. For finding latitude and longitude of every pixel

```
from osgeo import gdal

# Open tif file
ds = gdal.Open('aerial.tif')
# GDAL affine transform parameters, According to gdal documentation
xoff, a, b, yoff, d, e = ds.GetGeoTransform()

def pixel2coord(x, y):
    """Returns global coordinates from pixel x, y coords"""
    xp = a * x + b * y + xoff
    yp = d * x + e * y + yoff
    return(xp, yp)

# get columns and rows of your image from gdalinfo
rows = 36+1
cols = 34+1

if __name__ == "__main__":
    for row in range(0,rows):
        for col in range(0,cols):
            print (pixel2coord(col,row)) #longitude , #latitude
```

You can find the longitude and latitude values of all the pixels of the georeferenced satellite image using the above snippet.

4. Georeferencing image

The basic idea behind **georeferencing an image is to define the relationship between the X and Y coordinates (essentially pixels) of the image, and latitude and longitude coordinates of where those pixels**. Each of these matches is called a ground control point (GCP).

In task1B you used georeferencer plugin in QGIS for having the image georeferenced. The georeferencer plugin used GDAL library in its background to do the task. You can also generate this command from QGIS georeferencer GUI.

In QGIS after opening georeferencer GUI and matching the features of aerial and base image, Go to File>Generate GDAL Script, you will see the below command.

Command looks like:

```
gdal_translate
-gcp pixelx1 pixely1 longitude1 latitude1
-gcp pixelx2 pixely2 longitude2 latitude2
-gcp pixelx3 pixely3 longitude3 latitude3
-gcp pixelx4 pixely4 longitude4 latitude4
```

-of GTiff
map-original.jpg
map-with-gcps.tif

here pixelx1 is your x co-ordinate and pixely1 is the y co-ordinate of the image which is not georeferenced(aerial image in our case).

longitude and latitude are the geolocation from base image which is georeferenced.

Example command:

```
gdal_translate -gcp 4203.7 2347.0 4.2946 52.0825 -gcp 3830.5 1673.9  
4.2744 52.0888 -gcp 5122.7 1611.8 4.3054 52.1025 -gcp 5522.5 2981.5  
4.3371 52.0862 -gcp 3593.9 2691.5 4.2849 52.0710 -gcp 6560.4 4341.7  
4.3844 52.0761 -gcp 4432.0 4929.8 4.3406 52.0464 -gcp 486.8 2458.7  
4.2042 52.0447 -gcp 1171.9 4210.5 4.2483 52.0252 -gcp 6704.4 907.2  
4.3332 52.1289 -gcp 1880.0 1993.3 4.2314 52.0650 -of GTiff map-  
original.jpg map-with-gcps.tif
```

The above command is used to assign longitude and latitude values to the image which is not georeferenced.

You can use the subprocess python module to run this command in python script.

Once georeferncing is done the image will not have the proper co-ordinate system. You need to change the co-ordinate reference system using the step 2.

References:

- [Georeferencing and digitizing old maps with GDAL](#)
- [Georeferencing Satellite Images](#) | [TileMill](#)

[Task 2D: Finding geolocation of the suspicious objects using drone](#)

[Smit](#) closed #2 October 18, 2022, 8:41am

[Smit](#) unlisted #3 October 18, 2022, 8:41am