Im2DSM

Version 1.8, 04.11.2022

Projects single channel raster images to DSM using 3 Ground Control Points.

Input:

- DSM
- Image/s (.asc, .tif) only single channel
- Camera Location
- Camera Parameters (focal length, sensor size or FOV)
- Ground Control Points or precalculated pitch yaw, roll angles

Usage

Two files:

- Im2DSM.py
- Im2DSM_Parameters.txt (same directory as Im2DSM.py)

Im2DSM_Parameters - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
#############################################################################################################
### Input Parameters for Im2DSM.py
### Processing a single channel .asc Image with given Camera Parameters to a DSM
###
### this file has to be within the same directory as Im2DSM.py
#############################################################################################################
#
### Working directory (if not location of .py file)
IN_DIRECTORY     =       C:/Users/beneh/Desktop/DataThermo/ExGreen/wannenkogel-2021-10-27
OUT_DIRECTORY    =       C:/Users/beneh/Desktop/DataThermo/MonoTest
FILE_FORMAT      =       .tif
#
### filenames of DSM and Viewshed (has to be format .tif)
DSM_INPUT        =       C:/Users/beneh/Desktop/DataThermo/DOM_small/small-DOM_05_25832.tif
VIEWSHED         =       C:/Users/beneh/Desktop/DataThermo/DOM_small/viewshed_webcam_wannenkogel_05_small.tif
#
### if format is .asc define rows to skip (Thermos from VarioCam ASC_SKIP = 17)
ASC_SKIP         =       0
#
### Camera location coordinates and FOV in degrees (fov in degree = (angle horizontal, angle vertical))
CAMLOCATION_X    =       657549
CAMLOCATION_Y    =       5209670
CAMLOCATION_Z    =       2645
FOCAL_LENGTH     =       22.0
SENSOR_X         =       22.8
SENSOR_Y         =       14.5
FOV_SENSOR_X     =       NA
FOV_SENSOR_Y     =       NA
#
### Minimum angle between 3 GCPs to avoid near colinearity
CO_LINEAR        =       30
#
### Adjustments for Distortion
DISTORTION       =       -0.08
DIST_ZERO        =       1.3
DIST_MODE        =       SIN_SQRT
#
### Adjustements for Euler angles
ADJUST_LEFT      =       0
ADJUST_UP        =       0.1
ADJUST_ROLL      =       -0.6
#
### If Camera has INS (set to NA if not)
ANGLE_UP         =       NA
ANGLE_LEFT       =       NA
ANGLE_ROT        =       NA
#
### Reference points for camera orientation (Set to NA if INS is available)
GCP_FILE         =       C:/Users/beneh/Desktop/DataThermo/GCP/GCP_Wannenkogel.csv
ACCURACY         =       YES
```

Required Python modules:

os, matplotlib.pyplot, numpy, pandas, math, osgeo, rastrio, warnings

Adjustment Parameter file

IN_DIRECTORY

- Full path to unprojected images
- Only .tif or .asc and single channel are supported

OUT_DIRECTORY

Full path of directory the projected images should be stored in.

new folder will be created containing:

- Projected images
- Viewshed (optional)
- Accuracy.csv (containing accuracy values for GCPs, if ACCURACY != NA)

FILE_FORMAT

.asc or .tif

DSM_INPUT

Full path of DSM/DEM to use for projection (hast o be .tif)

VIEWSHED

- Full path to viewshed from camera location (hast o be .tif)
- If not available set line to NA, viewshed will then be calculated from given DSM via gdal viewshed. Probably you have to change path to gdal_viewshed.exe in line 172 of the .py script.
- BUT: in this case camera location has to be within DSMs boundaries (big DSM → monoplotting takes longer; better to precalculate viewshed and then clip viewshed and DSM to area of interest)

ASC_SKIP

- If unprojected images are .asc then choose line where to start reading textfile (e.g. IRBIS thermograms ASC_SKIP = 17)

CAMLOCATION_...

- XYZ Coordinates of camera location (same system as DSM)

FOCAL_LENGTH and SENSOR_...

- Camera settings of focal length and sensor width (X) and height (Y)
- Set to NA if FOV is given

FOV_SENSOR_...

- Camera field of view in X and Y direction
- Set to NA if only focal length and sensor are given

CO_LINEAR

- (has nothing to do with this script…)
- default 0

DIST_MODE

- Set to NA, if images are undistorted with matching GCPs
- Two options:
    - SIN: for a sinus curve estimation (values around 0.1)
    - SIN_SQRT: for a sinus squareroot estimation (values around 0.01)
- ! dont use if not really necessary

DISTORTION

- Set to 0, if images are undistorted witch matching GCPs
- Values depend on DIST_MODE

DIST_ZERO

- Defines radius where distortion is 0 again
- Value is: (pic_Y/2)/DIST_ZERO
- e.g. DIST_ZERO = 1 for a pic 3000x2000px → the radius around the image center will be 1000px

ADJUST_LEFT

- manual adjustement of yaw angle
- e.g. 2 means 2° counterclockwise, -2 means 2° clockwise

ADJUST_UP

- manual adjustment of pitch angle
- e.g. 2 means 2° up, -2 means 2° down

ADJUST_ROLL

- manual adjustment of roll angle
- e.g. 2 means 2° counterclockwise, -2 means 2° clockwise

ANGLE_...

- if yaw, pitch and roll are given beforehand
- set to NA if calculation from GCP is necessary

GCP_FILE

- file hast o be .csv containing: Index, name of GCP, Raster X, Raster Y, Raster Z, image X, image Y

1	,Name,X,Y,Z,pic_X,pic_Y
2	0,GCP1,658735.665,5210652.289,2499.709,2201,760
3	1,GCP2,658630.39,5210703.37,2471.507,1980,705
4	2,GCP3,658554.914,5210679.572,2451.726,1929,618
5	3,GCP7,658632.064,5210834.537,2490.55,1806,757
6	4,GCP8,658563.794,5210773.111,2441.523,1787,642
7	5,GCP9,658453.721,5210873.012,2453.858,1497,671
8	6,GCP11,658220.269,5210958.453,2398.088,1007,538
9	7,GCP16,659290.351,5210917.725,2691.096,2428,1133
10	8,GCP19,658332.146,5210716.104,2357.018,1504,396
11	9,GCP_20,658896.104,5211178.114,2716.692,1755,1168
12	10,GCP_22,658673.665,5211202.623,2703.755,1466,1156

- Image X=0 and Image Y=0 is the lower left corner of the image (NOT numpy!)

ACCURACY

- Set to NA if accuracy assessment of GCPs is not necessary
- NA recommended for fast calculation
- Set to YES (or any other) for accuracy assessment

WORKFLOW

Calculation FOV

$$FOV_x = 2 * \tan^{-1} \frac{0.5 * sensor_x}{focal}$$

Pitch, yaw and roll from GCPs

Vector plane in real world is calculated to determine pitch and yaw.

Processing coordinates to DSM

An img_x, img_y coordinate is assigned to every DSM raster cell based on FOV, image size and pitch, yaw, roll.

$$raster\_x_{i,j} = ((\propto_{center} + FOV_x) - \propto_{i,j}) * \frac{size_x}{FOV_x}$$

Raster_x$_{i,j}$ ist he corrected with roll angle.

Masking with viewshed

Simple viewshed

Projecting images

Assigning image values to raster based on *Raster_x*$_{i,j}$ and *Raster_y*$_{i,j}$.