simactive
HIGH-END PHOTOGRAMMETRY SOFTWARE

QUICKSTART GUIDE

Running Correlator3D Via Python Scripts

1 OVERVIEW

Correlator3D (C3D) has been developed to run in both the graphical user interface (GUI) and in command-line mode. The latter gives the user the option to create scripts that can be executed via batch files, through the Python programming language. This method only requires a minimal amount of programming, along with the associated files to execute a processing workflow.

This guide will describe the process of creating a Python script along with the files required to run a project in Correlator3D.

GET IN TOUCH
www.simactive.com
contact@simactive.com

2 PROJECT SETUP

In addition to C3D, users will need to install Python which can be downloaded from the following address:

https://www.python.org/

Prior to processing, an exterior orientation (EO) file containing the image names and corresponding coordinates (X, Y, Z and possibly omega, phi, kappa angles) is required. If the EO data are stored within the EXIF of the image, an extraction tool such as ExifTool can be used:

https://exiftool.org/

Additionally, a directory structure will be needed to store the input data.

In the following example, the name used will be:

E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES

3 EXIF EXTRACTION

If EO information is available as a TXT or CSV file, no extraction is needed.

If it is stored in the images, the following command line can be used to extract EXIF coordinates from the images:

Exiftool -T -filename -gpslongitude -gpslatitude -gpsaltitude -n

E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\IMAGES\*.JPG

E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\EO.csv

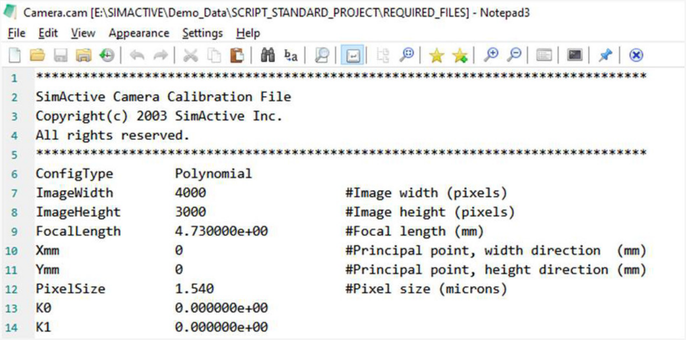
The converted EO file should then be placed in the REQUIRED_FILES folder. Two additional files to be included in that folder prior to processing are:

- ✓ Camera calibration file
- ✓ C3D Script.spt file

Each can be generated as per the instructions described in the following sections.

4 CAMERA CALIBRATION FILE

The following figure shows a standard calibration file used by C3D. The values should be modified to match the camera used for a specific project.


Camera.cam [E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES] - Notepad3
File
Edit
View
Appearance
Settings
Help

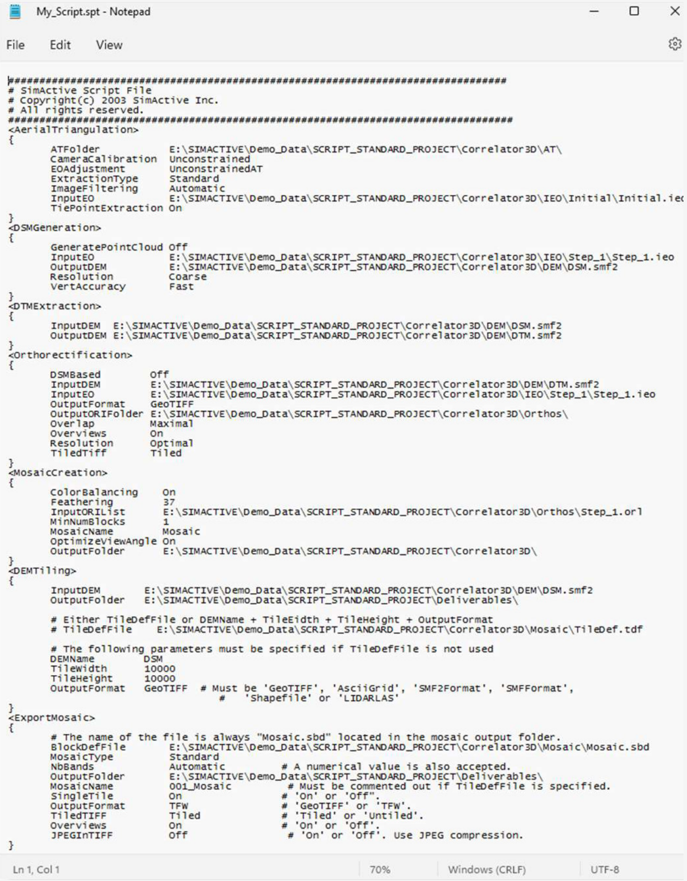
```
**************************************************************************
SimActive Camera Calibration File
Copyright(c) 2003 SimActive Inc.
All rights reserved.
**************************************************************************
ConfigType      Polynomial
ImageWidth      4000                    #Image width (pixels)
ImageHeight     3000                    #Image height (pixels)
FocalLength     4.730000e+00            #Focal length (mm)
Xmm             0                       #Principal point, width direction  (mm)
Ymm             0                       #Principal point, height direction (mm)
PixelSize       1.540                   #Pixel size (microns)
K0              0.000000e+00
K1              0.000000e+00
```

5 C3D SCRIPT FILE

An example of a script file, which can be modified to fit specific processing requirements, can be found in the following directory:

\Program Files\SimActive\Correlator3D\Sample Files\Script Folder

The C3D_Script.spt file includes all the processing options available in C3D. Users can modify a copy of the file and change values to match processing needs. In the following example, it is called My_Script.spt.

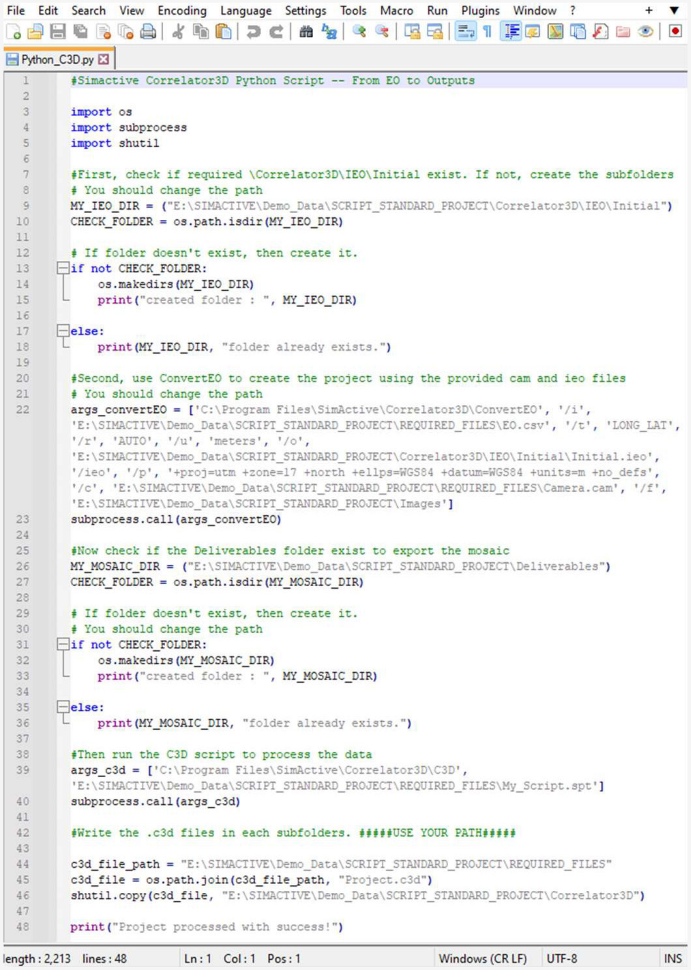


```
################################################################################
# SimActive Script File
# Copyright(c) 2003 SimActive Inc.
# All rights reserved.
################################################################################
<AerialTriangulation>
{
      ATFolder           E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\AT\
      CameraCalibration  Unconstrained
      EOAdjustment       UnconstrainedAT
      ExtractionType     Standard
      ImageFiltering     Automatic
      InputEO            E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\IEO\Initial\Initial.ie
      TiePointExtraction On
}
<DSMGeneration>
{
      GeneratePointCloud Off
      InputEO            E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\IEO\Step_1\Step_1.ieo
      OutputDEM          E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\DEM\DSM.smf2
      Resolution         Coarse
      VertAccuracy       Fast
}
<DTMExtraction>
{
      InputDEM  E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\DEM\DSM.smf2
      OutputDEM E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\DEM\DTM.smf2
}
<Orthorectification>
{
      DSMBased        Off
      InputDEM        E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\DEM\DTM.smf2
      InputEO         E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\IEO\Step_1\Step_1.ieo
      OutputFormat    GeoTIFF
      OutputORIFolder E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\Orthos\
      Overlap         Maximal
      Overviews       On
      Resolution      Optimal
      TiledTiff       Tiled
}
<MosaicCreation>
{
      ColorBalancing    On
      Feathering        37
      InputORIList      E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\Orthos\Step_1.orl
      MinNumBlocks      1
      MosaicName        Mosaic
      OptimizeViewAngle On
      OutputFolder      E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\
}
<DEMTiling>
{
      InputDEM       E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\DEM\DSM.smf2
      OutputFolder   E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Deliverables\

      # Either TileDefFile or DEMName + TileEidth + TileHeight + OutputFormat
      # TileDefFile    E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\Mosaic\TileDef.tdf

      # The following parameters must be specified if TileDefFile is not used
      DEMName        DSM
      TileWidth      10000
      TileHeight     10000
      OutputFormat   GeoTIFF  # Must be 'GeoTIFF', 'AsciiGrid', 'SMF2Format', 'SMFFormat',
                              #   'Shapefile' or 'LIDARLAS'
}
<ExportMosaic>
{
      # The name of the file is always "Mosaic.sbd" located in the mosaic output folder.
      BlockDefFile       E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\Mosaic\Mosaic.sbd
      MosaicType         Standard
      NbBands            Automatic          # A numerical value is also accepted.
      OutputFolder       E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Deliverables\
      MosaicName         001_Mosaic          # Must be commented out if TileDefFile is specified.
      SingleTile         On                 # 'On' or 'Off".
      OutputFormat       TFW                # 'GeoTIFF' or 'TFW'.
      TiledTIFF          Tiled              # 'Tiled' or 'Untiled'.
      Overviews          On                 # 'On' or 'Off'.
      JPEGInTIFF         Off                 # 'On' or 'Off'. Use JPEG compression.
}
```

6 CREATING THE PYTHON SCRIPT

Once the extracted EO, the camera calibration file and the C3D_Script.spt files have been created and placed in the REQUIRED_FILES folder, a Python script can be produced. The following figure shows a sample script file (Python_C3D.py) that is designed to complete the following steps:

1. Check if the required folders exist and if not, create them.
2. Use ConvertEO (a program included with a C3D installation) to create the project using the provided camera and EO files.
3. Check to see if the Deliverables folder exists allowing the export of the mosaic. If not, create this folder.
4. Run the C3D script file (My_Script.spt).
5. Write the C3D files into the project folder.


File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? + ▼
Python_C3D.py
length : 2,213 lines : 48
Ln : 1 Col : 1 Pos : 1
Windows (CR LF)
UTF-8
INS

```
#Simactive Correlator3D Python Script -- From EO to Outputs

import os
import subprocess
import shutil

#First, check if required \Correlator3D\IEO\Initial exist. If not, create the subfolders
# You should change the path
MY_IEO_DIR = ("E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\IEO\Initial")
CHECK_FOLDER = os.path.isdir(MY_IEO_DIR)

# If folder doesn't exist, then create it.
if not CHECK_FOLDER:
    os.makedirs(MY_IEO_DIR)
    print("created folder : ", MY_IEO_DIR)

else:
    print(MY_IEO_DIR, "folder already exists.")

#Second, use ConvertEO to create the project using the provided cam and ieo files
# You should change the path
args_convertEO = ['C:\Program Files\SimActive\Correlator3D\ConvertEO', '/i',
'E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES\EO.csv', '/t', 'LONG_LAT',
'/r', 'AUTO', '/u', 'meters', '/o',
'E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D\IEO\Initial\Initial.ieo',
'/ieo', '/p', '+proj=utm +zone=17 +north +ellps=WGS84 +datum=WGS84 +units=m +no_defs',
'/c', 'E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES\Camera.cam', '/f',
'E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Images']
subprocess.call(args_convertEO)

#Now check if the Deliverables folder exist to export the mosaic
MY_MOSAIC_DIR = ("E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Deliverables")
CHECK_FOLDER = os.path.isdir(MY_MOSAIC_DIR)

# If folder doesn't exist, then create it.
# You should change the path
if not CHECK_FOLDER:
    os.makedirs(MY_MOSAIC_DIR)
    print("created folder : ", MY_MOSAIC_DIR)

else:
    print(MY_MOSAIC_DIR, "folder already exists.")

#Then run the C3D script to process the data
args_c3d = ['C:\Program Files\SimActive\Correlator3D\C3D',
'E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES\My_Script.spt']
subprocess.call(args_c3d)

#Write the .c3d files in each subfolders. #####USE YOUR PATH#####

c3d_file_path = "E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\REQUIRED_FILES"
c3d_file = os.path.join(c3d_file_path, "Project.c3d")
shutil.copy(c3d_file, "E:\SIMACTIVE\Demo_Data\SCRIPT_STANDARD_PROJECT\Correlator3D")

print("Project processed with success!")
```

7 PROCESSING

From the command prompt, the newly created Python script (Python_C3D.py) can be simply executed. Once the Script is completed, it will prompt a "Project processed with success" message.

Upon completion, a C3D project file (.c3d format) will have been created, which allows the visualization of the results within the GUI. The template .c3d file located in the REQUIRED_FILES folder can be simply copied andpasted in the Correlator3D folder where the project has been created. Then the project file can be opened in C3D.