

TouchTerrain standalone installation

([link to google doc](#)) Last edited: Aug. 14, 2017

*(Aug. 11, 2017): The JSON file now has a **CPU_cores_to_use** parameter. Default value is 0, which means use **all** available CPU cores. This can greatly speed up processing of multiple tiles! Multi-core seems to work for me on Mac, Windows 10 and Linux but it's not well tested. If you run into problems set "CPU_cores_to_use": 1, to force it to use only a single processor.*

Install Python and modules

- Install Python 2.7 (official site: <https://www.python.org/>) (should probably use the 64 bit version)
- Make sure python works in a terminal (e.g.in the Windows Power Shell)
- Make sure you can run pip in the command line. Linux users can install pip with: `sudo apt install python-pip`, or see <https://pip.pypa.io/en/stable/installing/>
- Install **numpy** (`pip install numpy`)
- Install **pillow** (`pip install pillow`), package will show up as PIL after install
- Install **vectors** (`pip install vectors`)
- (Alternatively you can use anaconda or miniconda, which uses conda in lieu of pip)

Install Earth Engine (ee) and its dependencies

- Below is my workflow for installing it on my Windows 10 PC, it's based on the official Earth Engine site here: https://developers.google.com/earth-engine/python_install
- Alternatively, have a look at: <https://github.com/earthlab/tutorials/blob/master/documentation/intro-google-earth-engine-python-api.md> which describes setting up earth engine as well.
- Install the google-api-python-client package and it's dependencies:
`pip install google-api-python-client`

```

PS C:\Users\charding> pip -V
pip 9.0.1 from c:\python27\lib\site-packages (python 2.7)
PS C:\Users\charding> pip install google-api-python-client
Collecting google-api-python-client
  Downloading google_api_python_client-1.6.1-py2.py3-none-any.whl (52kB)
    100% |#####| 61kB 245kB/s
Collecting uritemplate<4dev,>=3.0.0 (from google-api-python-client)
  Downloading uritemplate-3.0.0-py2.py3-none-any.whl
Collecting oauth2client<5.0.0dev,>=1.5.0 (from google-api-python-client)
  Downloading oauth2client-4.0.0-py2.py3-none-any.whl (184kB)
    100% |#####| 194kB 2.2MB/s
Collecting httpplib2<1dev,>=0.9.2 (from google-api-python-client)
  Downloading httpplib2-0.9.2.zip (210kB)
    100% |#####| 215kB 1.7MB/s
Collecting six<2dev,>=1.6.1 (from google-api-python-client)
  Downloading six-1.10.0-py2.py3-none-any.whl
Collecting pyasn1-modules>=0.0.5 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client)
  Downloading pyasn1_modules-0.0.8-py2.py3-none-any.whl
Collecting pyasn1>=0.1.7 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client)
  Using cached pyasn1-0.1.9-py2.py3-none-any.whl
Collecting rsa>=3.1.4 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client)
  Downloading rsa-3.4.2-py2.py3-none-any.whl (46kB)
    100% |#####| 51kB 1.7MB/s
Installing collected packages: uritemplate, pyasn1, pyasn1-modules, rsa, six, httpplib2, oauth2client, google-api-python-client
Running setup.py install for httpplib2 ... done
Successfully installed google-api-python-client-1.6.1 httpplib2-0.9.2 oauth2client-4.0.0 pyasn1-0.1.9 pyasn1-modules-0.0.8 rsa-3.4.2 six-1.10.0 uritemplate-3.0.0
PS C:\Users\charding>

```

- If **no error** is returned by the following command, you can skip the next step.
python -c "from oauth2client import crypt"
- If there's an **error**, you'll need to install PyCrypto. Windows Python 2.7 binary are here: <http://www.voidspace.org.uk/python/modules.shtml#pycrypto> (32 or 64 bit? Depends on which Python you installed, probably 64 bit ...) On Linux you can run:
sudo apt install libssl-dev openssl to install the required cryptographic libraries
- pyOpenSSL is a Python wrapper for the OpenSSL library, it can be installed from the Python Package Index by running the following command:
pip install 'pyOpenSSL>=0.11'
- Install the Google Earth Engine python module (ee):
pip install earthengine-api

- pip list should now show you these modules as installed:

```
cffi (1.9.1)
cryptography (1.7.1)
earthengine-api (0.1.102)
enum34 (1.1.6)
google-api-python-client (1.6.1)
httplib2 (0.9.2)
idna (2.2)
ipaddress (1.0.18)
numpy (1.12.0)
oauth2client (4.0.0)
olefile (0.44)
Pillow (4.0.0)
pip (9.0.1)
pyasn1 (0.1.9)
pyasn1-modules (0.0.8)
pyparser (2.17)
pycrypto (2.6)
pyOpenSSL (16.2.0)
rsa (3.4.2)
setuptools (28.8.0)
six (1.10.0)
uritemplate (3.0.0)
```

- Set Up your Authentication Credential

```
python -c "import ee; ee.Initialize()"
```

and you'll get an error that describes what you have to do to get your authentication code:

```
PS C:\Users\charding\.config> python -c "import ee; ee.Initialize()"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "C:\Python27\lib\site-packages\ee\__init__.py", line 90, in Initialize
    credentials = _GetPersistentCredentials()
  File "C:\Python27\lib\site-packages\ee\helpers.py", line 41, in _GetPersistentCredentials
    raise EEException('Please authorize access to your Earth Engine account ')
ee.ee_exception.EEException: Please authorize access to your Earth Engine account by running
earthengine authenticate
in your command line, and then retry.
```

- Type:

```
earthengine authenticate
```

This will have you log into your google account in a browser and then show you your authentication code (a very long string). Copy/paste it into the terminal and hit return:

```
PS C:\Users\charding\.config> earthengine authenticate

Opening web browser to address https://accounts.google.com/o/oauth2/auth?
th%2Fearthengine+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdevstorage.read_w
2.0%3Aaob&response_type=code&client_id=517222506229-vsmmajv00u10bs7p89v5m89qs
Please authorize access to your Earth Engine account, and paste
the resulting code below.
If the web browser does not start, please manually browse the URL above.

Please enter authorization code: DYSUIUDIUBSYDIUYEB^&DBSBDGSGB
```

- If your pasted authentication code was correct, you'll get:

Successfully saved authorization token.

This means a text file called **credentials** (containing that code) was create on your system (for me in my home folder, in `.config\earthengine\`):

```
Directory: C:\Users\charding\.config\earthengine

Mode                LastWriteTime         Length Name
----                -
-a-----          1/18/2017   4:15 PM           66 credentials
```

- Update (Feb.7): `earthengine authenticate` may not work. If you get an
- `EEException` with a 404 html, sign up for a Earth Engine account. The [Earth Engine FAQ](#) (How do I get access?) tells you how to request access.

Test Earth Engine

- Now you can test the ee python API. Open a python interactive shell and type this, which should print a dictionary with metadata about the SRTM90 layer:

```
import ee
ee.Initialize()
image = ee.Image('srtm90_v4')
print(image.getInfo())
```

```

Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32
Type "help", "copyright", "credits" or "license" for more information.
>>> import ee
>>> ee.Initialize()
>>> image = ee.Image('srtm90_v4')
>>> print image.getInfo()
{'bands': [{u'crs': u'EPSG:4326', u'crs_transform': [0.0008333333333333, 0
id': u'elevation', u'data_type': {u'max': 32767, u'type': u'PixelType', u'
ons': [432000, 144000]}}, u'version': 1463778555689000L, u'type': u'Image'
m:time_end': 951177600000L, u'system:asset_size': 18827626666L, u'system:t
>>>

```

Create STL terrain models

- To create your terrain model(s), edit the example_config.json file and save it with another name. My file is called test.json, it's pretty much the example_config.json file except it has 2 x 2 tiles and more z scaling:

```

{
  "DEM_name": "USGS/NED",
  "basethick": 1,
  "bllat": 44.50185267072875,
  "bllon": -108.25427910156247,
  "fileformat": "STLb",
  "ntilesx": 2,
  "ntilesy": 2,
  "printres": 0.5,
  "tile_centered": true,
  "tilewidth": 80,
  "trlat": 44.69741706507476,
  "trlon": -107.97962089843747,
  "zip_file_name": "terrain",
  "zscale": 3.0
}

```

- Run the standalone python script with your edited config json file as argument:
python TouchTerrain_standalone.py <your_config_file.json>

```

PS C:\Users\charding\TouchTerrain_for_CAGEO-master> ls *.json

Directory: C:\Users\charding\TouchTerrain_for_CAGEO-master

Mode                LastWriteTime         Length Name
----                -
-a----             1/18/2017   4:55 PM          336 example_config.json
-a----             1/18/2017   4:58 PM          338 test.json

PS C:\Users\charding\TouchTerrain_for_CAGEO-master> python TouchTerrain_standalone.py test.json
wrote example_config.json with default value, use it as a template but make sure to rename it!
reading test.json
bllon = -108.254279102
printres = 0.5
tile_centered = True
ntilesx = 2
ntilesy = 2
bllat = 44.5018526707
basethick = 1
zip_file_name = terrain
trlat = 44.6974170651
zscale = 3.0
DEM_name = USGS/NED
tilewidth = 80
trlon = -107.979620898
fileformat = STlb
tile  1 1 MainThread
10 % MainThread
20 % MainThread
30 % MainThread
40 % MainThread

```

- This will print out the progress % as it creates the tiles and saves them as a zip file:

```

90 % MainThread
1 2 done! MainThread
tile  2 1 MainThread
10 % MainThread
20 % MainThread
30 % MainThread
40 % MainThread
50 % MainThread
60 % MainThread
70 % MainThread
80 % MainThread
90 % MainThread
2 1 done! MainThread
tile  2 2 MainThread
10 % MainThread
20 % MainThread
30 % MainThread
40 % MainThread
50 % MainThread
60 % MainThread
70 % MainThread
80 % MainThread
90 % MainThread
2 2 done! MainThread
done
finished writing terrain_2017_01_18_17_00_42.zip

```

- The zip file contains the STL files for each tile and a info text file:

 NED_-108.12_44.60_log.txt	Text Document	1 KB
 NED_-108.12_44.60_tile_1_1.STL	Meshmixer Document	965 KB
 NED_-108.12_44.60_tile_1_2.STL	Meshmixer Document	952 KB
 NED_-108.12_44.60_tile_2_1.STL	Meshmixer Document	966 KB
 NED_-108.12_44.60_tile_2_2.STL	Meshmixer Document	961 KB

```
NED_-108.12_44.60_log.txt - Notepad
File Edit Format View Help
Log for creating 3D model tile(s) for NED_-108.12_44.60
DEM_name = USGS/NED
trlat = 44.6974170651
trlon = -107.979620898
bllat = 44.5018526707
bllon = -108.254279102
printres = 0.5
ntilesx = 2
ntilesy = 2
tilewidth = 80
basethick = 1
zscale = 3.0
fileformat = STLb
process started: 17:00:42 221000
```