HRSC data processing pipeline documentation

David Trethewey

December 16, 2016

Abstract

This is an attempt to document the data processing of the 179 High Resolution and Stereo Camera fields used in my 2014 MSc dissertation. The scripts processing the data were written on a somewhat ad-hoc basis at the time. To some extent the processing was documented in the dissertation itself, but this document will cover it at a more low level, as well as work done since the dissertation, such as the Top Trumps webpages. The scripts themselves are not very well written, and often do things like hard-code file paths in an idiosyncratic way.

1 Top-level summary of steps.

This is a list of the steps involved.

1. Download the Colin Souness 2012 paper, with the supplementary data, including a list of the locations of the 1309 entries in his catalogue of Martian glacier-like forms.

2. Identify the HRSC coverage for the locations of the glaciers. This was done during the dissertation manually, using the web interface in the Mars Orbital Data Explorer, to attempt to identify the HRSC tile that gave the best coverage for each Souness glacier. Later on, I made an automated check, using the HRSC DTM coverage shapefile available, along with shapefiles I had generated based on the Souness catalogue. This was using the OGR Python API.

3. Reproject the ND4 nadir image, and the DA4 areoid elevation to an equirectangular coordinate system, using 40°as the reference latitude. Done using the `gdalwarp` command line program, via scripting in Python (the `os.system` library). Resample a copy of the ND4 image to the resolution of the DTM.

4. Use LandSerf to create the derived topographic variables. Initially GDAL itself was used, but LandSerf was thought to offer more as regards the curvature layers. Used Landscript, which was generated by Python.

5. A 10 band layerstack including all the curvature layers was created for each tile, including LandSerf's feature classification (1500m window), however in the further analysis a 6 band stack was used, including nadir, dtm, slope, aspect (from N), longitudinal curvature, and cross-sectional curvature. This was partly due to processing time, but also because including all of the curvature layers is duplicating the same information. Two different versions were used, using the absolute aspect from N, and the raw aspect (which has a discontinuity at 0/360°. The absolute aspect was used for the segmentation, but the raw for zonal stats (check detail of this).

6. Add 9999 to make the no data value zero, and avoid negative values of feature layers, which can cause problems with RSGISlib. Done with gdal_calc.

7. RSGISlib segmentation done - run via Python script for all of the tiles.

8. Populate stats to the raster attribute table of the segment clumps output of RSGISlib segmentation.

9. Export the RAT to a text .csv file.

10. Convert to shapefile with gdal_polygonize. Join stats to output shapefile. (check detail of this).

11. Zonal stats for each fields. Heads and extent areas of Souness glaciers.

12. Collect stats together for all fields.

13. Compare to overall distribution of variables in the HRSC tiles.

14. Gaussian fits to histograms, get functions of glacier-like-form abundance for each feature variable.

15. Create classifier functions.

16. Visualise in QGIS and do some basic statistical tests.

17. Subset the ND4 and DA4 images for Top Trumps. Colour the DA4 output with a pseudocolour scheme.

18. Generate JSON for Top Trumps.

19. Javascript for Top Trumps webpage.

2 Downloading data

Data is obtained from Mars Orbital Data Explorer, using Data Product Search, selecting under the Mars Express heading 'DTMRDR - Digital Terrain Map Reduced Data Record'. Originally when doing this manually, find footprints with 'Find by Location or Feature', specifying the latitude and logitude range. When filling in a few gaps where better coverage of a Souness GLF was found, I already know the product ID, so used 'Filter by Product ID'.

Use the panchromatic nadir image and the areoid digital terrain model. These are the Product IDs that end in 'ND4.IMG' and 'DA4.IMG' respectively.

Each of the tiles was put in a series of directories, which were named nnnn for the digits in the product ID. These were the only directories immediately under the top-level of the processing directory. Many of the scripts processing the data lived in this same directory, which is currently `/media/davydh/TOSHIBA EXT/ioSafeBackup/RemoteSensingPlanSci_MSc/SounessCatalog3_b`

3 Reprojection and resampling

The panchromatic nadir image and the areoid digital terrain model were reprojected using `gdalwarp` to a common system, which is an equirectangular projection, at a standard latitude of 40 °. They are also converted from `.img` format (ERDAS imagine format) to `.kea`. A copy of the nadir image is created resampled to the DTM resolution, which is discovered via `gdalinfo`.

A script entitled `python_gdalwarp_reprojMars_eqcyl_midlat_all.py` did this for all tiles under the top-level data directory, by calling the following commands.

```
  gdalwarp_cmdbase = "gdalwarp -srcnodata -32768 -dstnodata -9999 -ot Int16 -t_s
gdalwarp_cmdbase_nd4 = "gdalwarp -srcnodata 0 -dstnodata -9999 -ot Int16 -t_srs
targetres = ' -tr '+str(tres)+' '+str(tres*-1)
gdalwarp_cmdnd4 = gdalwarp_cmdbase_nd4 + ' ' + inputfilename_nd4 + ' ' + outputf
gdalwarp_cmdnd4_resam = gdalwarp_cmdbase_nd4 + targetres + ' ' + inputfilename_n
gdalwarp_cmdda4 = gdalwarp_cmdbase + targetres + ' ' + inputfilename_da4 + ' ' +
```

The variable `tres` is the resolution of the DTM, which was read via `gdalinfo`.

4 Creating derived topographic layers

The derived topographic layers are created in LandSerf. This was done by running a script for each individual tile, in the Landscript language used in the program. This can however be generated for each tile using Python. The scripts are currently in a directory in my home drive `/ioSafeBackup/RemoteSensingPlanSci_MS` The DTM is converted from .kea to .asc using `gdal_translate` for processing in LandSerf, using the following command `gdaltranscmd = "gdal_translate -of AAIGrid "+DTMinFile+" "+ascOutFile` . The following layers were generated:

- Slope
- Aspect
- Profile Curvature
- Plan Curvature
- Cross-sectional Curvature
- Longitudinal Curvature
- Mean Curvature
- LandSerf features (with a 1500m window)

The layers are described in the PhD thesis of Jo Wood. The aspect layer is later converted to absolute aspect from north. The main processing was done with a 6 band layerstack, which had

- Nadir image resampled to DTM resolution
- DTM elevation
- slope
- aspect from N (or the original aspect - these were created as 2 variants)
- Cross-sectional curvature
- Longitudinal curvature.

`gdal_calc` is used to calculate the absolute aspect from north, and `gdal_merge` to create a layerstack.

5 Layerstacking

Files ending in `LandSerfLayerstack2.kea` are 10 band layerstacks, with absolute aspect from N (generated with `gdalmerge_Landserf2_python3.py`), files ending in `LandSerfLayerstack3` are 6 band layerstacks with absolute aspect from N (generated with `gdalmerge_Landserf3_python3.py`), files with `LandSerfLayerstack_rawAsp.kea` are 6 band layerstacks with the raw aspect (generated with `gdalmerge_Landserf_rawAspect_python3.py`). This last script also generates a version with 9999 added, so that 0 becomes the no data value, and the values for all features are positive.

6 RSGISlib segmentation

Segmentation was done using `RSGISLib`, as described in http://rsgislib.org/rsgislib_segmentation.h
The script `segmentation_mars_version008b_allLandSerf.py` is the version used in the dissertation. This works with the 6 band layerstacks, with absolute aspect from north, and 9999 added. The script `segmentation_mars_version010_rawasp_Lan`
uses the 6 band raw aspect version of the layerstacks, with 9999 added. A minimum object size of 0.2 km^2 was used in the dissertation.

7 Populating segment stats

This was done with the `rastergis` module of `rsgislib`. The main script was
`PopulateStats_HRSC_Mars2000EqCyl_lat0_40_2e5sqm_only_add9999_allLandSerfLayerstack`
This populates the raster attribute table of the segment clumps file with stats of each band. The minimum, maximium, mean and standard deviation in each band is calculated within each segment. The stats are populated based on the layerstack before 9999 was added.

8 Convert segments to shapefile, and zonal stats

This was probably not the best way to do this. What I did in the dissertation is convert the segment clumps file to a shapefile, and the raster attribute table to plain text. The main script was `python_gdal_polygonize_Mars2000EqCyl_lat0_40_objectsz_`

Zonal stats were calculated with the Souness extents and head areas, based on the layerstack before 9999 is added. The script for this is in a different place, at `/ioSafeBackup/RemoteSensingPlanSci_MSc/PythonScripts/MarsPythonScript` on the main hard drive rather than the external one. The original version was accidentally deleted.

9 Collating the stats

The zonal stats for the different files are collated together with the script `python_zonalstats_HRSC_Mars2000Eqcyl_lat0_40_layerstack_combinefiles.py`. Again the original was accidentally deleted.

10 Creating the classifier function

The next step is to compare zonal stats for the Souness objects with the overall histograms of the HRSC tiles. Scripts such as `plot_histograms_allfields_ratios_totaltile` are used. A different script is written for each of heads, extents, contexts, context9s, and heads5km, though heads and extents are what are really of interest. This creates a 4 component gaussian fit to the ratio of the GLF distribution as a function of each feature variable.

The script `CSVfile_addGaussianClassifier_allLandSerfLayerstacksMars2000_lat0_40_n` uses these fits to write the classifier function to the ASCII table. This version doesn't use the nadir image digital number, since these are scaled locally so not globally comparable.

11 Visualisation and statistical tests

Visualisation in QGIS, done via joining shapefile to ASCII file so that the classifier can be visualised. Intersecting the shapefile with the context9 shapefiles, and joining to the ASCII file was done automatically as a QGIS script. This is also used by the Top Trumps.

12 HRSC tile image histograms

In script `rsgislib_imagehistogram.py`.

13 Top Trumps

`subset_nd4_test.py` clips nadir image to bounding box around context, and similarly for a 3 band subset of the layerstack, and rasterize the extents, head, left and right mid-channel, and terminus. convert LnK shapefile (expect joined isectCtx9) to raster. Colorise DTM with another script.