README for performing topographic analysis on filtered Dragon’s Back Pressure Ridge topographic data.

NL\_means\_filter.cpp driver file to perform non-local means filtering following Baudes et al (2005).Copyright (C) 2013 Martin D. Hurst

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Martin D. Hurstmhurst@bgs.ac.uk

Curvature, Slope and Aspect

Curvature, slope and aspect of a DEM derived from a polynomial surface fitted over a prescribed horizontal length scale are computed using curv\_slope\_aspect.py. The script can be run using a python interpreter (e.g. IDLE) or at the command line by typing:

E:\DBPR> curv\_slope\_aspect.py

The analysis can be customised by editing the last 6 lines of code under the condition which tells python what to run if the code is executed:

If \_\_name\_\_ == “\_\_main\_\_”:

Here, the working directory and project name are defined, and the name of the DEM ASCII grid. The *scale* parameter is the size of the window (in pixels) over which a polynomial is fitted in order to derive curvature, slope and aspect. For example a *scale* of 13 on a 1 m resolution DEM results in a polynomial being fitted to a 12x12m window centred on the pixel of interest.

The function outputs ASCII grids of curvature, slope and aspect which can be imported into ARCGIS for visualisation, and are required for subsequent topographic analysis.

Extracting Hilltops

A network of hilltops is generated using the file hilltops.py. This file contains several functions that script ArcGIS functionality using the arcpy module. The script can be run using a python interpreter (e.g. IDLE) or at the command line (in a windows environment) by typing:

E:\DBPR>python hilltops.py

The analysis can be customised by editing the last 20 lines of code under the condition which tells python what to run if the code is executed:

If \_\_name\_\_ == “\_\_main\_\_”:

Here, the working directory and project name are defined so that python knows where to find the DEM. The DEM resolution should be specified and an area threshold (in m2) for defining the channel network. ASCII grids of slope and curvature generated by curv\_slope\_aspect.py should be present in the working directory.

The script generates two folders, “.\\hydro\\” and “.\\basins\\” which will contain the datasets generated by running hydrological analysis on the DEM and extracting drainage basins at each stream order respectively.

A third folder “.\\hilltops\\ will contain the slope and curvature rasters and the extracted network of hilltops. This can be viewed and edited in ARCGIS.

To generate data for individual hilltop segements the resulting hilltop network can be split into segments of a particular length.

* To do this, firstly in ARCGIS, the dissolve function is run (ArcToolbox > Data Management Tools > Generalization > Dissolve) to merge the existing segments into a single line object.
* Secondly, the dissolved line can be split into equal length segments using the Editor toolbar. Use the select cursor to highlight the dissolved hilltops line and run (Editor > Split), and use the “Into Equal Parts” option. Divide the Line Length by the desired segment length to get the number of equal parts required. by the setting the distance (e.g. for 1 m LiDAR use 50 m segment length to result in a sample size of n=100). Save edits.
* The resulting hilltop segments will contain some hilltops that consist of more than one line because of the way the splitting process works. This can be corrected by running the Multpartin to Singlepart tool (ArcToolbox > Data Management Tools > Features > Multipart To Singlepart).
* Finally, there may be some segments that are too small for the analysis. To calculate the length of each segment, open the attribute table and from the drop down menu add a new field to the attribute table called ‘length’. Right click on the new field and click ‘Calculate Geometry’ to calculate the length of each line in the shapefile. You can then remove and segments that are considered too short by selecting them in the attribute table, starting editing with the Editor Toolbar, and pressing delete.

It is necessary for each segment to have its own ID for the next stage of the topographic processing. In the attribute table, select the ID field (or add one if it doesn’t exist), right click on the top bar and click ‘Calculate Field’. Set ID = FID + 1. The hillslope routing requires this data to be in raster format. First, create a buffer around the hilltop network using (ArcToolbox > Analysis Tools > Proximity Tools > Buffer) and setting the Linear unit to the DEM resolution (e.g. 1 m for LiDAR) to analyse 1 pixel either side of the topographic divide. You will then need to convert (ArcToolbox > Conversion Tools > To Raster > Polygon to Raster), setting the ‘Value field’ to ID and specifying the cell size to the resolution of your original DEM. In the ‘Environments’ settings go to ‘Processing Extent’ and select your original DEM to define the extent and Snap Raster (so that the two datasets coincide spatially). This raster will then need to be converted to \*.flt format (ArcToolbox > Conversion Tools > From Raster > Raster to Float) and named something like “db\_seg.flt (where db is the project ID).

Performing Hillslope Analysis

The hillslope tracing algorithm is compiled with the command:

$ make –f hillslope\_tracing\_make.make

which generates the executable ‘hillslope\_tracing.out’. This executable only requires a single input argument, the project description (e.g. ‘db’) but has a number of requirements in terms of associated topographic data. The algorithm requires a number of \*.flt files to be present: a filled DEM (db\_fill.flt) and a stream network (db\_stnet.flt). These rasters can be found in the ‘\hydro\’ folder created by ‘hilltops.py’ and converted to \*.flt in ARCMap. Also required is the slope raster (db\_slope.flt), aspect (db\_aspect.flt; this was created by curv\_slope\_aspect.py but will need converting from ASCII grid format to \*.flt), curvature (db\_curv.flt) and segments (db\_seg.flt) raster generated in the previous section.

This analysis takes a long time! The resulting file generated will be called something like db\_hillslope\_metrics.txt. This contains data for every trace from a hilltop pixel to the stream network (or alluvial plain in the case of Dragon’s Back Pressure Ridge, where db\_stnet.flt was replaced by a mask of the alluvial plain). The data is in 7 columns, the X and Y coordinates of the start of the trace, the segment number, hilltop curvature, mean slope, relief, and hillslope length.

Mean values for each segment can be generated using the script ht\_get\_means.py (edit lines 107-110 to customise for a specific project) which will generate a new text file (e.g. db\_mean\_metrics.txt) containing the mean values for each hilltop segment.