Basic DEM generation from Airborne LiDAR using Open-Source Tools

Edward Fisher ARSF-DAN
Mark Warren ARSF-DAN

1 Introduction

This practical aims to introduce you to a number of LiDAR tools in order to generate a very simple DSM (Digital Surface Model). Please note that this method is not suitable for measuring small changes as data are averaged on import.

DEMs (Digital Elevation Model) including DSMs and DTMs (Digital Terrain Model) are a common product of Airborne LiDAR surveying. They often form the basis for many computational models where terrain is a factor e.g. flood modelling, line of sight analysis and topographic mapping. Many of the tools used in this practical are command-line based, this allows the processes to be scripted.

2 Software

2.1 LAS Tools

LAS Tools is a partly open-source library of tools designed to efficiently manage large datasets. The licences for the tools vary depending on commercial usage. The tools are available as an ArcGIS toolbox, stand alone GUI or command line functions. There are various older version in use currently these may have a different syntax and are unmaintained by the developers. Also, it should be noted that for some tools (e.g. Lasground), if unlicensed will add very small distortions to the data and set some attribute information to zero, in order to protect from illegal commercial use.

Installation through OSGeo: http://www.liblas.org/start.html

Website and alternative download: http://www.cs.unc.edu/∼isenburg/lastools/

2.2 GRASS

GRASS is the former US military GIS system which has since been made open-source and forms the backbone for many GIS packages such as QGIS. GRASS is predominantly command line driven however some elements use GUIs. GRASS can be downloaded by itself or as part of QGIS. For this practical (if your using a windows machine) we will be using the Msys facility this gives the user a UNIX command-line environment.

Manual: http://grass.osgeo.org/grass64/manuals/

Downloads: http://grass.osgeo.org/download/

2.3 QGIS

QGIS (Quantum GIS) is an open-source desktop GIS available natively on Windows, Linux and Mac. In recent years it has grown in popularity becoming the GIS of choice for many organisations /academics. QGIS takes advantage of a number of existing open-source packages such as GDAL and GRASS amongst others. Further to this there are a vast array of plug-ins available to provide extra functionality and task specific tools.

Manual: http://www.qgis.org/en/documentation/manuals.html

Download: http://hub.qgis.org/projects/quantum-gis/wiki/Download

2.3.1 Example Data

The data used in this practical were collected on the 6th of July 2011 by the ARSF over Svalbard, as part of a project to assess the influence of climate change.

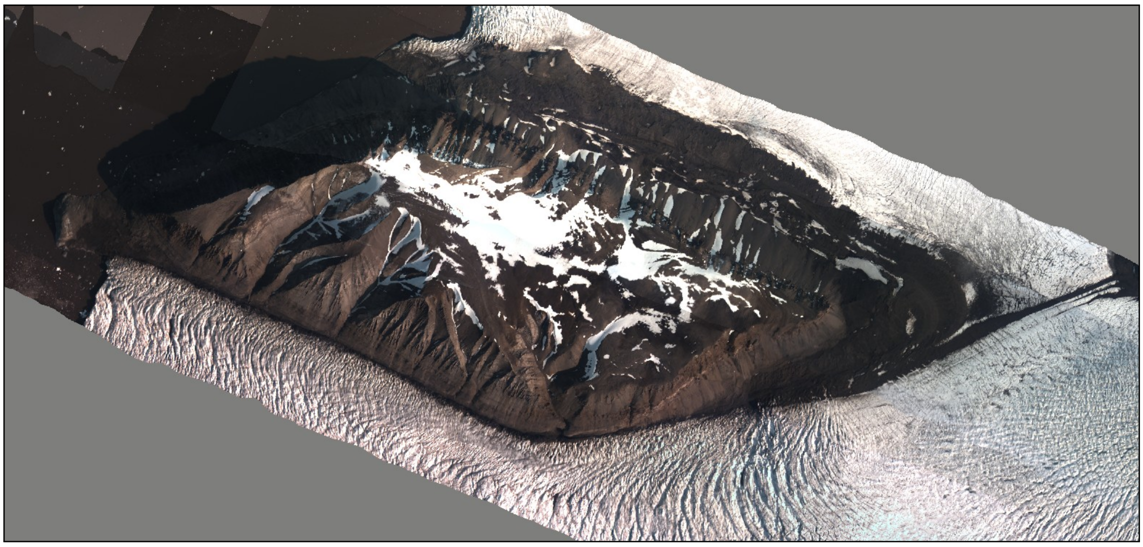


Figure 1: Location of test data - ARSF Hyperspectral Composite Image Position: Lat 78°53' 39.88" Lon +12°39' 6.84"

External File access by FTP
If you are not using our computers then the data is still avaliable from our FTP site (See Appendix for futher details).

ftp://arsf-training:Retae5ziez@ftp.rsg.pml.ac.uk/practicals/LiDAR/

3 Exercise

Using LAS tools it is possible to filter the point cloud according to the point classification. This is very useful for removing noise and features which we do not want to include within the DEM. Point clouds can be classified using a number of packages including LAG (ARSF) and Bentley Microstation. ARSF data is classified as '1' for default classification and '7' for noise. These classification values follow the standards set by ASPRS (American Society for Photogrammetry and Remote Sensing.

Other available tools for filtering and classifying point clouds:
http://www.opentopography.org/index.php/blog/detail/tools_for_lidar_point_cloud_filtering_classification

3.1 Basic Unix Commands

Some of the packages used in this practical are based on UNIX and as such expect a basic knowledge of the operating system. Here are a few of basic commands used for this practical.

pwd = Current directory address
cd = Change directory (also works in Windows)
.. = Up one directory e.g. cd ../.. = Move up two directories
. = Current directory
ls = List current directory content
ls -l = Long list (Write permissions, Owner etc)
-h = Often after a command gives the help file e.g. r.in.xyz -h

It is also possible to scroll through your previous commands by using the up and down arrow keys and the tab key can be used to auto complete the line.

3.2 las2las and las2txt

The las2las and las2txt tools can be used to filter the point cloud by a number of criterion. In this example we will keep it simple by only removing the 'noise' values i.e. Class 7. The tool names are fairly self explanatory las2las will take the input las file filter it and output a las file, whereas las2txt will take a las file and convert it into ASCII. In this case it is probably best to use las2txt as GRASS requires point clouds to be entered in ASCII format. If lastools are already set up on your computer for use on the command-line then you can use the following commands.

(Please Note: If you wish to skip this section to save time or wish to continue with the practical whilst las tools is processing then the filtered converted files have also been provided)

If you are using windows quotation marks are required around the file names e.g. "C :\...\...*.LAS".

3.3 Command-Line Instructions

3.3.1 Single File

```
las2las -i <input_las_file.LAS> -keep_class 1 -o <output_las_file.LAS>
```

```
las2txt -i <input_las_file.LAS> -keep_class 1 -o <output_las_file.LAS>
```

3.3.2 Directory (Kept as separate lines)

```
las2las -i *.LAS -keep_class 1 -odir <output_directory> -olas
```

```
las2txt -i *.LAS -keep_class 1 -odir <output_directory> -otxt
```

Merging flightlines has its advantages and disadvantages. First big problem is loading into GRASS in a later part of the practical as GRASS loads the points into RAM which may become full when using large datasets. There is a way around it using percentage restriction, but this is an extra complication. Also individual flightline information is lost. For these reasons the merge files function should not be used in this practical.

Also please note that these commands will only keep the XYZ coordinates, in order to keep more you will need to use -parse followed by the character (representing a field) you wish to retain. For more details see the -h option (as below). If you choose to add more fields then some later instructions in this tutorial will possibly require adapting.

3.3.3 Help / Further Options

```
las2las -h
```

```
las2txt -h
```

If you prefer to use a GUI (graphical user interface) approach then you can use las2txt as follows.

3.4 LASTools GUI Instructions

3.4.1 How to Start las2txt GUI

Windows
Find the downloaded LASTools folder and double click on the relevant .exe file within the bin folder.

Linux/Mac
The GUIs are designed for Windows based usage however it is possible to use them with tools such as WINE for Linux or Parallels for Mac.

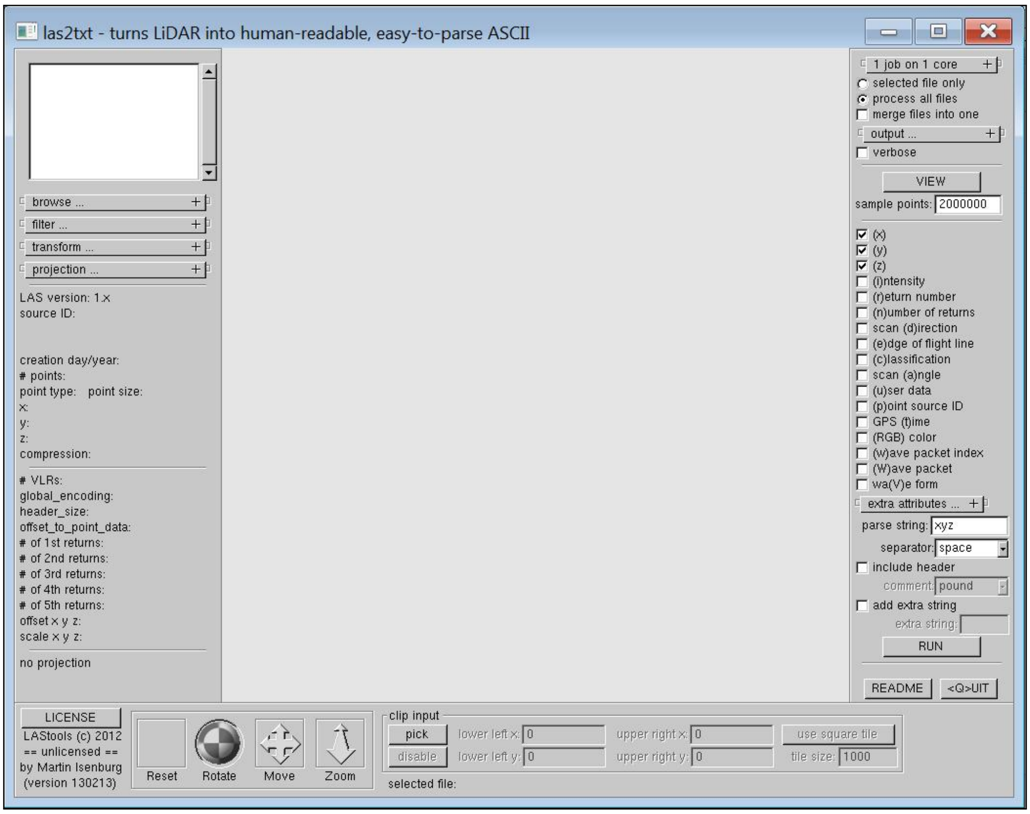


Figure 2: las2txt GUI setup

Now using the GUI load in the LAS files which require filtering.This can be achieved by using the browse tab and double clicking on the lines required. We then need to apply a filter. Click on the filter tab and look in the drop-down menu 'by classification or return', select keep _classification. Then in the box for 'number or value' put 1 and press add. This rule can be removed by selecting the line and pressing delete.

Once the filter is set up then the output needs to be defined. The options for this are in the top right corner. Press output, then the ... button and find/make a suitable location to save the files.

At this point you can choose to merge all the flight lines into one file. For the reasons previously mentioned this option should not be used.

Next the information that you wish to include should be selected e.g. X,Y,Z, Intensity, Point Number etc. For these instructions only X,Y,Z have been selected. Please note that different combinations of information included with alter the column order in later steps i.e. including more information may result in a different column order to that in these instructions.

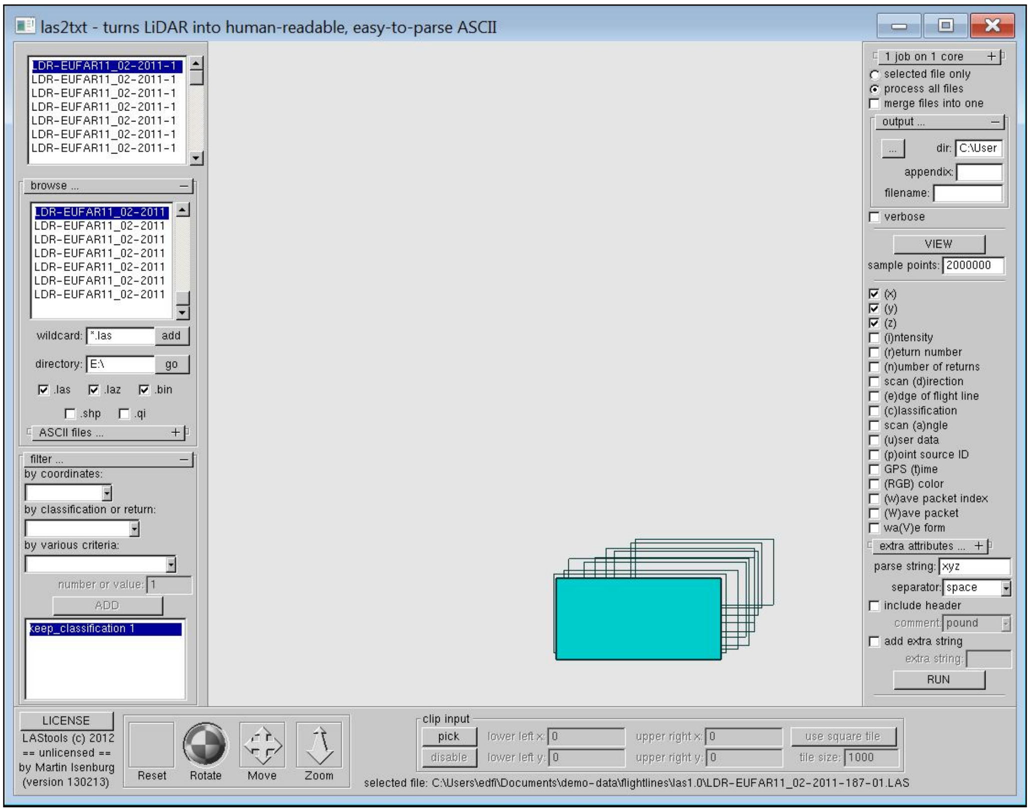


Figure 3: las2txt GUI setup

Now the parameters have been set up press run (Default window size sometimes hides run button so just re-size). This will then show you the command-line command that it will run which you can edit manually if you wish. It can be useful to take note of this if you wish to later run the same settings through command-line. If the command looks ok then press start. The process can take a while depending on the size of the dataset etc. You may wish to continue the practical using the supplied ASCII files to save time.

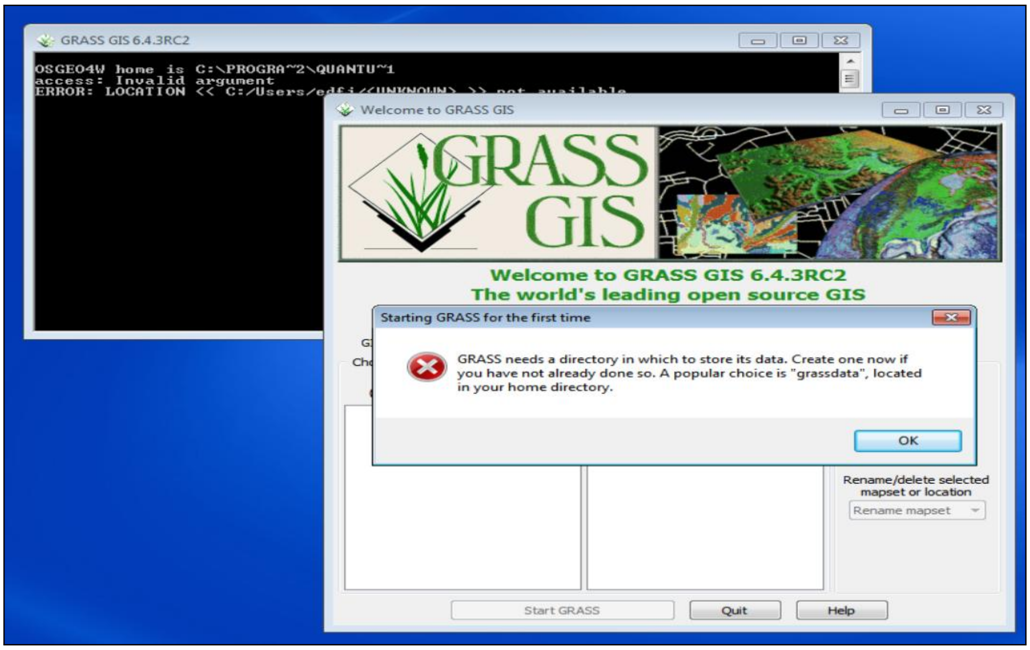
Once the filtering is complete the data is then ready to be mosaicked into one point cloud to create a DEM for the AOI (Area Of Interest). This can be achieved using GRASS.

3.5 Using GRASS

3.5.1 Setting up GRASS

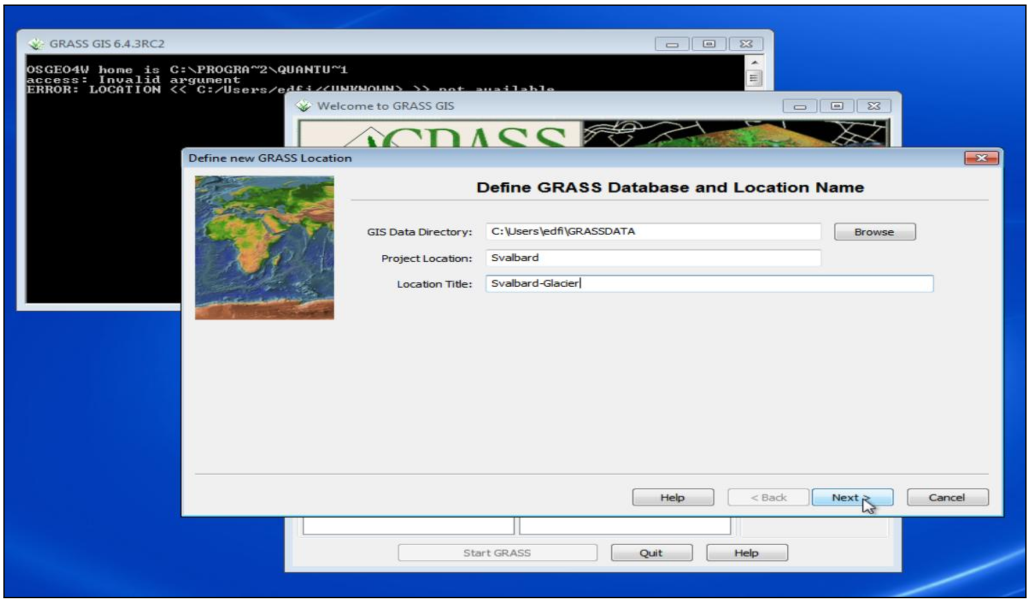
In Windows start Grass using 'GRASS GIS 6.4.3RC2 GUI with MSYS' (Linux or Mac start GRASS normally). MSYS will open GRASS with its standard GUI along with a UNIX console. If you are using a Linux or Mac system then the UNIX console should start as default.

When opening GRASS for the first time you will be prompted to set a GRASS directory location. This is effectively a database where all the project areas are stored. Find or make a suitable location.

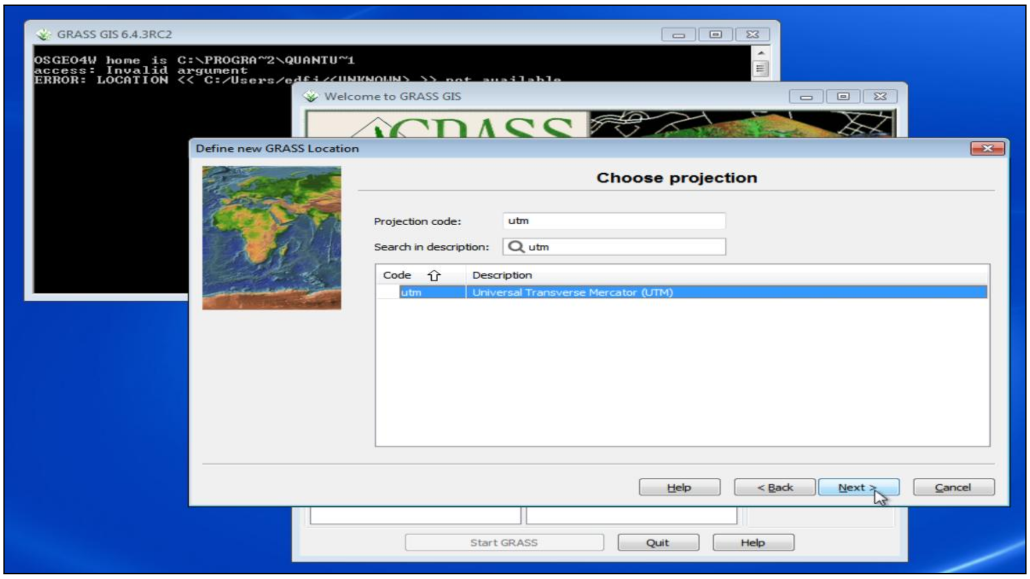


Once the GRASS directory has been assigned then the environment can be defined i.e. Projection etc. To do this the location wizard can be used.

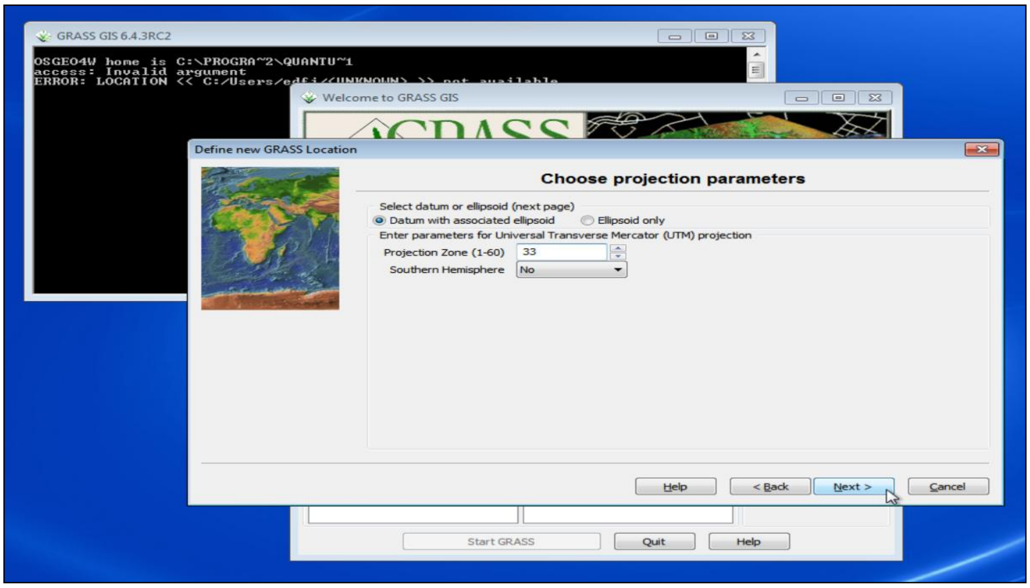
1. Firstly the project and locations need to be given titles.



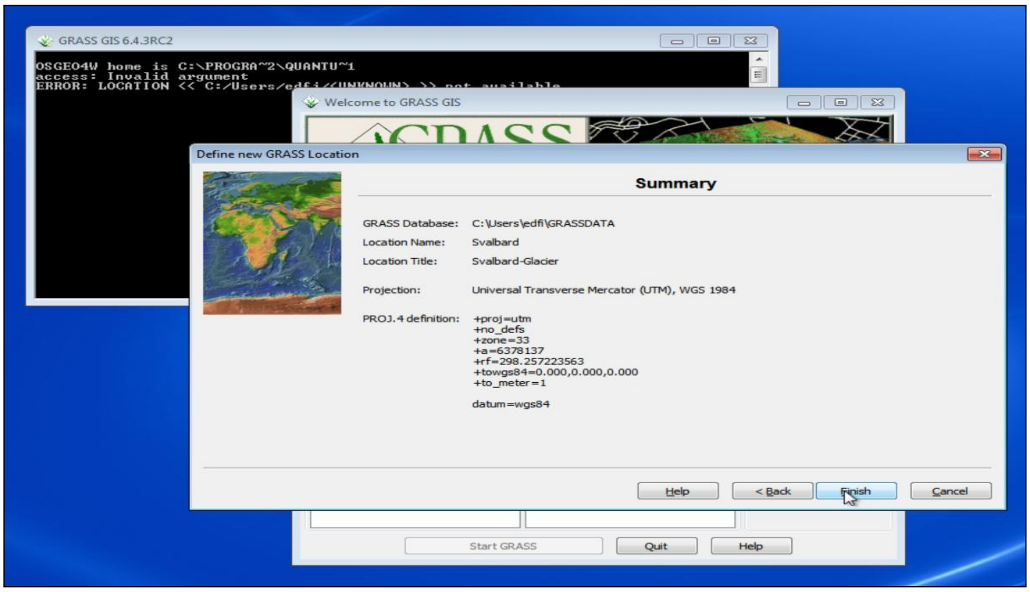
2. Next in the case of this practical we can 'select coordinated system parameters from a list' press next then select UTM (Universal Transverse Mercator) and press next.



3. The UTM is a global reference system divided up into 60 zones in the two hemispheres. Svalbard is in UTM Zone 33. Further to this we want to use a 'Datum with associated ellipsoid'. Press next.



4. Here in a similar way to the projection we need to define the datum. In this case WGS84 is suitable. Once this is complete on the next window you will get a summary of the 'GRASS location'.



Having defined the location you will be asked if you 'want to set the default region extents and resolution now'. Say NO as this will be defined later using the extents of the LiDAR data.

Quick Start Guide:
http://grass.osgeo.org/grass65/manuals/helptext.html

3.5.2 Mosaicking

GRASS can now be started using the Svalbard projection location. This will launch the GIS Layer Manager and the map display. Each of the following processes can take a while depending on the file sizes.

Each ASCII line can now be loaded into GRASS. To do this the region for each line must be defined before importing.

Step 1

Navigate to the location of your data using the Unix commands introduced previously. Then the following command can be used to scan through the data to collect the extent of the line.

```
r.in.xyz -s  input=<LDRfilename> output=<outputmapname> x=1 y=2 z=3
fs=‘ ’
```

where:
-s = Scan data for extent
input = ASCII point cloud
output = Grass variable name
fs = Field separator

e.g.

```
GRASS 6.4> r.in.xyz -s input=LDR-EUFAR11_02-2011-187-01.txt output=line_1 x=1 y=2 z=3 fs=' '
x: 447248.859000 451843.643000
y: 8757174.318000 8759520.952000
z:  110.161000  399.804000
Range:     min        max
GRASS 6.4>
```

Step 2

The output from step 1 can then be plugged into g.region using the following command. This is in order to define the 2D space that the point cloud will populate. You may wish to add a small buffer.

```
g.region n=max_northing s=min_northing w=min_easting e=max_easting res=2.0
```

e.g.

```
GRASS 6.4> g.region n=8759520.952 s=8757174.318 w=447248.859 e=451843.643
```

Step 3

Once the region has been defined we can load in the related line. This is achieved by using the same r.in.xyz command as before except the -s flag is removed. The points in this line will then be loaded into GRASS.

```
r.in.xyz input=<LDRfilename> output=<outputname> x=1 y=2 z=3 fs=‘ ’
```

e.g.

```
GRASS 6.4> r.in.xyz input=LDR-EUFAR11_02-2011-187-01.txt output=line_1 x=1 y=2 z=3 fs=' '
Reading data ...
 100%
Writing to map ...
 100%
r.in.xyz complete. 7305809 points found in region.
GRASS 6.4>
```

These three steps then need to be repeated for all the flightlines which are required in the DEM. It is possible using BASH script to write a loop to do this automatically i.e. scan the line, define the area, load the points, repeat until done.

(Aside: If at any point you wish to visualise the lines by using the GUI windows of GRASS. In the GIS Layer Manager add the Raster you wish to view by pressing the ‘Add raster map layer’ button. Then by pressing the eye icon in the GIS Map Display.The map should be appear. If not then press the ‘Zoom to selected map area’ button.If you wish to add a legend or scale bar this is possible using the ‘Add map elements button’.)

When this is complete the overall region for the mosaic needs to be defined. The command below creates the space which can be used to combine all the flightlines into one file.

```
g.region rast=line_1,line_2,...line_n
```

This space is then populated using the r.patch command

```
r.patch in=line_1,line_2,...line_n  out=mosaic_lidar
```

e.g.

```
GRASS 6.4> g.region rast=line_1,line_2,line_3,line_4,line_5,line_6,line_7,line_8,line_9,line_10
```

Followed by

```
GRASS 6.4> r.patch in=line_1,line_2,line_3,line_4,line_5,line_6,line_7,line_8,line_9,line_10 out=mosaic_lidar
 100%
Creating support files for raster map <mosaic_lidar>
GRASS 6.4> 
```

More information
`r.in.xyz -h` = Help file

Manual page for commands:
http://grass.osgeo.org/grass64/manuals/r.in.xyz.html
http://grass.osgeo.org/grass64/manuals/g.region.html

3.6 Filling Holes

3.6.1 Small Holes

It is inevitable that as with most airborne data collection methods holes will occur in the data due to clouds or other naturally occurring events. There are various different approaches for filling these gaps depending on the size and other data available. This section will introduce two simplistic methods for dealing with holes.

Firstly for dealing with small holes it is possible to patch a lower resolution grid into the existing grid. This is achieved by re-sampling the original mosaic and then creating a composite raster.

```
g.region res=5
```

This g.region command will change the grid size to 5 metre pixels which should be sufficient for covering small holes.

```
r.resamp.stats input=mosaic_lidar output=lowres_mosaic
```

r.resamp.stats will then resample using the new grid as defined by g.region. This creates a lower resolution mosaic. The new lower resolution grid can then be appended into the original mosaic by using the r.patch command.

```
r.patch in=mosaic_lidar,lowres_mosaic out=high_low_patched
```

e.g.

```
GRASS 6.4> g.region res=5
GRASS 6.4> r.resamp.stats input=mosaic_lidar output=lowres_mosaic
 100%
GRASS 6.4> r.patch in=mosaic_lidar,lowres_mosaic output=high_low_patched
 100%
Creating support files for raster map <high_low_patched>
GRASS 6.4> 
```

3.6.2 Large Holes

Larger holes in the dataset can be addressed in a few ways. Firstly the introduction of other data from an alternative source, for example ASTER or TanDemX data. However assuming that there is no other data available the existing data can be interpolated to cover the gaps.

One method uses an spline interpolation to fill the null values. This can be implemented as below

```
r.fillnulls input=raster_to_fill output=filled_raster tension=40. smooth=0.1
```

More info: http://grass.osgeo.org/grass64/manuals/r.fillnulls.html

e.g.

```
GRASS 6.4> r.fillnulls input=mosaic_lidar output=lidar_fillnulls tension=40. smooth=0.1
```

3.6.3 Filtered Data

Particularly in noisy datasets it is often a good idea to smooth the data. This can help to remove spikes in the data and some noise which may have otherwise escaped previous classification. One way to do this would be to use the following command. Other filtering options can be explored by looking at the options in the r.neighbors -h.

```
r.neighbors input=raster_to_filter output=filtered_raster method=median size=3
```

e.g.

```
GRASS 6.4> r.neighbors input=lidar_fillnulls output=filtered_nulls_raster method=median size=3
 100%
GRASS 6.4>
```

More info: http://grass.osgeo.org/grass64/manuals/r.neighbors.html

3.7 Writing out of GRASS

Although the process of writing out files from GRASS in itself is straightforward, care needs to be taken in order to output in the corrected format. The three common ways are:

Write out as ASCII:

```
r.out.ASCII input=raster_name output=output_filename null=0
```

e.g.

```
GRASS 6.4> r.out.ascii input=filtered_nulls_raster output=DEM_ascii null=0
```

Alternatively if your are using ENVI and require a BIL file then you could use:

```
r.out.gdal format=ENVI type=Float32 input=raster_name output=output_filename nodata=9999
```

e.g.

```
GRASS 6.4> r.out.gdal input=filtered_nulls_raster output=DEM_gdal nodata=9999
ERROR 6: SetColorTable() only supported for Byte or UInt16 bands in TIFF format.
Exporting to GDAL data type: Float32
Checking GDAL data type and nodata value
 100%
Exporting to GDAL raster
 100%
r.out.gdal complete.
```

Or if you are using ArcGIS or a program that requires the data to be in an ArcGrid e.g. TerrainBender then you will require this output.

```
r.out.arc  input=raster_name  output=output_filename
```

e.g.

```
GRASS 6.4> r.out.arc input=filtered_nulls_raster output=DEM_arc
 100%
GRASS 6.4> █
```

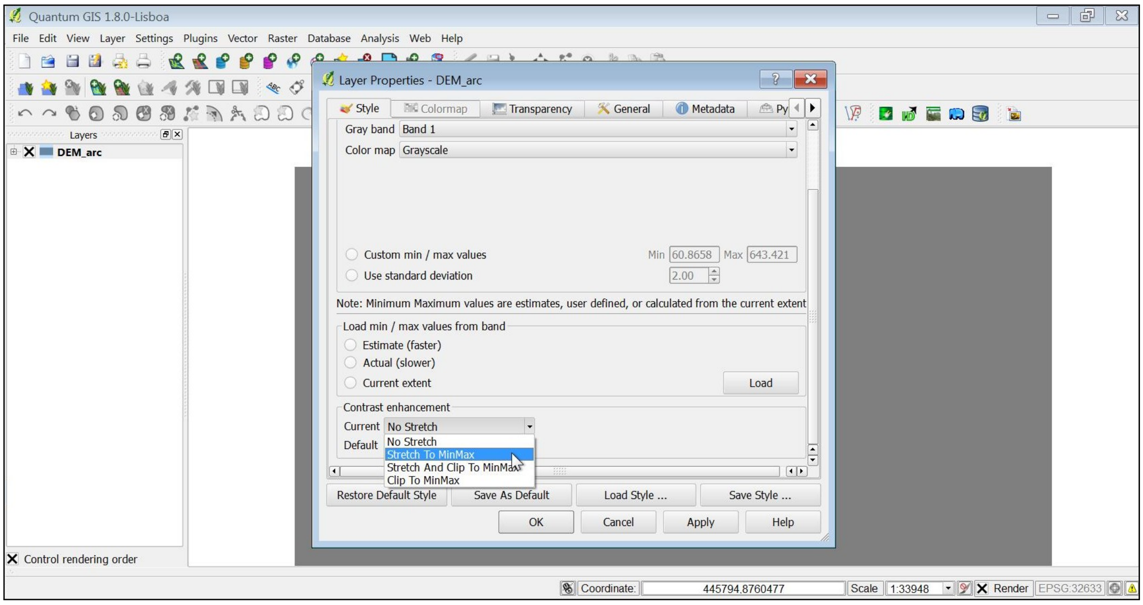
For more raster tools and output options see:
http://grass.osgeo.org/grass64/manuals/raster.html

3.8 Simple Products from DEMs using QGIS

3.8.1 Hillshade

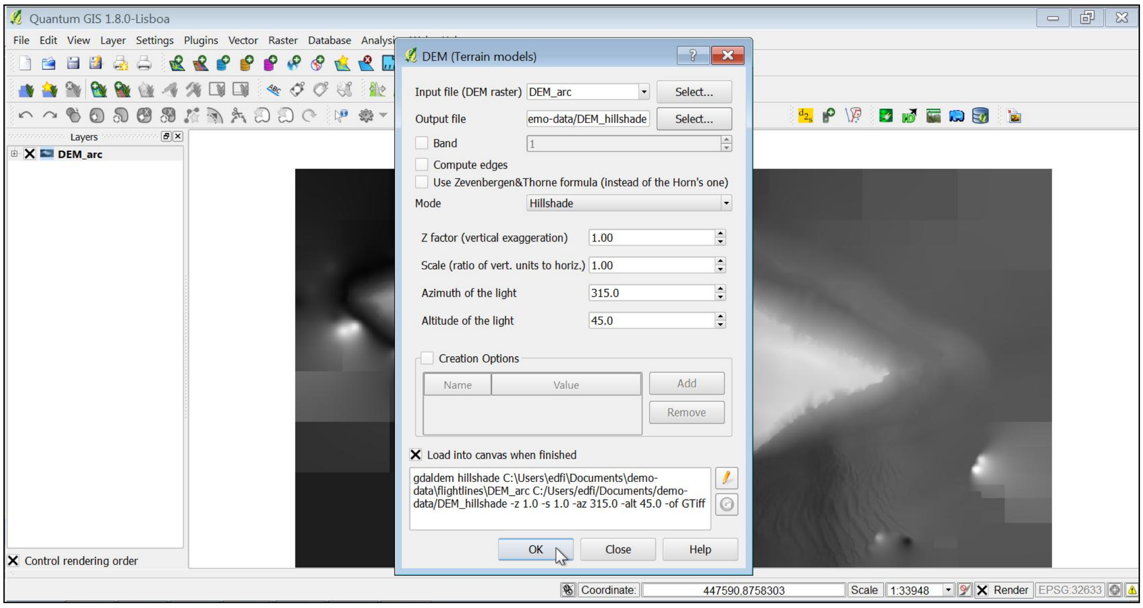
From this created DEM we can easily create some simple products using QGIS. Firstly a Hillshade model can be produced by simply loading in the Raster DEM just produced by GRASS (all three outputs should work in QGIS). When loading in the DEM you will be asked what Coordinate Reference System (CRS) you wish to use, in the case of the example dataset use WGS84 / UTM zone 33N.

Once loaded the DEM will appear as just a grey square. To help visualise the DEM the you can use contrast stretch. To do this right-click on the DEM in layers and go to properties (or double click on the layer). Then under the styles tab at the bottom there is the option to contrast stretch the image. In this case the best results were from 'Stretch to MinMax'.



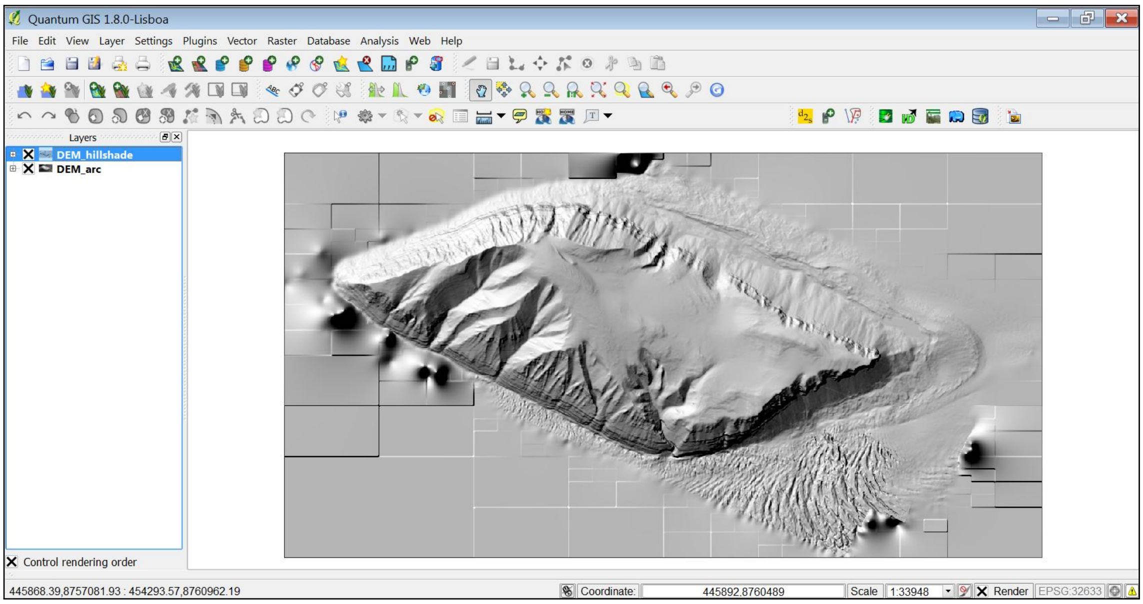
Although this step is not required to create a Hillshade model it does help to give an impression of what to expect.

Next the model can be created using the tool under Raster → Analysis → DEM (Terrain models).



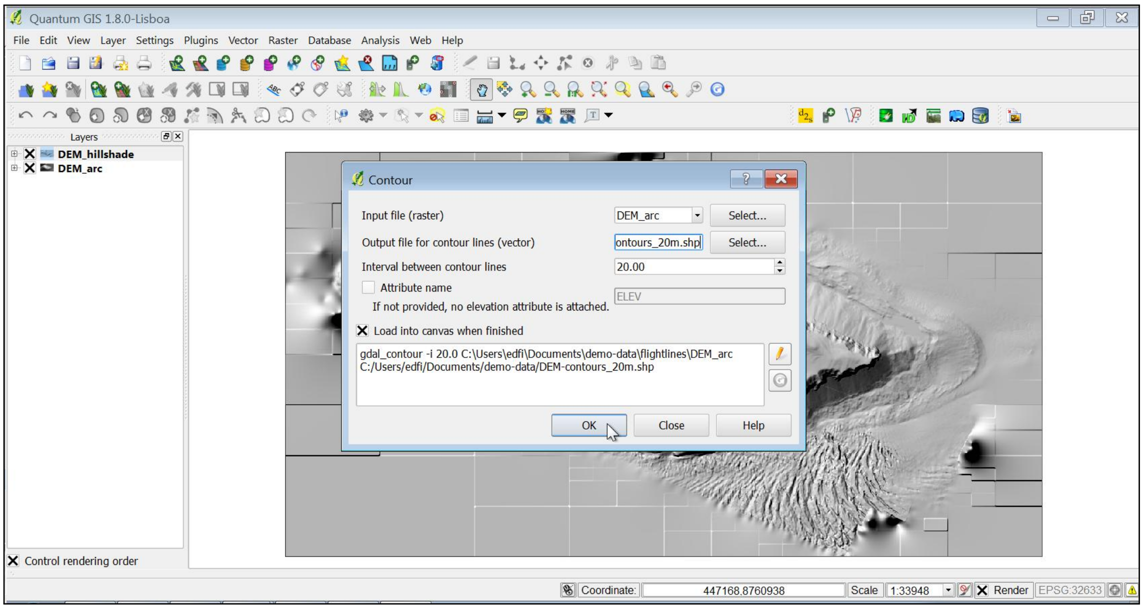
Again there are settings which can be played about with but for the most basic model everything can be left as default. To do this an output file needs to be assigned and the 'Load to canvas when finished' box should be ticked.

The resultant model should look similar to the to the one below.

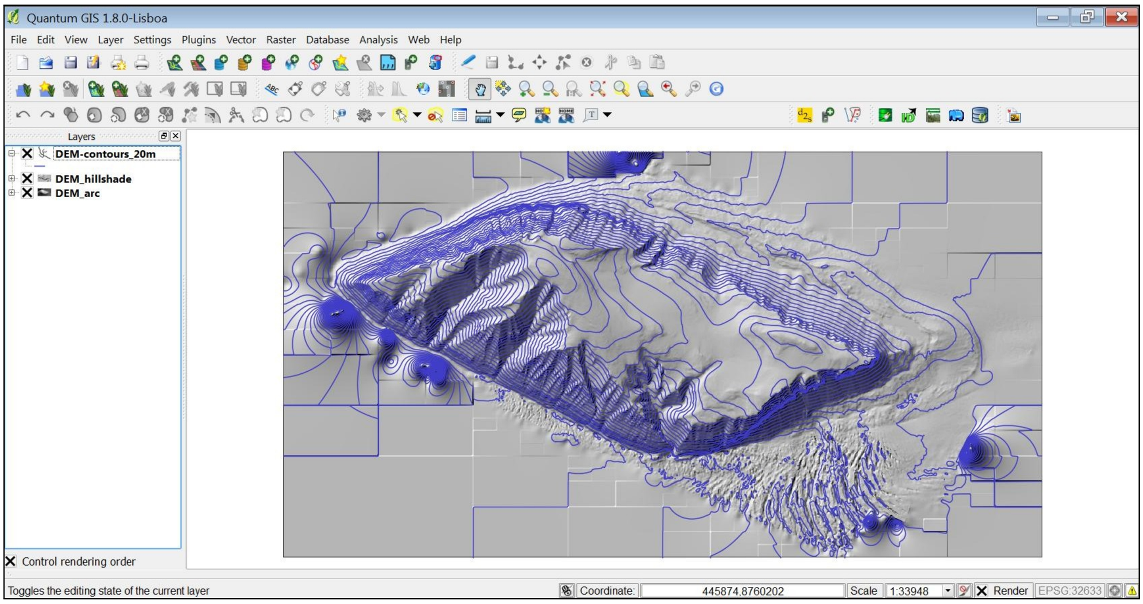


3.8.2 Contours

For some applications such as topographic mapping it can be useful to have contour lines. These can be extracted from a DEM by going to Raster → Extract → Contour. As this dataset has rapid changes in its relief a larger interval (default 10m) may be advisable.



A 20 metre contour interval will result in lines similar to the ones below.



3.8.3 Alternative Method

If you prefer to use the command-line then you could do this in GRASS:

http://grass.osgeo.org/grass64/manuals/r.contour.html

http://grass.osgeo.org/grass64/manuals/r.shaded.relief.html

3.9 Downloads and Additional

GRASS: http://grass.osgeo.org/

QGIS: http://www.qgis.org/

Lag Download: http://arsf.github.com/lag/
(Currently only runs on Linux)

LasTools: http://www.cs.unc.edu/ isenburg/lastools/

Interesting Links: http://www.lidarbasemaps.org/

Tool and community: http://www.opentopography.org/

4 Appendix - FTP File list

ftp://arsf-training:Retae5ziez@ftp.rsg.pml.ac.uk/practicals/LiDAR/

Filename	Description
`DEM_Practical.pdf`	PDF copy of instructions
`LAS1.0`	Binary LiDAR File Standard - Three Columns X,Y,Z Location: /lidar_dem_tutorial_data.zip/lidar_dem_tutorial/las1.0/
`ASCII`	Text Format LiDAR - Three Columns X,Y,Z Location: /lidar_dem_tutorial_data.zip/lidar_dem_tutorial/las2txt-filtered/