LiDAR Processing in TNTmips

A Sample LiDAR processing workflow and Manual for using the Terranean SMLs.

June 26, 2009

This document outlines a work flow to produce DEMs from LiDAR LAS files using TNTmips and a set of SML scripts produced by Terranean Mapping Technologies. The scripts are a work in progress but provide a starting point for anyone wanting to process LiDAR using TNTmips and especially for developing SML scripts to analyse or process LiDAR.

TNTmips generic raster and geometric tools makes it a powerful and flexible environment for LiDAR processing, especially for analysis of LiDAR point clouds. Terranean operates two different brands of LiDAR processing software and there is no question that TNTmips has a way to go before it can produce DEMs from LiDAR as efficiently as the specialised LiDAR software. However, TNTmips has made excellent strides in this direction in the past six months, supporting native LAS format, the new profile tool for displaying and selecting LiDAR data and optimising some processes such as TIN generation to better handle massive point clouds. At this rate, TNTmips will soon be a fully functional and efficient LiDAR processing system. The advantage of TNTmips over other software are the generic tools and the SML scripting which provide far greater flexibility for processing and analysing LiDAR data than specialised LiDAR processing software.

It should be noted that Terranean Mapping Technologies uses an infra-red full waveform Riegl type LiDAR. This type of LiDAR is much more useful for measuring above ground information, such as vegetation, but less cost effective for broad area DTM survey than other types of LiDAR that are designed for flying high and fast. High density LiDAR lends itself to being processed as raster surfaces because there are more points per square metre and hence points/pixel. The surfaces are filtered and edited to produce a DEM that is then used to classify the LiDAR points into ground and non-ground.

By contrast, with low density LiDAR (<1 point/m2),the point cloud is usually classified first then a terrain surface is interpolated from the ground points.

These SML scripts have been tested on our high density LiDAR but it has not been determined how well they will work where the LiDAR point spacing is greater than the DEM pixel size.

Disclaimer*:*

> *The SML scripts referred to herein are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Terranean Mapping Technologies shall not under any circumstances be liable to any person for any special, incidental, indirect or consequential damages, including, without limitation, damages resulting from use of or reliance on the information presented, loss of profits or revenues or costs of replacement goods, even if informed in advance of the possibility of such damages. Every effort has been made to ensure the accuracy of the information presented, however, Terranean Mapping Technologies assumes no responsibility for the accuracy of the information.*

Distribution:

> *The SML scripts referred to herein may be freely distributed and adapted provided that the source is acknowledged and author credited.*

Processing Steps

Step 1
Run LAS_to_Raster.SML to produce raw surfaces from unclassified LAS file. This script takes multiple LAS files and produces three rasters at user specified resolution. The output rasters are:

1. "Min" Z value of the lowest LiDAR point falling on the pixel.

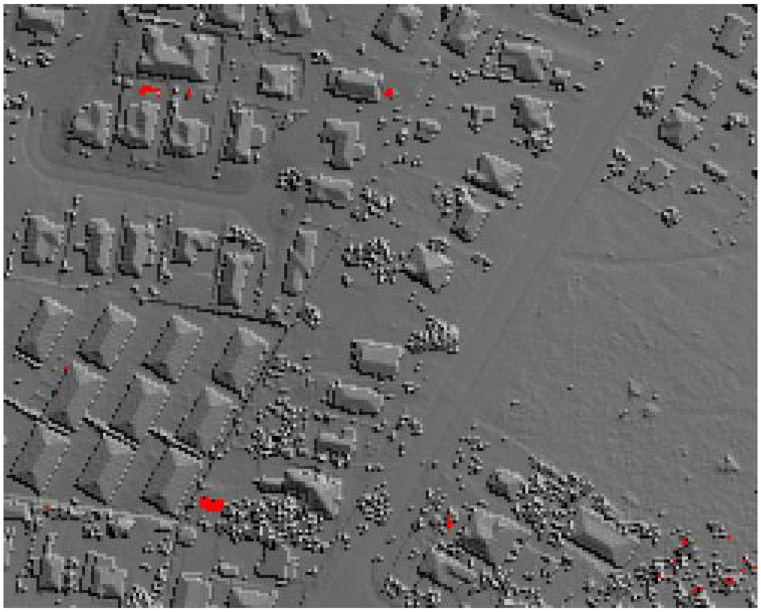
2. "Max" Z value of highest point on the pixel

3, "Count" Number of LiDAR points falling on the pixel.

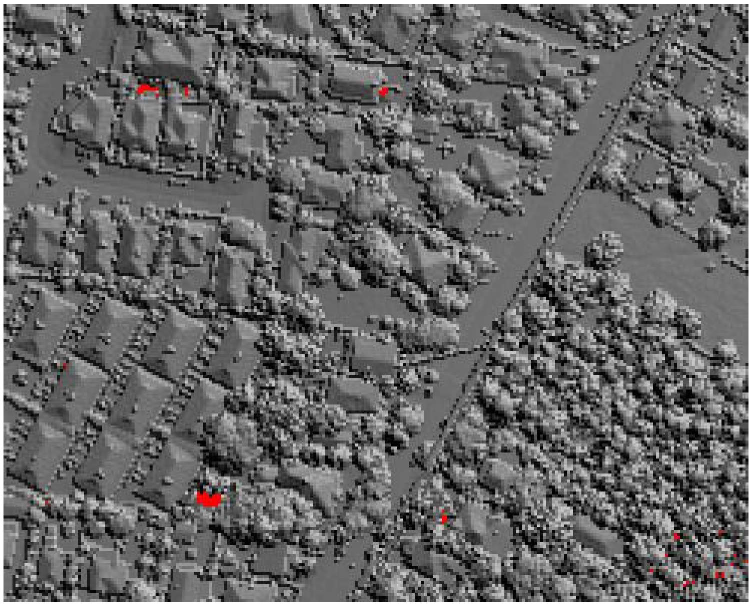
The Min surface is a good starting point for generating a DEM. It contains a lot of ground surface, plus buildings and probably some vegetation. The surface needs to be filtered to remove the non-ground features.

The Max surface represents vegetation and also hard surfaces which have nothing above them (roofs, bare ground). The difference between Max and Min is the highest point above the hard surface.

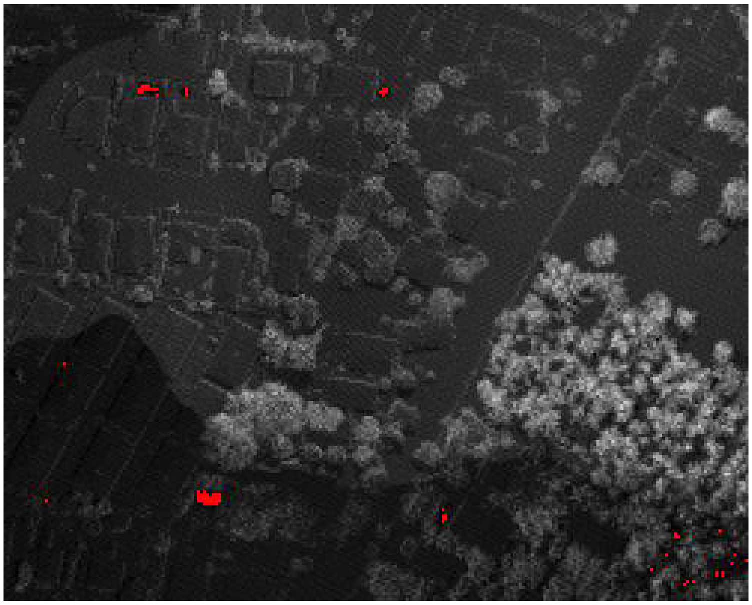
Note that the script takes a set of LAS files that would normally be flight strips or tiles. If the LAS files are strips, these need to be leveled against each other. Adjustment to ground control can be applied to the final DEM or any intermediate products, but strip leveling should be applied before generating surfaces. TNTmips currently does not have a strip leveling tool, although it is possible to measure the offsets between strips using the profile tool and then run a script to adjust z-values by the required amount for each strip. Until TNTmips includes a strip leveling tool it is best to work with LiDAR that has been leveled and tiled.



Min Raster – Lowest LiDAR point on pixel.

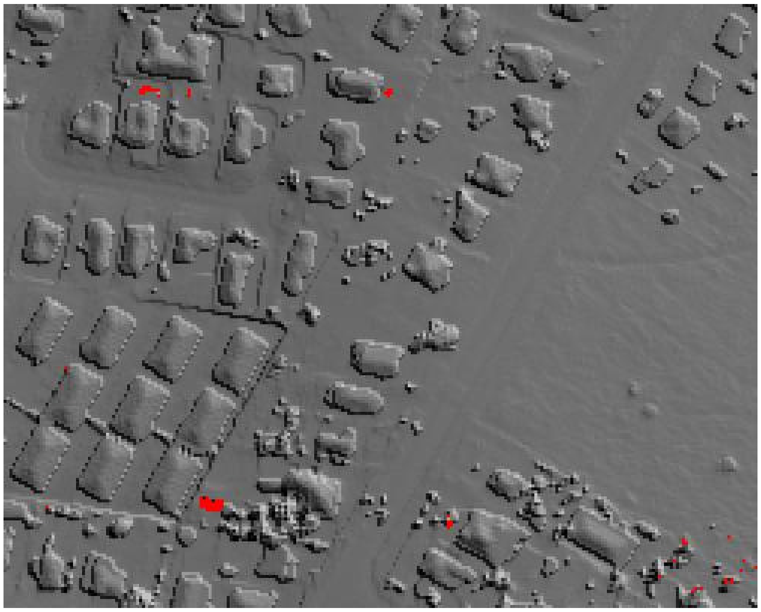


Max Raster – Highest LiDAR point on pixel.



Count Raster – Total LiDAR Points per pixel.

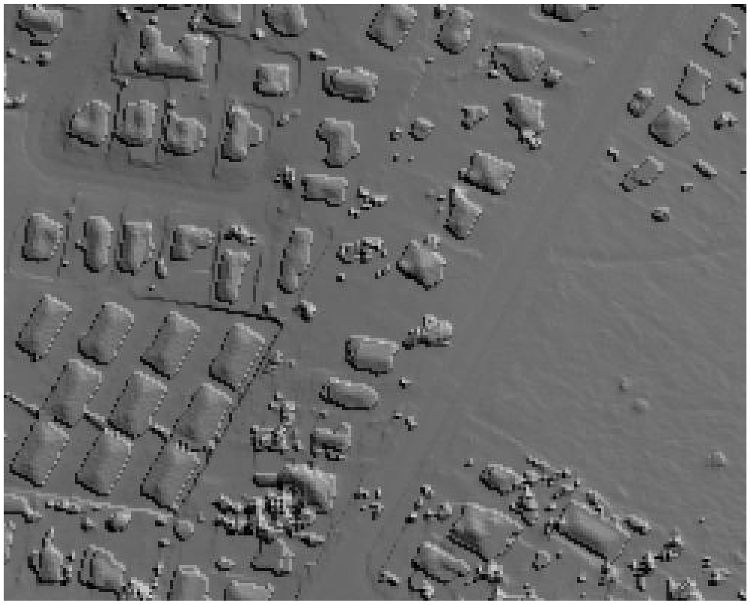
Step 2
Use the Raster\Filter\Spatial Filter… to run a 3x3 median filter on the Min Raster. This will remove anomalous high points. It will also remove points that are below ground, for example points at the bottom of drains.



Min Raster after 3x3 Median filter has been run

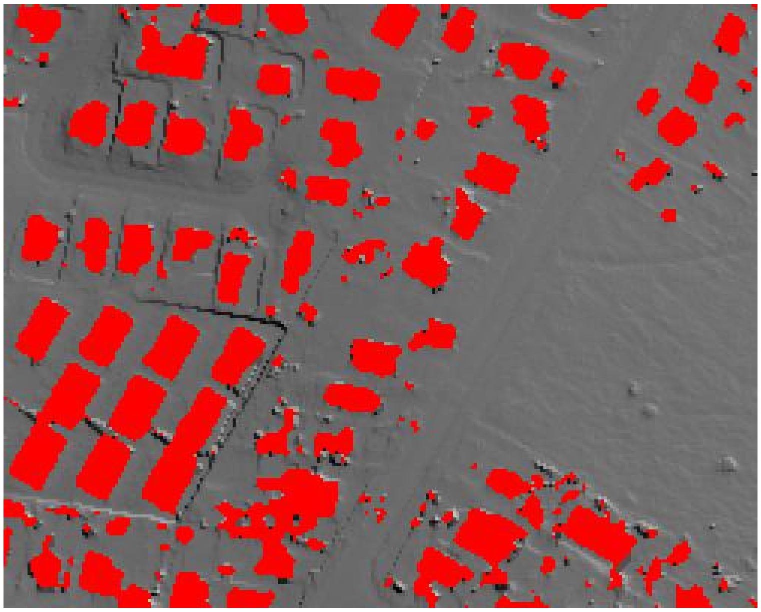
Step 3
Run Interp.SML to fill gaps resulting in water, building shadows etc. This script is also used to fill gaps created by filtering and editing. The algorithms use two interpolations:

1. Fill small holes with the mean of the neighbouring pixels. (default 3x3 filter).
2. Triangulate across larger gaps by TINing from gap-edge pixels.



Interp.SML fills gaps.

Step 4
Run *Filter_LiDAR_DEM SML*. This script is designed to remove built non-ground objects such as buildings and cars from the DEM while retaining important ground features like retaining walls and banks. The filter will not produce a perfect result but generally removes more than 90% non-ground cells from the input DEM when using the default parameters. It also removes some ground pixels that need to be replaced at a later stage. The filter changes non-ground cells to NULL.



The majority of the Non-Ground areas are made null

For good results, surface filtering to remove non-ground data is an interactive process. Typically we will start with a sequence of filters that worked well for another job, try it on a test sample and fine tune it before running the filter on an entire dataset. Sometimes it is necessary to mask different areas and filter, eg. urban and forest areas, separately. It can be challenging to find the best filtering process for specific datasets, but once the filters are set up, the batch process will do 90% of the work and minimal editing is required for final cleaning.

This SML provides the framework for one method of terrain filtering. It is a work in progress and we hope others will make improvements and share with the TNTCommunity. The literature is full of methods for LiDAR surface filters and LiDAR point cloud filters. This sml constucts a TIN by finding nodes at the lowest points in the landscape, within square search window, typically larger than 30 pixels. If the LiDAR surface is less than a threshold height above the TIN surface, the surface is kept. If the LiDAR surface is above the threshold, the pixels are treated as non-ground pixels and filtered out. A problem with this script in its current form is that the temporary TINs do not cover the full extent of the DEM and some data is lost on the edges.

Three parameters control the filtering:

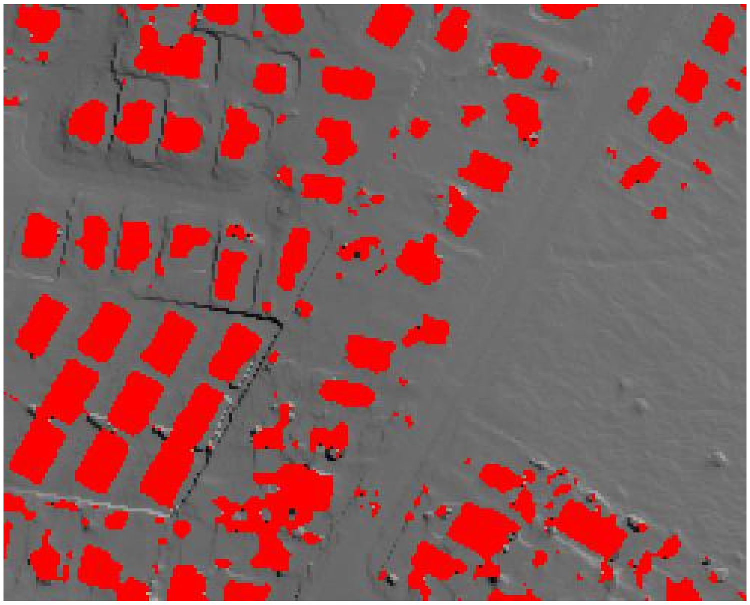
1. "*horizontal*" sets the horizontal extent of the moving window that finds the TIN nodes. Increasing this makes a coarser TIN, possibly removing ridge lines and hill tops. Decreasing this makes the TIN finer, increasing the probability that a TIN node will be located on a non-ground surface such as the roof of a large building. In this case the building will not be filtered out because the TIN is going over the top of it.

2. "*vertical*" sets the maximum height above the TIN surface for which pixels are treated as ground surface. Where ever the LiDAR surface exceeds this height above the TIN, the pixel is considered "non-ground" and filtered out.

3. "*max_slope*" In steep terrain it is more difficult to distinguish terrain features from non-ground. This parameter prevents the filter from removing pixels in areas where the slope of the TIN exceeds max_slope.

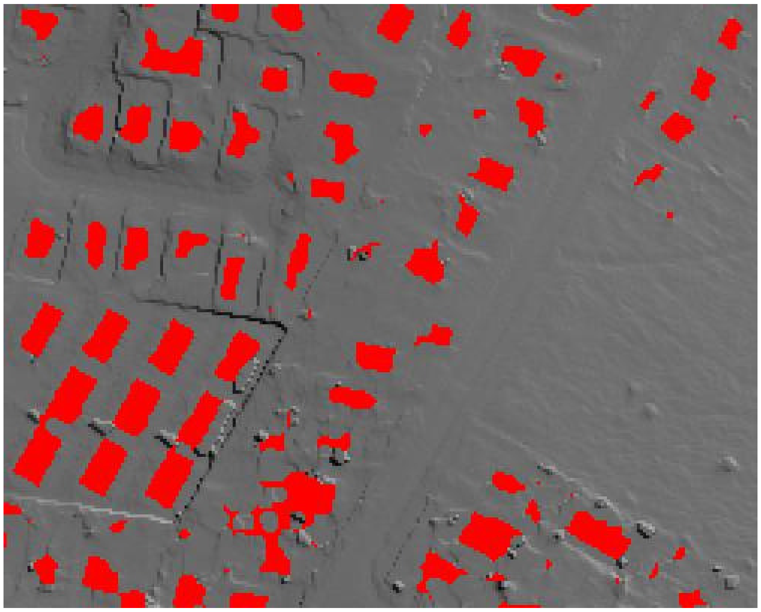
By increasing "horizontal" and reducing "vertical", more of the surface is filtered out. Also, increasing max_slope increases the amount of the surface that can be filtered.

Step 5
Run another Median filter. The default values for the Filter_LiDAR_DEM SML tends to retain edges of buildings, running a median filter removes some of these smaller artifacts.



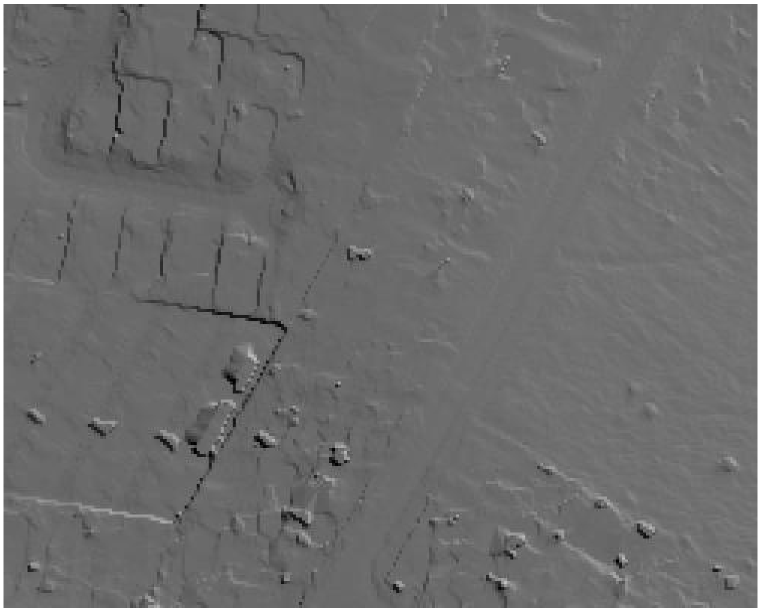
A few of the remaining spikes are filtered out

Step 6
Run *Edgefill.sml*. This fills in a small number of pixels around the edges of gaps. It uses a Focal Median function.



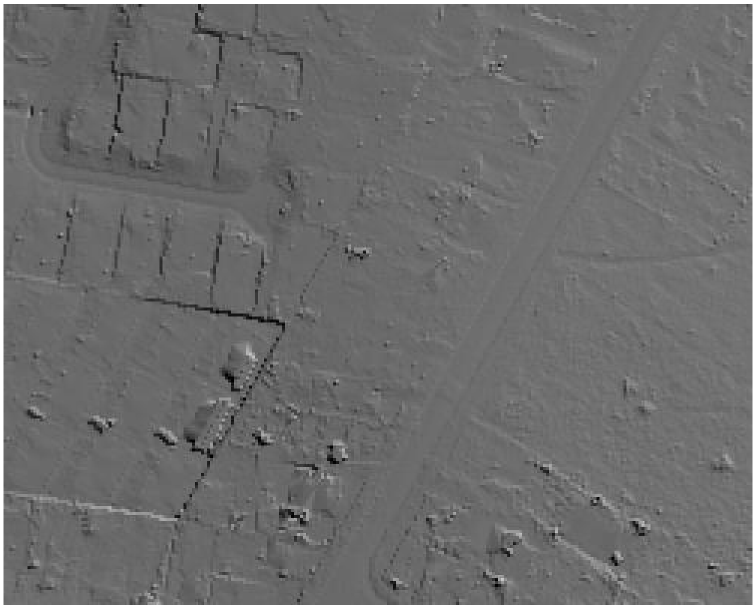
Edgefill.SML reduces the number of spikes in the interpolated DEM

Step 7
Run *Interp.SML* on the output from the edge fill. This will interpolate across non-ground features that have been filtered out.



Interp.SML fills gaps in the filtered DEM

Step 8
Run *Reinstate.sml*. Most of the non-ground features will have been filtered out and the DEM contains no gaps. However, some ground pixels will also have been filtered out. This SML reinstates data into the filtered raster from the original interpolated Min raster where the height difference is small – ie where the ground surface has been filtered out. Reinstate.SML asks for a threshold value. Where the difference between the filtered surface and the raw data is less than the threshold, the interpolated values will be replaced with the original raw pixel values. A threshold of around 0.5m seems to work in most situations. The script takes two rasters as input. Use the output from Step 7 as FilteredDEM and the output from Step 1 as DEM.



Reinstate.SML replaces interpolated surfaces with the raw values where the values are similar.

Step 9
Classify LAS files. The SML Classify_LAS.sml classifies the LiDAR into ground and non-ground points, depending on their height above the DEM. A parameter "*threshold*" sets the maximum distance from the DEM for points to be classified as ground points. Often two thresholds are used to classify definite ground and non-ground points leaving intermediate points within a range (say 0.1 – 0.5 metres) unclassified.

Manual editing of the DEM or classified points can be performed in TNTEdit, using the relief shaded DEM as a reference to identify anomalies. We generally edit the classified point cloud to reclassify points and then interpolate a final DEM from the classified point cloud using TNTmips Surface Modeling module.

Classify_LAS.sml reads and modifies LAS database records and converts between LAS coordinates and raster coordinates. It would be a good starting point for developing other scripts that analyse and manipulate LAS files, either in isolation or in conjunction with rasters.