DTM assignment 2

Qu Wang, 4700686, Pablo Ruben, 4273818

18 December 2018

1 Overview of results

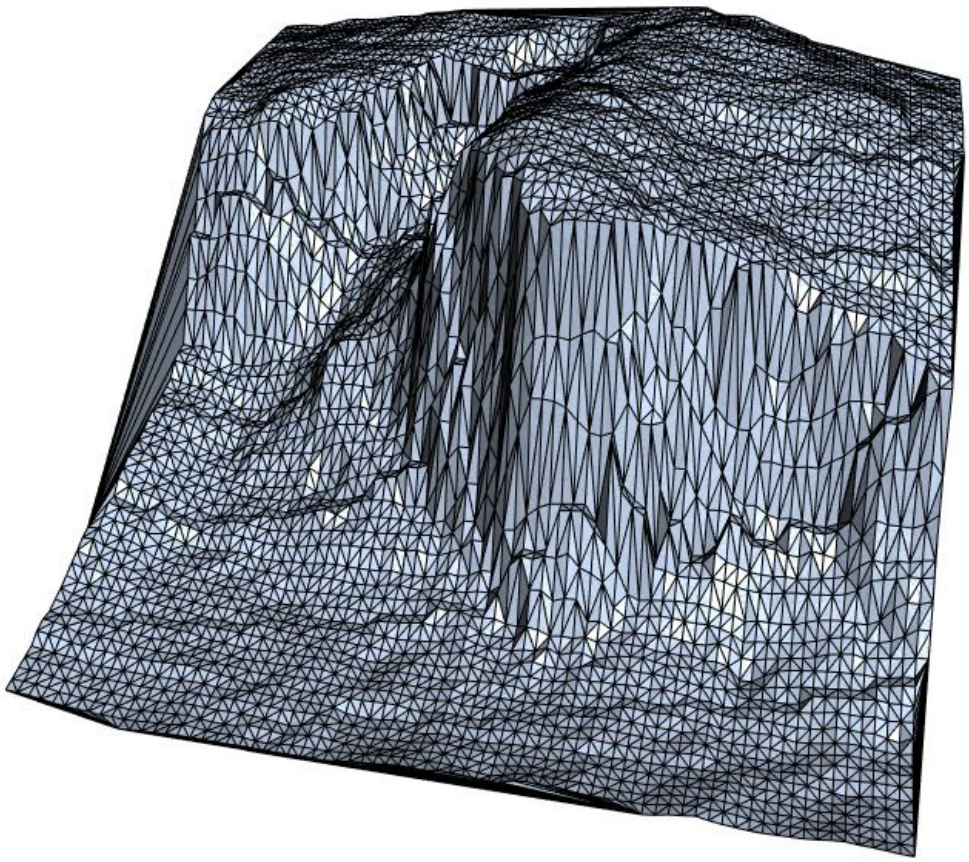
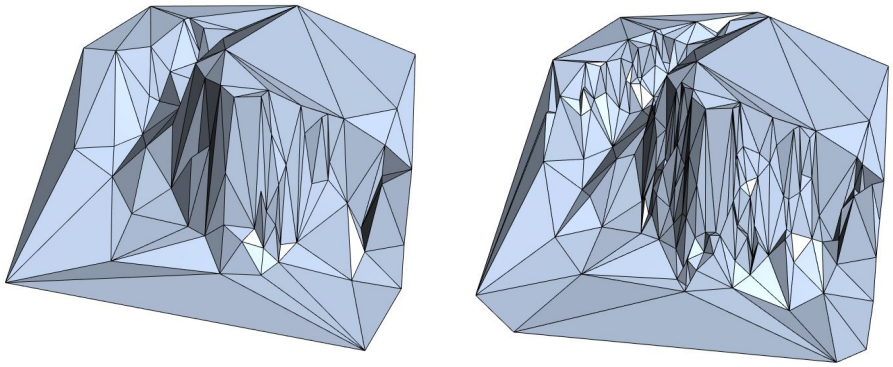
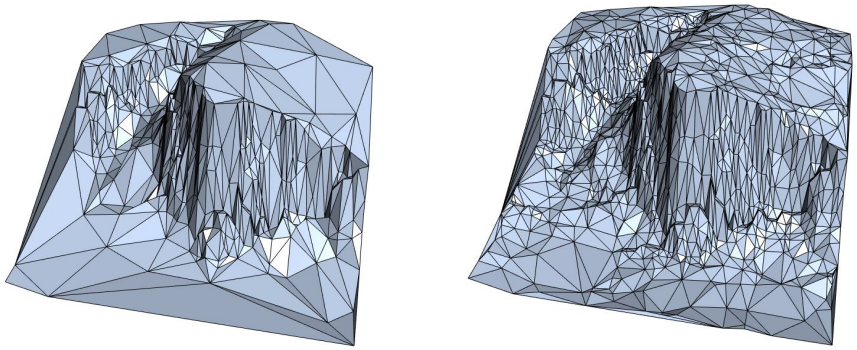


Figure 1: Original DTM file, before refinement



(a) Result with 100m threshold

(b) Result with 50m threshold



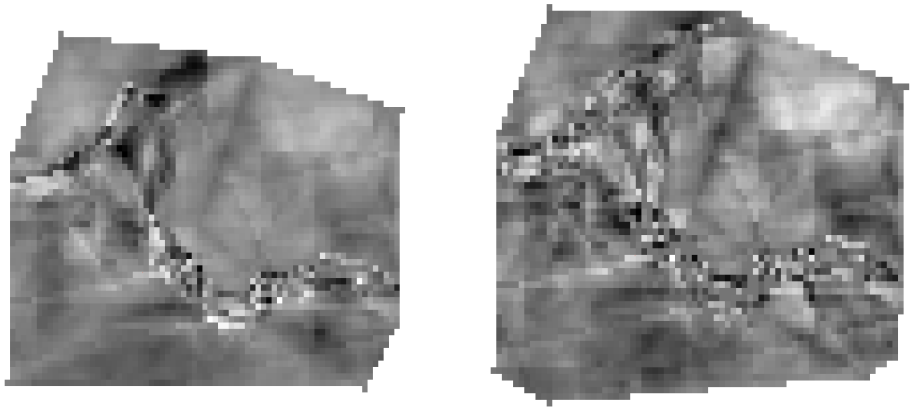
(c) Result with 25m threshold

(d) Result with 10m threshold

Figure 2: Results of the DTMs after refinement using different thresholds.

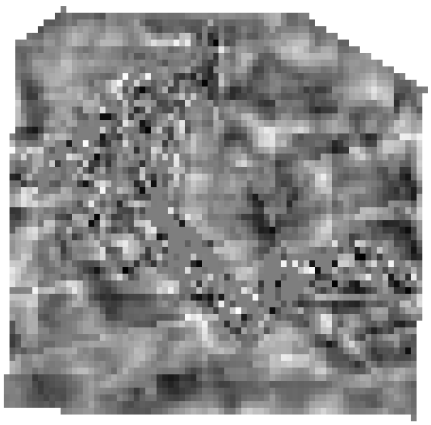
Figure 2 shows the resulting meshes from the refinement algorithm applied on Figure 1 and Figure 3 shows the error results for the respective thresholds. For each of the figures, the extreme values are represented in white and black. For instance for Figure 3 the top left (threshold = 100m) a white pixel indicates a value of about -100 and a black of about 100m. For these figures, a bounding of 3* the width and height of the minimum bounding box was used.

There are 4405 vertices in the original TIN, and Table 1 shows the number of vertices left with respect to different error threshold, after simplifying with a 10 meter error threshold, there are 982 important point remaining. One can observe that the larger the threshold the smaller the number of vertices that remain.

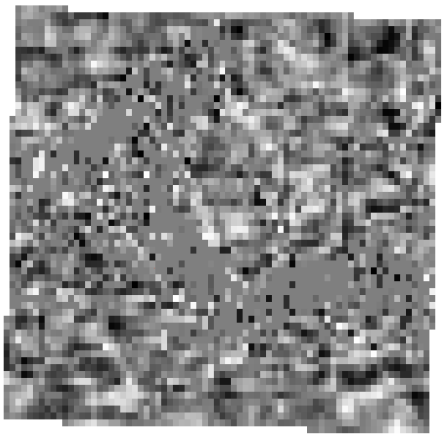


(a) Differences for 100m threshold

(b) Differences for 50m threshold



(c) Differences for 25m threshold



(d) Differences for 10m threshold

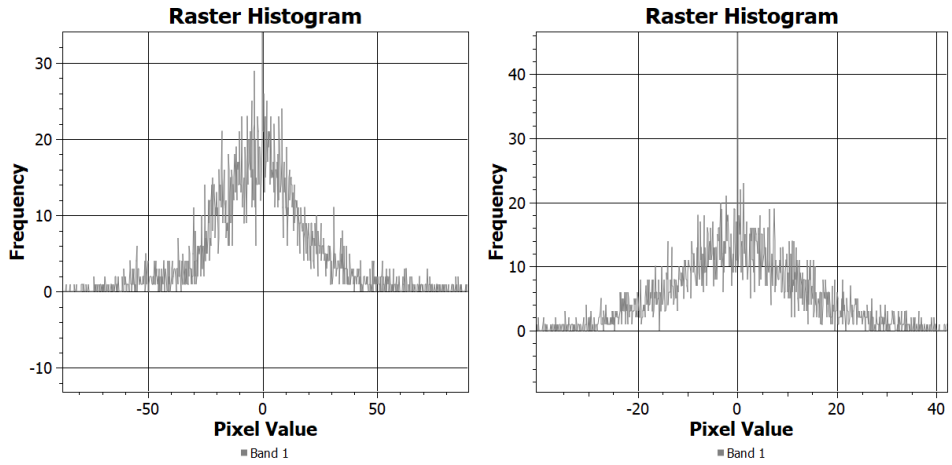
Figure 3: Results for different thresholds, note that white and black are the big (respectively positive and negative) differences, grey are the smaller differences.

2 Spatial pattern of errors

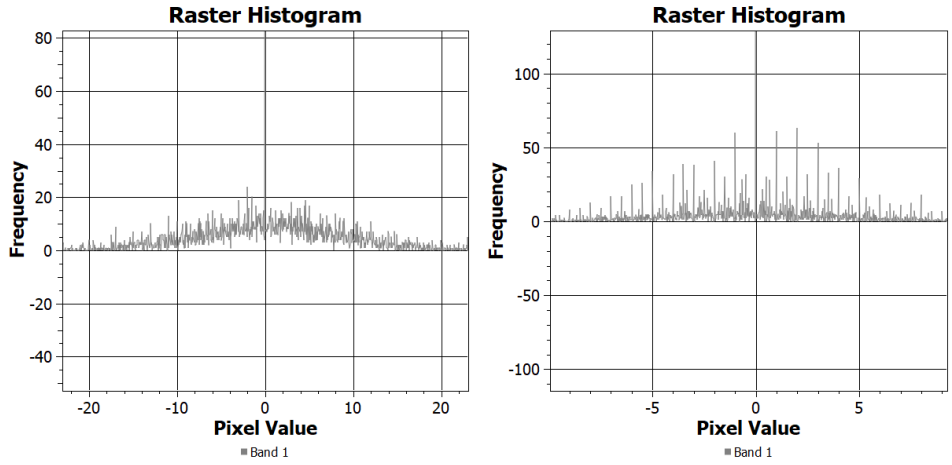
One can observe that the errors in Figure 3 are concentrated on the zone where the slope gradient is strongest for the 100 and 50m refinement threshold. From the 25m threshold on, one can observe that a grey (thus low difference) zone appears close to the sloped zone. This is probably due to the fact that where the slope gradient is strongest and the threshold is low, points are more likely to be kept. In opposition, for low slope gradient, more errors (relative to the threshold) can be found for a low threshold than for a high one.

3 Statistical distribution of errors

The histogram of the errors in difference grids is shown in Figure 4. For different error thresholds,they all line within [-error_threshold, +error_threshold]. And we can observed, the statistical distribution is a Gaussian one, centralizing at 0, and the variance becomes larger with the error threshold increasing. It is because that simplifying a TIN with a smaller error threshold will result in more vertices, which will give a more detailed TIN terrain model. Then the differences tend to be smaller. While using a large error threshold, the simplified TIN will result in less vertices, and show larger differences.



(a) Histogram for 100m threshold difference (b) Histogram for 50m threshold difference



(c) Histogram for 25m threshold difference (d) Histogram for 10m threshold difference

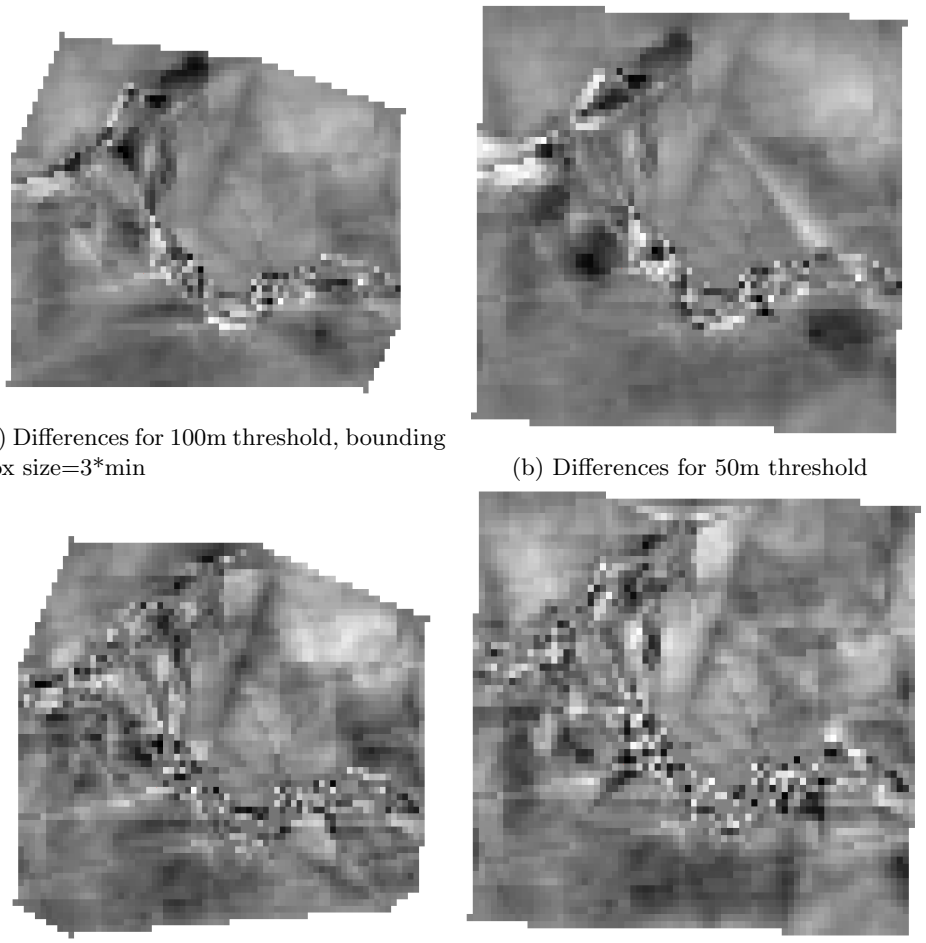
Figure 4: Histogram of the errors in difference grids

The highest difference might potentially be higher than the threshold if the differences are not checked after insertion of the last point. This was consciously avoided in our code as the maximum difference value is only updated after all points have been checked again: instead of using the second highest difference, all differences should be recomputed - even after insertion of the (potentially) last point.

4 Performance of the different thresholds

Another interesting effect that is the change of the convex hull depending on the size of the bounding box as seen in Figure 5. The bounding box is used to create the two initial triangles at the start of the refinement process and the size used to calculate it was added as a variable to compare results. Two clear trends can be observed:
- the convex hull is better conserved with a low error threshold (probably because the ratio of conserved points is higher).
- with a smaller bounding box, the convex hull is better conserved than with a big bounding box. This is probably related to the fact that the bigger the initial triangles, the lower the decrease of height between the inserted point and the initial triangles. This phenomenon rather affects the flat ends of

the DTM - on Figure 3 one can see that the limits of the DTM with a high slope gradient are less affected by the convex hull modification.



(a) Differences for 100m threshold, bounding box size=3*min

(b) Differences for 50m threshold

(c) Differences for 50m threshold, bounding box size=3*min

(d) Differences for 50m threshold, minimum bounding box

Figure 5: Results for different thresholds, note that white and black are the big (respectively positive and negative) differences, grey are the smaller differences.

A good way to improve the performance of the algorithm would be a selective update of the differences within the refinement algorithm. In fact, in the current situation, all differences get updated as soon as a point is inserted. Actually only the points inside the triangle which was modified by the point insertion need to be updated. This would require indexing the list of points using the existing triangles but might eventually speed up the code (the expected effect is a bigger acceleration as the number of triangles after refinement increases).

As shown in 1, the refinement computation time more than linearly increases as the threshold decreases. In fact, as the threshold decreases, the number of iterations needed to add all points that should be included in the refined version increases. As with each added point the number of triangles increases by 2, the time needed to find the triangle containing a point (to calculate the interpolated value and compare it to the threshold) increases more than linearly. Interestingly, the refinement computation time needed per vertex in the end set is rather stable (around 0.3 s/pt). It therefore seems that the number of points in the end file is a good indicator of the needed computation time

	computation time for refinement [s]	computation time for differences [s]	number of vertices	number of faces	computation time for refinement / number of vertices [s]
error threshold			bbox size = 3		
10m	311.57	0.40	982	1946	0.317279925
25m	116.72	0.42	382	752	0.305544009
50m	59.00	0.44	193	375	0.305716331
100m	22.80	0.46	78	147	0.292281615
error threshold			bbox size = 1		
10m	313.82	0.41	974	1930	0.322198229
25m	123.96	0.34	403	794	0.307595723
50m	65.62	0.38	213	414	0.308075659
100m	26.65	0.35	87	162	0.306273153

Table 1: Overview of computation time and number of vertices for different bounding box and vertices size

(although the exact ration might be different from one terrain to the other). This is true for both the usage of a small and a big bounding box - this variable does not seem to considerably influence the bounding box. The computation time needed to calculate the differences shows much less variation - no clear trend can be identified here. The computing time is likely to depend on the number of triangles that need to be visited to 'travel' from the last visited to the triangle containing the new point (this is especially critical once a line switch occurs in the raster). To understand this more in depth a look at the precise distribution of triangles within the DTM would be needed. Also, several datasets should probably be tested in order to get more reliable, average indications.