Simplification Algorithm for Airborne Point Clouds

Matthew Westaway[1], George Sithole[2]

[1] Geomatics Division, University of Cape Town, Cape Town, South Africa, wstmat003@myuct.ac.za
[2]Geomatics Division, University of Cape Town, Cape Town, South Africa, george.sithole@uct.ac.za

This paper has been through a process of double-blind peer review

Abstract

The processing of point cloud data is influenced by the size of the point cloud. Today the size of point clouds has increased due to improvements in laser scanning technology; therefore processing of point cloud data has become computationally exhaustive and storage demands have increased. Consequently, accurate point cloud simplification algorithms are being sought. In this paper, a point cloud simplification algorithm is proposed. In the presented algorithm, important points (feature points) are preserved and redundant points (non-feature points) are removed, whilst ensuring the point cloud satisfies a minimum density requirement globally.

The importance of a point can be evaluated by the relationship it has with its surrounding points. This relationship is quantified by using measures of information theory. These measures use the surface geometry about a point as a proxy for information. Five measures from past work are compared by quantitatively and visually measuring the accuracy of the simplified point cloud. The measure of information theory that results in the most accurate detection of feature points is found.

1. Introduction

With the development of modern higher resolution laser scanners the size of point clouds has increased dramatically and subsequent operations on the point clouds have become computationally demanding. The size problem can be overcome by structuring the points in trees such as octrees and kd-trees, segmenting the point cloud into fewer features, optimizing the memory by using out of core algorithms and forcing the graphical processing unit of a computer to perform the computation or by point cloud simplification. Point cloud simplification speeds up subsequent mesh operations and is therefore computational less intensive. With the development of recent point-based modelling, visualization and multi-resolution applications the demand for meshing has decreased. This has renewed interest in simplification algorithms that minimize the loss in quality of the simplified surface through improving decisions on which points are relevant (feature points) and which are redundant and can be deleted (non-feature points). Such decisions are being improved by developments in the field of information theory i.e. the increasing ability to detect important points, without any prior knowledge of the source and regardless of the structure of the point cloud.

This paper presents a point cloud simplification algorithm that preserves feature points and satisfies a minimum density requirement globally. The focus is on airborne lidar point clouds as previous work on point cloud simplification has only dealt with terrestrial point clouds. Lidar point clouds are non-uniformly distributed and so have a point density that varies significantly from area to area in comparison to terrestrial point clouds. They therefore require a unique simplification algorithm that takes into account such variations.

The next section presents a review of current simplification algorithms. In the third section of the paper the proposed algorithm is discussed. The fourth section of the paper presents the results of varying levels of simplification. Finally a conclusion of the research is presented.

2. Previous work

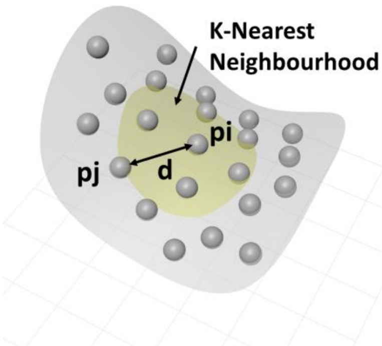
Simplification algorithms output a point cloud in which the points are either directly measured (subset) or sampled. The latter is a much antiquated method of simplification that has become less popular. The papers reviewed consider algorithms that work directly on the point cloud i.e. before surface reconstruction. All of the algorithms were applied to terrestrial point clouds.

2.1. Simplification by sampling

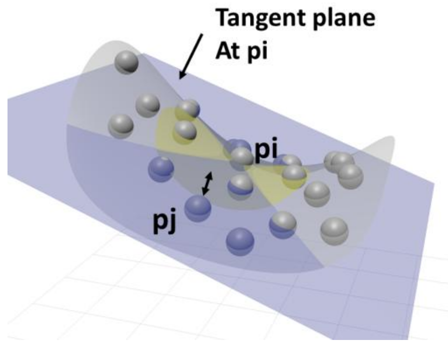
Sampling simplification involves sub-dividing a point cloud into equally sized cells and replacing all points that fall into the same cell by the average. This results in a uniform simplification (every point having an equal chance of being removed from the point cloud). Pauly et al. (2002) strongly opposed this algorithm noting if the wrong cell size was chosen clustering would easily join unconnected parts on the surface. As a result, Pauly et al. (2002) proposed two approaches. The first was a region growing approach where clusters were built by adding neighbouring points until a local sampling density was reached. The second was a hierarchical clustering approach; this algorithm followed the same steps as the region growing approach however the clusters were split if they had values higher than their maximum bounds. Splitting took place along the direction of greatest variation and thus created a more feature point preserving simplification. Pauly et al. (2002) found the hierarchical approach to be most accurate out of the two. Accuracy overall was not a strong point, instead the advantages of these approaches were confined to increased speed and improved memory efficiency.

2.2. Simplification by generating subsets

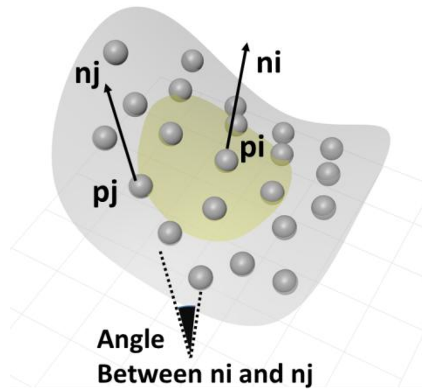
Subset simplification algorithms are either feature point preserving, uniform or a combination of the two. Linsen (2001) proposed a feature point preserving algorithm that used the sum of the following measures (shown in Figure 1) to quantify the relevance of each point, pi:



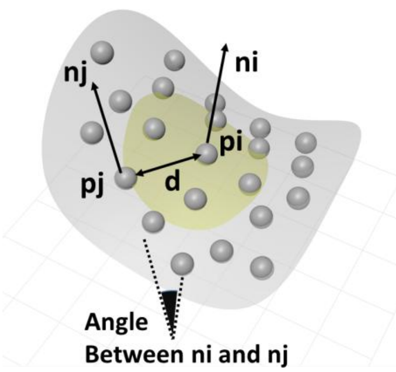
The sum of distances from pi to its k-nearest neighbours.



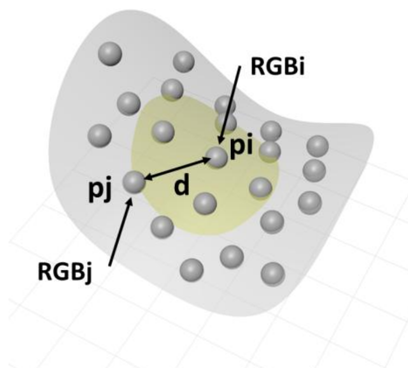
The sum of distances from the tangent plane of pi to its k-nearest neighbours.



The sum of angles between the normal at pi to the normal of its k-nearest neighbours.



The weighted sum of the distance from pi to its k-nearest neighbours. The angle between the normal ni and nj serve as weights.



The sum of RGB distances from pi to its k-nearest neighbours. The colour of a point is recorded as a tuple of three values that represent the respective amounts of red, green and blue light. Each tuple of values is treated as a Cartesian coordinate (xyz) and therefore by measuring the Euclidean distance between two points the RGB distance is found. The shorter the RGB distance the more similar the points are.

Figure 1. Shown are points in a point cloud that describe a surface. Simplification aims to remove those points that are least relevant to the description of the surface. Measures for choosing relevant points are shown in the five panels.

Once each point is assigned a value of importance, all points are ranked in this respect and a percentile of the points is preserved. Linsen (2001) found that had distance to nearest neighbours $M_d(p_i)$ been used alone instead of a sum of the aforementioned measures a forty percent simplification would have resulted in the loss of fine details. As noted by Song and Feng (2009) such approaches are not ideal for point clouds that contain sharp edges, as is the case with airborne lidar point clouds. They presented an algorithm that preserved edge points. Their algorithm used the average distance from a point pi to the estimated tangent planes of its k-nearest neighbours. Once each point had been assigned a value of importance, points with low importance values were progressively removed. Each time a point was removed the normals of its neighbouring points were updated as their neighbourhood configuration had changed. The neighbours of affected neighbouring points importance was also updated. The algorithm then moved onto the next point and continued until the predefined number of simplified points required was reached.

A number of feature point simplification algorithms such as those of Linsen (2001) and Song and Feng (2009) made no attempt to optimize the distribution of non-feature points. For example with regard to Song and Feng (2009) points on planar areas would have very little importance and would all be removed. Song and Feng (2008) proposed an algorithm that explicitly dealt with the distribution of points on smooth surfaces based on a geometric deviation criteria. The algorithm minimized the geometric deviation between the input and output cloud. An objective function was developed to determine the composition of the simplified point cloud, such that the distance from each point on the simplified point cloud surface to the closest cluster of points in the original cloud was minimized. The drawback of this algorithm was that by focusing on the geometric deviation the preservation of topology was not guaranteed.

Moenning and Dodgson (2003) proposed a uniform simplification algorithm that ensured a user-controlled density i.e. a minimum distance between points. A Fast Marching method was used to eliminate points. The point cloud was simplified uniformly and iteratively until the density requirement was met. The user-controlled density of the uniformly distributed output point set was guaranteed, this meant the point set was sufficiently dense and suitable for certain further processing ,should the need develop. This was unlike Linsen (2001) whereby ,for example, a highly non-uniformly distributed input point cloud would result in a simplified point set of insufficient density, or for example a highly dense input point cloud might result in a highly non uniformly distributed output point set. The algorithm proposed by Moenning and Dodgson (2003) also supported feature sensitive point cloud simplification. This method involved weighting each point, for example by local surface variation. This resulted in more points being concentrated in regions of greater change in curvature. The density condition was still enforced and ensured a uniform distribution of the non-feature areas. An increase in the minimum density increased the uniformity of the simplified point cloud, at the expense of adaptivity.

Wang et al. (2010), like Moenning and Dodgson (2003) also presented a feature sensitive point cloud simplification that uniformly simplified non-feature points. Wang et al. (2010) recognized that in a non-uniform point cloud a value of importance assigned to a point must be defined in relation to the distribution of points, and so the following measures were used to quantify the importance of each point, p_i:

- The average distance to the k-nearest neighbourhood of points
- The curvature at the point, pi: The curvature at the point was determined by principal component analysis on the covariance matrix of the neighbourhood of points
- The normal included angle parameter: The average of the angles between point pi and each of its k-nearest neighbour's normals

Once each point had been assigned a value of importance the point cloud was partitioned into an Octree and the average diagonal length of child nodes computed. This value was defined as the feature threshold. The points were separated into feature points and non-feature points by comparing their feature parameter to the feature threshold. If the importance of a point was greater than a minimum value the point was a feature point. If not it was a non-feature point. The non-feature points were mapped to a sphere and their Cartesian coordinates were converted into spherical coordinates. A second sphere was created and sampled uniformly. The amount of sampling depended on the amount of simplification that was required e.g. fifty percent. At each point on this new uniform sphere the closest point on the non-feature point sphere was found and marked. All the marked points on the non-feature point sphere were inversely spherically sampled, and then added to the feature points. This was the simplification result.

3. Method

The above mentioned algorithms were developed using airborne point clouds. The programming language used is python 2.7. The packages sklearn (nearest neighbour computations), network x (graph computation) and numpy (efficient array computations) are used. The method followed in achieving the desired simplification can be broken down into the following three steps:

3.1. Point normal determination and assignment

The first step is to determine the normal at each point. This is achieved by fitting a plane to a point's k-nearest neighbours. The normal of the plane is then assigned to the point. Least squares is commonly used to fit a plane to a set of points, however this method is potentially slow for large point clouds (Soudarissanane et al., 2007). Instead, planes were fitted using the faster Principal Components Analysis (PCA). Here the Eigen vector corresponding to the smallest Eigen value is the normal. The next step is to ensure the normals are consistently orientated. The normals computed by PCA are un-orientated. In other words some may point inside the surface and some outwards. If the measure of information content used in the simplification algorithm is dependent on the normals of a point's neighbourhood it is important to ensure that all the normals point outward.

However this algorithm does not assume any structure of the data points. Therefore by developing a single algorithm that solves the problem of general point cloud simplification the same algorithm can be used to simplify any point cloud other than lidar i.e. terrestrial scans.

To ensure a global consistent orientation a graph is created in which edges are added between all points and neighbours. The edges between any two points j and k are weighted:

$$We_{(j,k)} = 1 - \left|N_j^T N_K\right| \qquad [1]$$

The point normal at j and k are given by N_j and N_k. Therefore larger weights are assigned to points with diverging normals. Next a weighted Minimum Spanning Tree is determined. The minimum spanning tree traces the route of least curvature i.e. the sum of edge weights traced is the minimum possible whilst going through each point. Finally the Minimum Spanning Tree is Depth-first searched. Depth-first searching traverses the tree starting at the point with the highest z value (this point is forced to point towards the positive z direction) and explores as far as possible before backtracking. If the current point p1 is assigned an orientation of n1 and the next point to be traversed is p2, and if $n1*n2 < 0$, then n2 is replaced by -n2. The outcome is a globally consistent outward normal orientation.

3.2. Assign value of importance to each point

Points in the cloud have varying degrees of importance in describing the surfaces in the cloud. Points on flat surfaces are the least important and those in areas of strong curvature are the most important. Points with a higher degree of importance are more likely to be preserved in the simplification. For each point a value of importance is assigned based on one of the following measures that have been suggested from past research:

1. Sum of distances from a point to its k-nearest neighbours. (Linsen, 2001)
2. Curvature at a point. (Wang et al., 2010)
3. The sum of angles between the normal of a point and the normals of k-nearest neighbours. (Linsen, 2001)
4. Average distance from a point to the estimate tangent planes of its k- nearest neighbours. (Song and Feng, 2009)
5. An equally weighted sum of the aforementioned measures.

3.3. Simplify

Initially a random ten percent of the points are selected and marked for preservation (preserved set of points). Thereafter a random ten percent of the remaining points are selected (seed points). At each seed point a radius is searched depending on the points importance. If a point marked for preservation is found within the radius searched then the seed point is removed from the cloud. If no such point is found then it is marked for preservation. Thereafter a further random ten percent of the remaining points is chosen as seed points. This process is repeated until there are no points left. The higher the points importance the smaller the radius searched and thus the smaller the chance of a point that has been marked for preservation appearing and the seed point being marked for deletion. This relationship between importance and radius searched is here user defined as being parabolic and can be seen in the graph in figure 2.

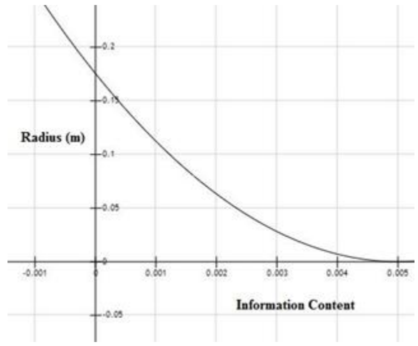


Figure 2. Graph showing the user defined parabolic relationship between the information content

The maximum radius searched:

- Ensures a minimum point density globally
- Is equal to the point spacing in areas of low importance i.e. planar areas.
- Controls the amount of simplification. A decrease in the maximum radius searched leads to lower maximum distance between neighbouring points, a higher minimum density and therefore lesser amount of simplification. This is shown in figure 3.

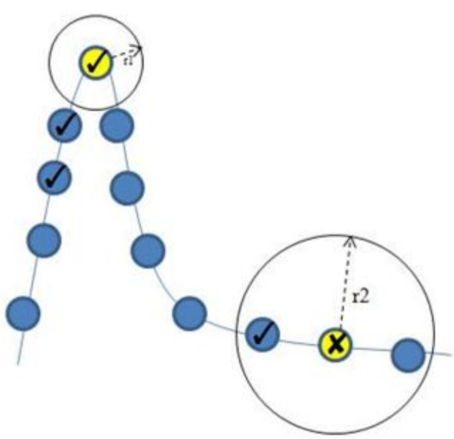


Figure 3. The yellow seed points are randomly selected. The point on the left has higher information content than the point on the right. Therefore a smaller radius is searched. In this radius there are no other points (blue) that are marked for preservation and so the point is marked for preservation (X). The opposite holds for the point on the right and it is marked for deletion.

The benefit of the simplification algorithm is not solely a reduction in computational effort. This is a natural response to reducing the size of a point cloud. Further value lies in having an algorithm that reduces points while consistently preserving the form of surfaces, especially in airborne point clouds where there are many discontinuities ,for example at rooflines and roof edges. This makes it possible to predict/quantify the fidelity of point cloud simplification and the effect it has throughout the point cloud processing/application pipeline ,for example the efficiency of point cloud registration may be improved as the removal of redundant points causes features on point clouds to be more pronounced and easier to match.

4. Results

Ten simplifications of a lidar point cloud named FUZA containing 58153 points were performed for each measure of information (figure 4).They were computed by varying the radius searched so that an equal spread of simplifications, ranging from ten to ninety percent, was achieved. The amount of nearest neighbours chosen depends on the density of the point cloud, which differs throughout. Six nearest neighbours was chosen experimentally. Visually determining the effect for each measure of information theory was difficult as they all appeared very similar and so a quantitative approach was used.

The accuracy of the resulting simplifications was quantitatively computed by using a least squares best fitting plane method. This involved, for each point in the reference (simplified) point cloud, fitting a plane to the points nearest neighbours and computing the distance to the point nearest to the plane on the compared original point cloud. The accuracy of simplifications can be seen in the graph in figure 5. For simplifications between 10 and 50 percent the standard deviation of distances and mean distance for all measures remain rather constant at around 7 and 3 cm respectively. For simplifications between 50 and 80 percent the standard deviation of distances and mean distance using curvature as a measure of information theory was the lowest in relation to the other measures.

The algorithm is computationally exhaustive due to the progressive nature of simplification. The nearest neighbour package used was the biggest run time constraint. The simplification of FUZA (58153 points) ran on average at a speed of 5.38 points per second, taking three hours to complete. The machine had a Microsoft Windows Experience Index rating of 7.4 out of 7.9 for the RAM and processor.

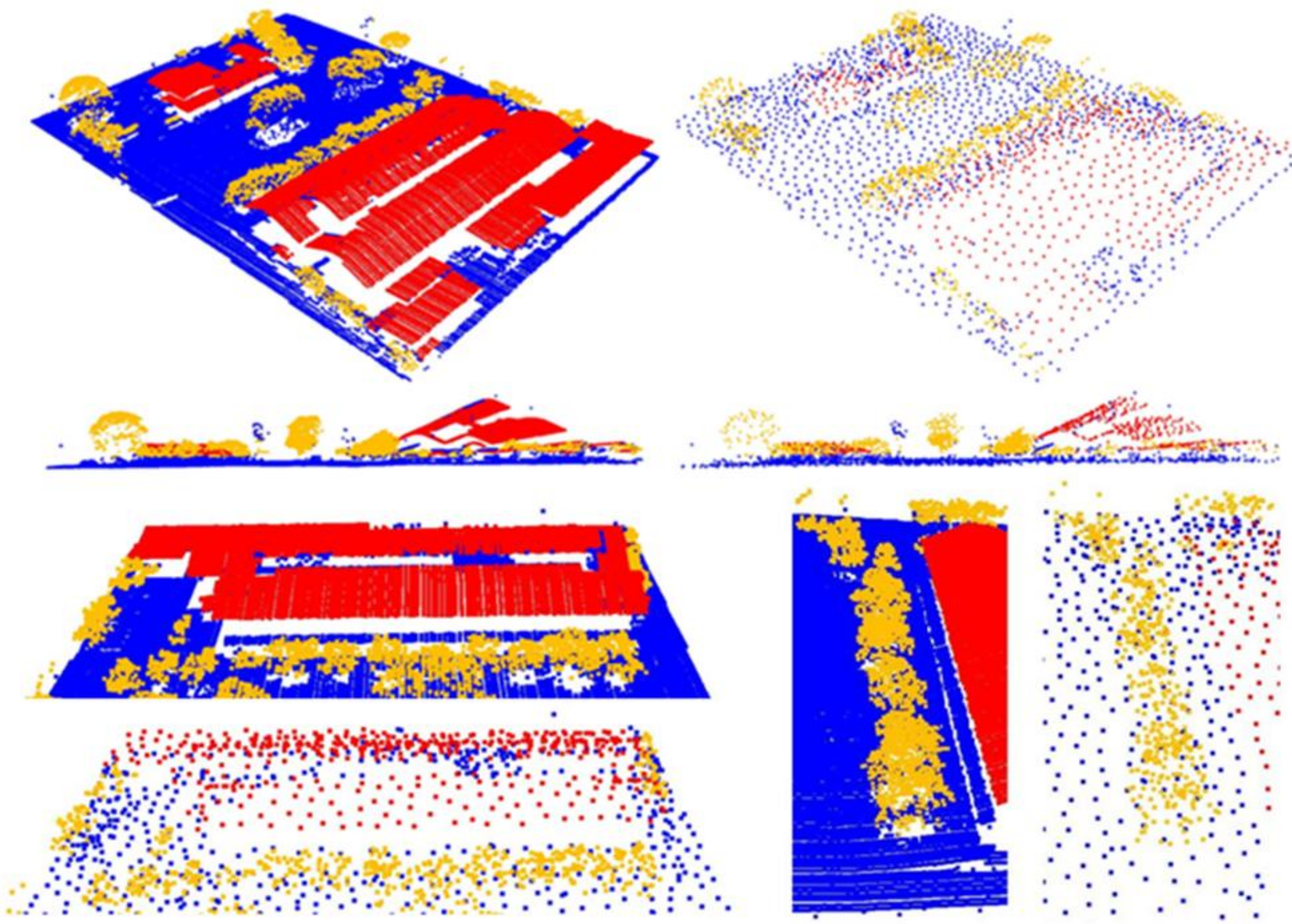


Figure 4. FUZA point cloud before (58153 points) and after a 95% simplification (3081 points) by Curvature. Mean Distance - 0.31m & Standard Deviation of Distances - 0.41m

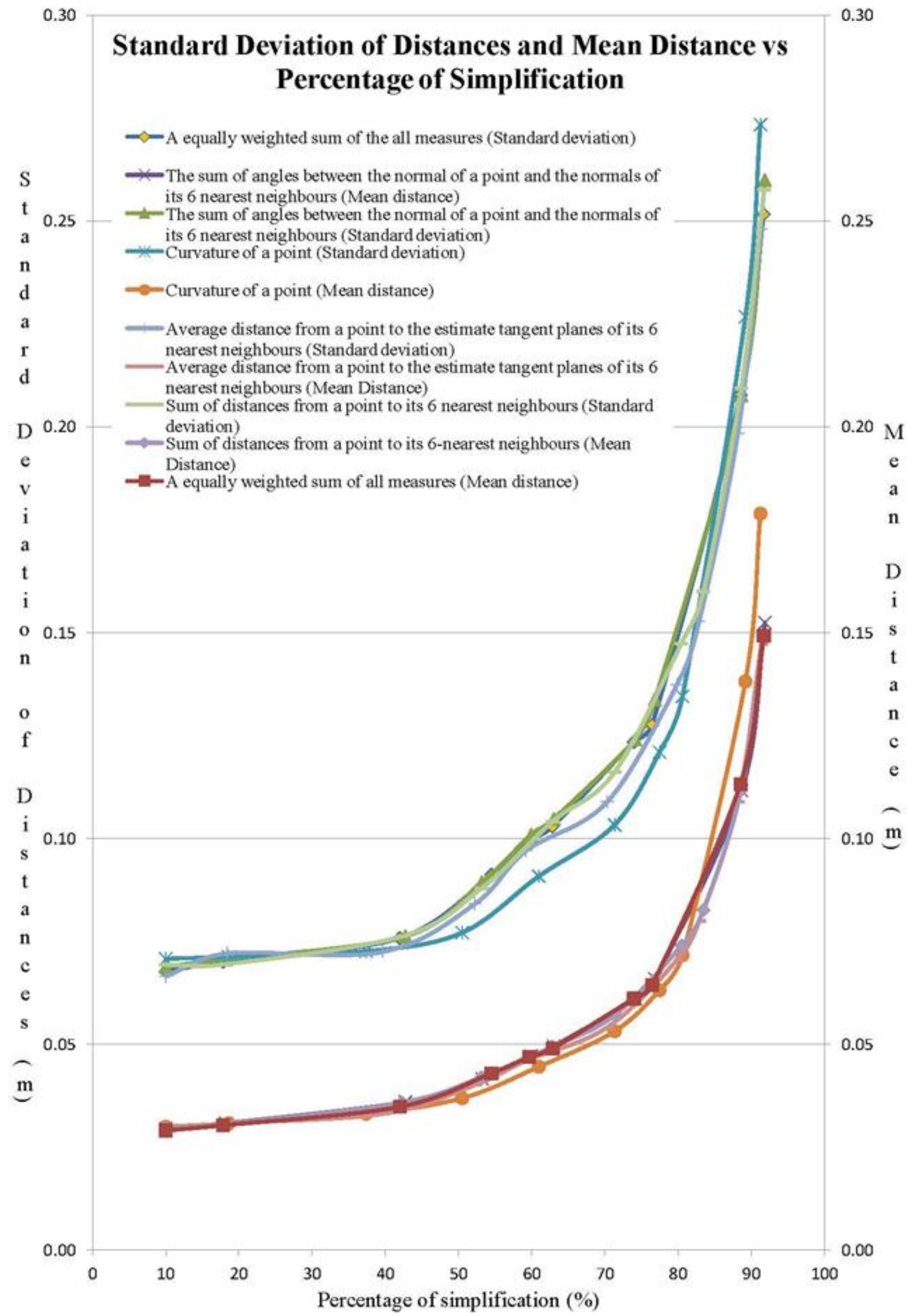


Figure 5. Percentage of simplification (For all measures of information content) vs. Standard Deviation of Distances and Mean Distance

5. Discussion

The measures of information that led to the poorest accuracy of simplification were those that relied on the direction of normals, namely the sum of angles between the normal of a point and then normals of its 6 nearest neighbours and the equally weighted sum of all measures. Although the construction of planes about the neighbourhood was relatively simple by PCA the selection of their orientations so as to define a globally consistent orientation of the surface was a major obstacle. The orientation was propagated by the algorithm described in section 3.1 above. This algorithm has the effect of propagating the orientation along directions of low curvature in the data, and largely avoids ambiguous situations that would be encountered by trying to propagate the orientation across sharp edges and corners. This is because in order to propagate over sharp edges and corners an infinite sampling density requirement is required.

There are often sharp edges and corners in lidar data where the orientation propagation will fail. Failures in the point cloud can be seen in figure 5. If the importance assigned to each point depends on the orientation of normal it is possible that a point of low importance such as a planar point is assigned a high value of importance. This would be because of a inconsistently orientated normal appearing in the points neighbourhood. Fortunately in the simplification of the point cloud used the points with an inconsistent normal orientation are feature points such as tree points and edge points and so should have a high importance.

The effect of false normal could be decreased by the assumption that they cannot be facing inwards as in airborne lidar datasets they can only face more or less upwards. However this would assume a structure of points and therefore confine the simplification algorithm to airborne point clouds.

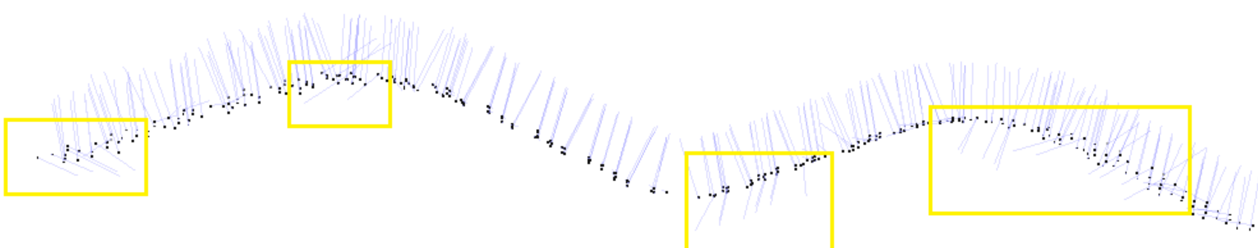


Figure 6. A side view of a roof in FUZA showing point normals. The yellow squares are areas where orientation has failed i.e. the normals are pointing inward.

6. Conclusion

A point cloud simplification algorithm has been presented that preserves feature points and ensures a minimum point density globally. By quantitative and visual analysis the following ranking of information content can be drawn. In order of decreasing accuracy of simplification:

1. Curvature of a point.
2. Average distance from a point to the estimate tangent planes of its k- nearest neighbours.
3. Sum of distances from a point to its k-nearest neighbours.

4. An equally weighted sum of all the measures.
5. The sum of angles between the normal of a point and the normals of k-nearest neighbours.

Future work will be on improving the accuracy of simplification and the reading of accuracy. The accuracy of simplification can be improved by firstly improving the selection of seed points by moving away from a random selection to a more planned selection. By selecting points at the edges of features (e.g., rooflines) as seed points the accuracy of simplification will be improved as edge points are integral to the geometry of scanned objects. Currently if the nearest point to an edge point is selected as an initial seed point and marked for preservation, the neighbouring edge points chances of remaining are small and thus the accuracy of simplification is weakened. Secondly the accuracy of simplification can be improved by replacing points marked for deletion by a single new point.

The reading of accuracy can be improved by measuring the nearest neighbour distances from each point in the original cloud to a mesh of the simplified cloud instead of locally modelled planes. Furthermore it would be interesting to compare the accuracy of this simplification algorithm to mesh-based simplification algorithms such as the Quadric Edge collapse.

7. References

Carsten Moenning and Neil A Dodgson. A new point cloud simplification algorithm. *Proc. Int. Conf. on Visualization, Imaging and Image Processing*, pages 1027-1033, 2003.

Hao Song and Hsi-Yung Feng. A global clustering approach to point cloud simplification with a specified data reduction ratio. *Computer-Aided Design*, 40(3):281-292, 2008.

Hao Song and Hsi-Yung Feng. A progressive point cloud simplification algorithm with preserved sharp edge data. *The International Journal of Advanced Manufacturing Technology*, 45(5-6):583-592, 2009.

Lars Linsen. Point cloud representation. *Univ., Fak. fur Informatik, Bibliothek*, 2001.

Lihui Wang, Jing Chen, and Baozong Yuan. Simplified representation for 3d point cloud data. *Signal Processing (ICSP), 2010 IEEE 10th International Conference*, pages 1271-1274. IEEE, 2010.

Mark Pauly, Markus Gross, and Leif P Kobbelt. Efficient simplification of point-sampled surfaces. *Proceedings of the conference on Visualization'02*, pages 163-170. IEEE Computer Society, 2002.

Sylvie Soudarissanane, Jane Van Ree, Alexander Bucksch, and Roderik Lindenbergh. Error budget of terrestrial laser scanning: influence of the incidence angle on the scan quality. *Proceedings 3D-NordOst*, pages 1-8, 2007.