

HEURISTICAL FEATURE EXTRACTION FROM LIDAR DATA AND THEIR VISUALIZATION

Suddhasheel Ghosh and Bharat Lohani

Geoinformatics Laboratory, Department of Civil Engineering
Indian Institute of Technology Kanpur
Kanpur, 208 016 India
suddhasheel@gmail.com, blohani@iitk.ac.in

Commission V/3

KEY WORDS: LIDAR, segmentation, feature extraction, visualization

ABSTRACT:

Extraction of landscape features from LiDAR data has been studied widely in the past few years. These feature extraction methodologies have been focussed on certain types of features only, namely the bare earth model, buildings principally containing planar roofs, trees and roads. In this paper, we present a methodology to process LiDAR data through DBSCAN, a density based clustering method, which extracts natural and man-made clusters. We then develop heuristics to process these clusters and simplify them to be sent to a visualization engine.

1 INTRODUCTION

Over the last decade, **Light Detection And Ranging** (LiDAR) has become one of the most significant technologies used in the field of topographic mapping. As on date, LiDAR is a standard for capturing object surface information for 3D modelling of terrains. Although, these systems provide detailed valuable geometric information, the data still requires interpretation for object extraction and recognition for its use for the purpose of mapping and/or modelling.

Owing to the inaccuracies inherent in the LiDAR data (Habib, 2009), which are due to its components namely the GPS, INS and the laser ranging system and other factors like flight planning, flying conditions, atmospheric effects, terrain undulation and vegetation cover, the extraction of features is not a trivial task. Extraction of features from LiDAR data has been studied extensively in the past decade (Brenner, 1999, Haala and Brenner, 1999, Brenner, 2000, Forlani et al., 2003, Rottensteiner, 2003, Fujisaki et al., 2003, Haala and Brenner, 1997, Hofmann et al., 2003, Rottensteiner et al., 2005, Clode et al., 2007, Bretar, 2008). The study of literature strongly indicates that the focus of these studies have been on specific types of features. In this paper, we propose a novel approach for processing LiDAR data using clustering techniques for extraction of features which would be followed by a development of heuristics, to help generalize and direct the features to a visualization engine. This research is an extension of an earlier work done by the authors (Ghosh and Lohani, 2007) in order to devise a method for quickly processing and visualizing LiDAR data.

2 CLUSTERING METHODS

Data clustering is known by different names e.g. segmentation, data mining, etc. The main goal is to partition a given set of data points into disjoint subsets. Clustering methods can be divided into many groups namely, partition based, hierarchical, manifold based, evolutionary, fuzzy and hybrid.

Partition based methods like ISODATA, *k*-means or *k*-medioid algorithms take an input *k* for the number of clusters to be generated. Manifold based methods determine only a particular form

of surface in the dataset. Evolutionary methods using genetic algorithms, artificial bee colony, and particle swarm optimization technique have been efficiently used in clustering datasets. Fuzzy *c*-means and its extensions namely fuzzy ISODATA, Gata-Gava, fuzzy *c*-shells, fuzzy *c*-spherical shells, fuzzy *c*-ellipsoidal shells etc utilize an optimization procedure tuned to particular shapes of clusters (Baghshah and Shouraki, 2008).

Density based clustering methods like DBSCAN (Ester et al., 1996), OPTICS (Ankrest et al., 1999), DENCLUE (Hinneburg and Gabriel, 2007) etc. are capable of detecting arbitrary shaped clusters in spaces of any dimension. LiDAR data clusters have arbitrary shapes and therefore density based clustering methods are applicable for segmenting LiDAR data. Literature on clustering suggests that amongst the density based clustering techniques, DBSCAN has been very widely studied, implemented and extended. We have therefore chosen DBSCAN as the clustering algorithm for our study related to LiDAR data.

3 OBJECTIVE

In this paper, we draw out a complete pipeline to mine, extract and generalize specific types of features using clustering of LiDAR data and heuristical techniques which we design and demonstrate in the process. The clusters are generalized via the heuristic to enable redirection to a visualization engine.

4 STUDY AREA

In 2004, Optech Corporation conducted a LiDAR flight over the Niagra Falls with the ALTM sensor. The airplane flew at an average height of 1190m, with a DSS 301 SN0039 camera on board for the aerial photographs. The average density of the LiDAR data is 4.85 points per square metre, where all the returns are considered. Subsets of 100m×100m, were chosen based on the presence of features like gabled-roof houses, trees, and the ground. Results in this paper are presented for one of these subsets. The terrain for the given subset is flat.

5 METHODOLOGY

5.1 Mathematical notations and definitions

Let $S = \{\mathbf{P}_i(x_i, y_i, z_i), i = 1, \dots, N\}$ denote a set of three dimensional data points with cardinality N . Let x_{min} , x_{max} , y_{min} , y_{max} , z_{min} , and z_{max} denote the minimums and maximums of the x , y and z coordinates respectively. $\Delta(S)$ denotes the two-dimensional Delaunay triangulation (DTRI) of the set S . The longest edge of the triangle $\Delta \in \Delta$, is denoted by $\ell(\Delta)$. Given a Delaunay triangulation $\Delta(S)$ and a threshold τ , a **trimmed Delaunay triangulation** (TDTRI) of S is denoted by

$$\Delta_T^{(\tau)}(S) = \{\Delta \in \Delta | \ell(\Delta) \leq \tau\}.$$

Given a DTRI or TDTRI Δ , the set $\Psi(\Delta(S))$ is called the “outline” of the DTRI or TDTRI $\Delta(S)$. In an outline, each of the triangles are broken into edges and then the duplicate edges are removed. The remaining edges are then joined back to form the outline.

$\nabla(S)$ denotes the three-dimensional Delaunay tetrahedralization (DTET). The longest edge of the tetrahedron $\nabla \in \nabla$ is denoted by $\mathcal{L}(\nabla)$. Given a Delaunay tetrahedralization ∇ and a threshold η , a **trimmed Delaunay tetrahedralization** (TDDET) of S is denoted by

$$\nabla_T^{(\eta)} = \{\nabla \in \nabla | \mathcal{L}(\nabla) \leq \eta\}.$$

Given a TET or DTET ∇ , of a set of data points S , the set $\Phi(\nabla(S))$ is called the “peel” of the DTET or TDDET $\nabla(S)$. In a peel, each of the tetrahedrons are broken into triangles and then duplicate triangles are removed.

5.2 Adapting DBSCAN for LiDAR data

Since the LiDAR data has redundant points, we have modified the Euclidean metric as in equation [1].

$$d(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} \text{EuclideanDistance}(\mathbf{p}_1, \mathbf{p}_2) & \mathbf{p}_1 \neq \mathbf{p}_2 \\ 10^{30} & \mathbf{p}_1 = \mathbf{p}_2 \end{cases} \quad (1)$$

5.3 Visual clustering

A visual clustering of the data is carried out using the TerraScan module from TerraSolid Inc. The visually segmented dataset would be used as a benchmark for evaluating the quality of clustering achieved by the DBSCAN algorithm. It would be an interesting exercise to see if the DBSCAN clustering algorithm follows the natural and artificial features in the given dataset.

5.4 Comparing clusters with the actual ones

In DBSCAN, for each of the objects belonging to a cluster its ε -neighbourhood needs to contain at least $minPts$ objects. Ester et al. (1996) have attempted to develop a simple and effective heuristic to determine the parameters ε and $minPts$ using sorted k -dist graphs which map each point to the distance of the k -th nearest neighbour. However, this process is rather interactive and the authors recommend the use of the graphical representation of the k -dist graph to help the user estimate the correct threshold. Daszykowski et al. (2001) assume a uniform distribution of points and thus determine the threshold accordingly. Since the LiDAR data do not follow this statistical distribution, we determine the appropriate distance threshold ε , using an experiment and the Adjusted RAND Index (ARI) (Hubert and Arabie, 1985). The ARI has values ranging from 0 to 1, indicating similarities from the least to the exact between two clustering outputs. The clusters generated by DBSCAN are compared to the visually segmented dataset for a number of thresholds and the ARI is determined for each case. The highest value of the ARI gives the best threshold for the clustering algorithm.

5.5 Types of clusters

The various kinds of clusters that are obtained from the clustering are those containing a little number of points but spanning a small area, those containing a few number of points but spanning a large area, those shaped in the form of domes, those potentially containing planar areas and those following the terrain. Owing to the multiple returns present in the LiDAR dataset, a dome shaped cluster not only contains points at its periphery but also inside it. Each of these kinds of clusters would have to be given a different treatment for final generalization and visualization. The various types of clusters that could be obtained from the clustering process are given in figure [1]. In the next subsection, we develop the heuristics for treating the various types of clusters in the dataset.

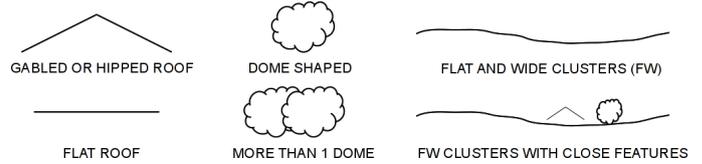


Figure 1: Types of clusters

5.6 Heuristics

The following definitions are introduced in order to develop and understand the heuristics used for the processing of LiDAR data.

Given a set of data points S , with N points, the centroid of the set is given as

$$\mathbf{P}_c \left(\frac{\sum_{i=1}^N x_i}{N}, \frac{\sum_{i=1}^N y_i}{N}, \frac{\sum_{i=1}^N z_i}{N} \right).$$

Points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and \mathbf{P}_4 are defined as $\mathbf{P}_1 : (x_{min}, y_{min}, z_{min})$, $\mathbf{P}_2 : (x_{min}, y_{max}, z_{min})$, $\mathbf{P}_3 : (x_{max}, y_{min}, z_{min})$, and $\mathbf{P}_4 : (x_{max}, y_{max}, z_{min})$. Let $\mathbf{P}_1^c, \mathbf{P}_2^c, \mathbf{P}_3^c$ and \mathbf{P}_4^c be the points closest to $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and \mathbf{P}_4 respectively. Let \mathcal{P} denote the least squares plane for the points $\mathbf{P}_1^c, \mathbf{P}_2^c, \mathbf{P}_3^c$ and \mathbf{P}_4^c . Then the distance of the point \mathbf{P}_c , from the plane \mathcal{P} for the set S is denoted by $\delta(S)$ and is called as the **apex distance** for the set S . Let w and b be the length and breadth of the minimal rectangle surrounding the projection of the points of S into the XY plane. The ratio defined as

$$AR(S) = \frac{\min(w, b)}{\max(w, b)}$$

is called as the aspect ratio for the set S . S is called wide if $\min(w, b) \geq W$ where W is a given threshold. S is called flat if $\delta(S) < f$ where f is a given threshold. It w, b and h denote the width, breadth and height of the minimal cuboid containing the points of S . Then

$$\rho(S) = \frac{\#(S)}{w \times b \times h}$$

denotes the point density per unit volume for the set S . If $\rho(S) \leq \rho_1$ or $\#(S) < M$, then this set of points is called as sparse, where ρ_1 is given threshold in terms of density, and M is a threshold in terms of number of points. If $\rho_1 \leq \rho(S) \leq \rho_2$, where ρ_1 and ρ_2 are given thresholds, the point set S would be referred as a dome shaped cluster. For the purpose of this work, the thresholds ρ_1, ρ_2 and M are determined empirically, through a visual study of the clusters generated. It can be however said that ρ_1 and ρ_2 depend on a priori information from the data and area under study, like scanning frequency and the features present on the terrain.

Firstly, each of the clusters generated by the clustering algorithm are checked for sparseness. The sparse clusters are not considered for further processing. The remaining clusters are checked for satisfying the flatness and wideness conditions. The clusters

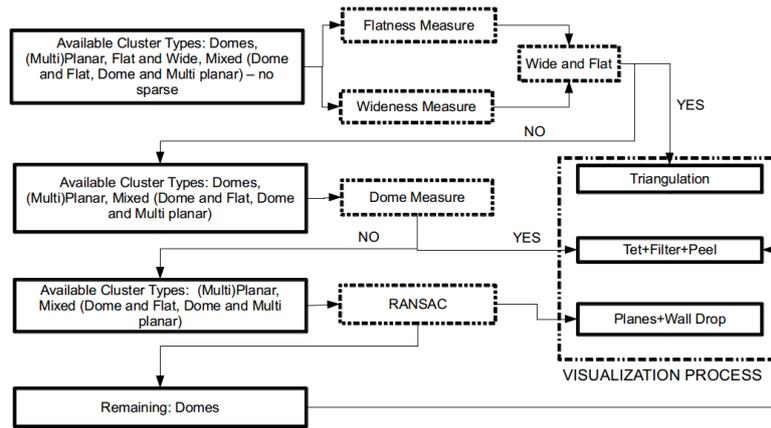


Figure 2: Schema for the processing of clusters

that satisfy these conditions are kept separately in a set FW for later processing by algorithm [1].

Algorithm 1 Treat flat and wide clusters

- 1: **Data:** Set of clusters $\langle C_i \rangle_{i=1}^n$ which passed the flat and wide test
- 2: **return** $\langle \Delta(C_i) \rangle_{i=1}^n$

The remaining clusters are now checked for the dome condition, with the user specified inputs ρ_1 and ρ_2 . The clusters that satisfy the dome condition are kept separately in a set D for being processed later by algorithm [2].

Algorithm 2 Treat dome shaped clusters

- 1: **Data:** A detected dome-shaped cluster C , and threshold η .
- 2: **return** “Peel” of the dome shaped cluster
- 3: Calculate $\nabla(S)$.
- 4: Calculate $\nabla_T^{(\eta)}$, thus containing n tetrahedrons.
- 5: Calculate Δ containing the triangular facets of the tetrahedrons in $\nabla_T^{(\eta)}$.
- 6: Eliminate the facets which are repeated twice in the above set. $\Phi(\nabla)$ is thus calculated.

Algorithm 3 Treat Planar Clusters

- 1: **Data:** A detected planar cluster C , plane fitting thresholds p and m , triangulation threshold τ , and baseheight h .
- 2: **return** Roofs with dropped walls.
- 3: $\langle \mathcal{P}_i^c \rangle = \text{RANSACOnCluster}(C, p, m)$, $i = 1, \dots, n$, where n is the number of detected planes
- 4: **for** $i = 1$ to $i = n$ **do**
- 5: $S = \langle \mathbf{p}_j \rangle_{j=1}^k \in \mathcal{P}_i^c = \text{Project} \langle \mathbf{p}_j \rangle_{j=1}^k \in \mathcal{P}_i^c$ to the respective plane \mathcal{P}
- 6: Calculate $\Delta(S)$
- 7: Let $\Delta'(S) := \Delta_\tau^r(S)$
- 8: Calculate $\Psi(\Delta'(S))$
- 9: Drop the walls to the baseheight h
- 10: **end for**

The remaining clusters are supposed to be potentially containing planes. The detection of the planes are done using the RANSAC algorithm proposed by Fischler and Bolles (1981). We devise our RANSAC based processing close to Bretar (2008), but however restrict the accepted models so that it has at least n number of points close to it. A certain number of points are rejected during the RANSAC process. In case these points are placed closely,

and exceed a certain number, we use triangulation before directing them to the visualization engine. The points belonging to the respective model are projected to the plane. The outlines of these planes are found out and walls are “dropped” to a base height. The algorithm for processing the clusters potentially containing planes is given in Algorithm [3]. The schema for the heuristics of processing the clusters is shown in figure [2].

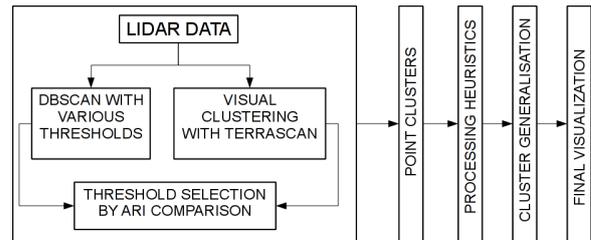


Figure 3: The processing pipeline for the LiDAR datasets

5.7 Summary of the pipeline for data processing

In our processing pipeline, a visual segmentation of the given dataset is conducted using the TerraScan module. Clustering of LiDAR data is conducted using DBSCAN for various thresholds and an ARI based comparison is made with the visually segmented dataset. The highest value of ARI would give us the best distance threshold for clustering. Each of the clusters thus obtained is passed through the rule base for generalization and redirection to the visualization engine. A graphical representation of this process is given in figure [3].

6 RESULTS AND DISCUSSION

We have used a visually segmented dataset as a benchmark for evaluating the performance of the clustering algorithm. The manually classified dataset had 31 clusters.

6.1 Identification of best thresholds for clustering

A pre-visualization of the LiDAR data indicated that if the ε -neighbourhood of a point consisted of less than three points, then we could label the point as an outlier. Thus, the *minPts* criteria for the points was chosen as 3. To determine the best distance threshold for DBSCAN, an experiment was conducted where ε was varied from a very low value of 0.7 meter to a high value of 4.0 meters. For each of the values of the distance threshold, the Adjusted RAND Index (ARI) was found out to compare the performance of the clustering algorithm for the given dataset with respect to the manually clustered dataset. The performance graphs

for these thresholds v/s the ARI is shown in figure [4]. The outputs of the DBSCAN algorithm with the best threshold are shown in figure [5].

Parameter Type	Parameter Name	Value
Sparseness	ρ	0.01
Wideness	Cardinality	600
	minLength	80m
	Aspect Ratio	0.7
Flatness	Apex To Base Distance	2.0m
Tetrahedralization	Trim (η)	2.5m
	Density	4

(a) Selected Parameters for testing heuristics

	Sparse	Flat and Wide	Dome-shaped	Planar
Sparse	5	0	0	0
Flat and Wide	0	5	0	0
Dome-shaped	0	0	15	1
Planar	0	0	1	13

(b) Contingency results for heuristics

Table 1: Parameters and contingency tables for testing heuristical data processing

6.2 “Peel”ing a dome shaped cluster

As described in Algorithm [2], a dome shaped cluster extracted from the clustered dataset, is “peel”ed by a process of using tetrahedralization followed by trimming of the triangular facets of the tetrahedrons. Figure [6b] shows the peel of a specific dome shaped cluster.

6.3 Determination of the thresholds for RANSAC algorithm

The RANSAC algorithm requires a distance threshold ε for fitting the planes and the number of draws from the set of points that one should make to find out the best model. Bretar (2008) gives a method for finding out the number of draws. This approach was modified by adding another threshold m , which specifies that the model drawn out by the RANSAC algorithm had to conform to at least m points from the dataset. It was decided by observation in the dataset, that the minimum size of the plane to be detected had to be at least $4m \times 4m$. Therefore M can easily be determined by multiplying this area by the areal density of the dataset.

An experiment was conducted to find out the best threshold for RANSAC by taking out some planar clusters and then varying

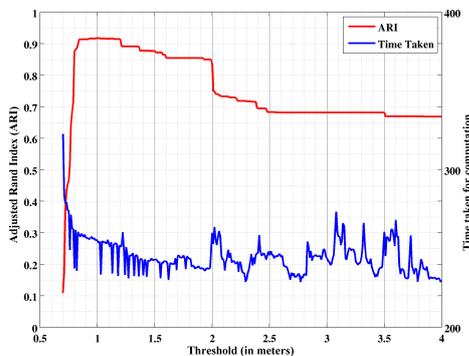


Figure 4: ARI Performance graphs for DBSCAN

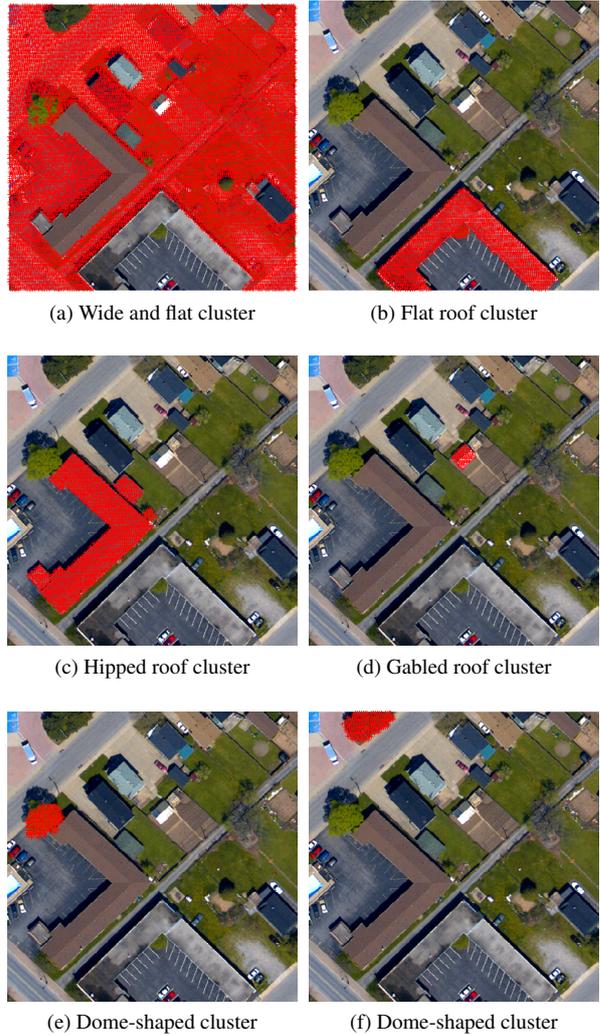


Figure 5: Outputs of DBSCAN and various cluster types

the thresholds from 0.01 to 1.00 in steps of 0.01. It is seen that in most of the cases, the best thresholds for determination of the planes lies between σ_h and $1.5\sigma_h$, wherein σ_h denotes the vertical error in the LiDAR dataset. Two detected buildings using potentially planar clusters are shown in figures [6d] and [6f]. Two graphs for determining the best threshold for RANSAC are shown in figure [7].

6.4 Efficacy of the heuristics

5 different subsets were extracted from the main dataset and each of them were clustered using DBSCAN and the same parameters. 40 clusters were manually extracted and sorted into various groups namely sparse, wide and flat, dome-shaped and potentially containing planes. Each of these clusters were also passed through the heuristics algorithm with a given set of parameters demonstrated in table [1a] which also sorted them into various groups. The efficacy of the heuristics is demonstrated by the contingency table [1b].

6.5 Processing of the dataset and three dimensional restitution

The given dataset is now passed through the pipeline using the thresholds described in table [1a], to generate a simplicial representation of the terrain and the features over it. The number of

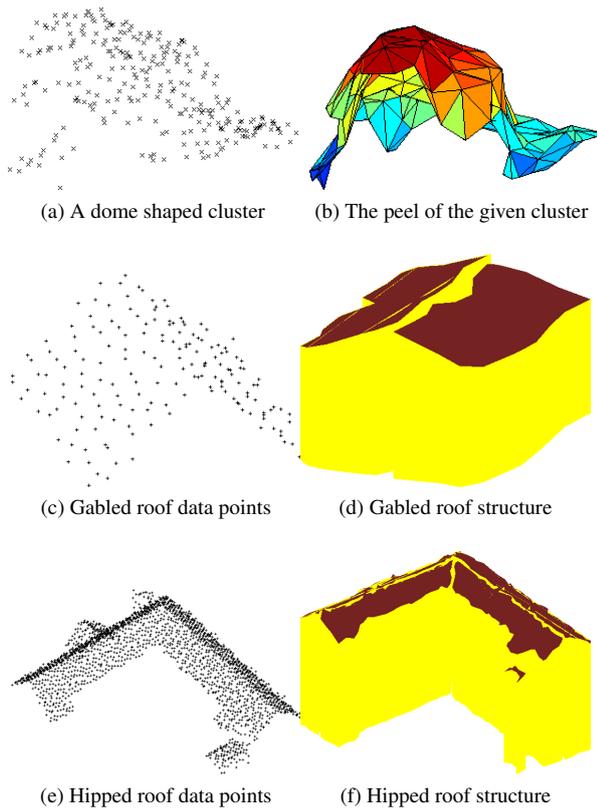
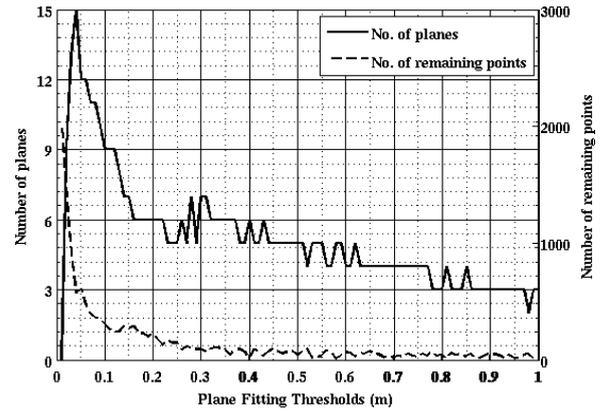


Figure 6: (a) A dome shaped cluster (b) A “peel”ed dome cluster using tetrahedralization and subsequent thresholding of the triangular facets. The colors of the triangles vary from dark blue to dark red based on the elevation of the centroid of the triangles (c) Data points for a gabled roof (d) The gabled roof structure generated by the pipeline (e) Data points for a hipped roof (f) The hipped roof structure generated by the pipeline.

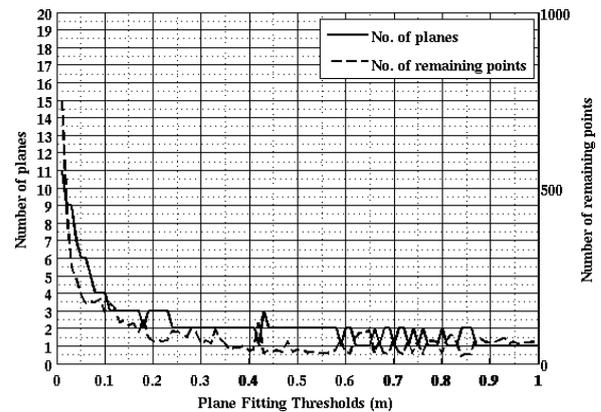
points in the clusters generated by DBSCAN vary from a very small number, in the order of 10 or less, to a very large number to the scale of 800 or more. To make the processing faster, it was empirically decided that clusters containing less than 100 points would not be processed through this heuristic process, and would be ignored.

The features thus generated by the heuristic process are converted into simplices and polygons with 3D vertices and stored into a file. This file is then read using a program developed using the OpenSceneGraph API and C++. OpenSceneGraph is a tool widely used for visualization of 3D data. The authors have used this tool for immersive visualization of the simplices and polygons generated by the algorithm. The 3D landscape model is also draped with the geocoded aerial photograph of the study area. The complete procedure of clustering, followed by heuristics and three dimensional restitution took about 600 seconds of time on a 1.5 GHz, 64 bit Kubuntu Linux Machine with 2GB RAM. Two different views of this 3D landscape model in anaglyph mode are shown in figure [8].

In the earlier attempt (Ghosh and Lohani, 2007) for near-realistic three dimensional visualization of the LiDAR data using triangulation and tetrahedralization, it was noted that both these approaches lead to the loss of features and increased the complexity of computation. This paper, has extended the earlier work to present a hybrid approach using triangulation or tetrahedralization based on the decision made by the heuristical process.



(a) Case 1



(b) Case 2

Figure 7: Threshold determination process for the RANSAC algorithm. The correct threshold seems to settle in between σ_h and $1.5\sigma_h$

7 CONCLUSION

This paper presented a density based algorithm DBSCAN (Ester et al., 1996) which was adapted for use with LiDAR data with a modified Euclidean metric for calculation of distances. The results show that the algorithms have demonstrated almost satisfactory clustering results, thus yielding clusters which are close to the ground features. The thresholds used in this study were determined by running several experiments and thus choosing the best threshold from existing statistical measures. A study on the analytical determination of the thresholds automatically from the parameters of the lidar data viz. pulse repetition frequency, flying height, flying velocity etc is currently in progress. We also demonstrated a heuristic based approach for processing the various kinds of clusters obtained through the process of clustering. It is seen that triangulation of the entire dataset essentially ends up in the loss of three dimensional details and a tetrahedralization makes the computation too complex. The heuristic based approach has therefore enabled us to pursue a hybrid approach where both triangulation and tetrahedralization have been used for processing of the clusters. This study was conducted on a small subset of a comparatively larger dataset, for testing feature extraction. For larger datasets, it will be imperative to implement spatial indexing by R*-trees and tiling to test the performances of these algorithms on larger datasets and develop a methodology for processing it in a smaller amount of time.

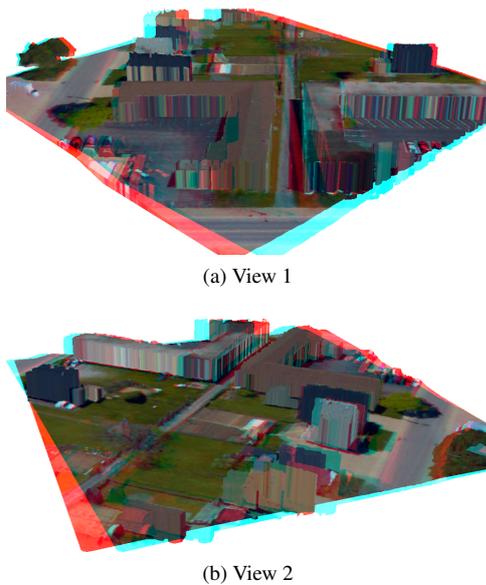


Figure 8: Anaglyph views of the data

References

- Ankrest, M., Breunig, M., Kriegel, H. P. and Sander, J., 1999. OPTICS: Ordering points to identify the clustering structure. In: *Proceedings of ACM, SIGMOD'99*, Institute of Computer Science, University of Munich.
- Baghshah, M. S. and Shouraki, S. B., 2008. A fuzzy clustering algorithm for finding arbitrary shaped clusters. In: *AICCSA*, pp. 559–566.
- Brenner, C., 1999. Interactive modelling tools for 3-D building construction. In: D. Fritsch and R. Spiller (eds), *Photogrammetric Week 99*, Herbert Wichmann Verlag, Heidelberg, pp. 23–34.
- Brenner, C., 2000. Towards fully automatic generation of city models. *International Archives of Photogrammetry and Remote Sensing* 33(3), pp. 85–92.
- Bretar, F., 2008. Feature extraction from lidar data in urban areas. In: J. Shan and C. K. Toth (eds), *Topographic Laser Ranging and Scanning: Principles and Processing*, Taylor and Francis Group, LLC, chapter 14, pp. 422–438.
- Clode, S., Rottensteiner, F., Kootsookos, P. and Zelniker, E., 2007. Detection and vectorisation of roads from lidar data. *Photogrammetric Engineering and Remote Sensing* 73(5), pp. 517.
- Daszykowski, M., Walczak, B. and Massart, D. L., 2001. Looking for natural patterns in data: Part 1. density-based approach. *Chemometrics and Intelligent Laboratory Systems* 56(2), pp. 83–92.
- Ester, M., Kriegel, H., Sander, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, pp. 226–231.
- Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), pp. 381–395.
- Forlani, G., Nardinocchi, C., Scaioni, M. and Zingaretti, P., 2003. Building reconstruction and visualization from LiDAR data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV, Part 5/W12, Ancona, Italy.
- Fujisaki, I., Evans, D. L., Moorhead, R. J., Irby, D. W., Aragh, M. J. M. and Roberts, S. D., 2003. Lidar based forest visualization: Modeling forest stands and user studies. In: *ASPRS Annual Conference*, Fairbanks, Alaska.
- Ghosh, S. and Lohani, B., 2007. Near-realistic and 3d visualisation of lidar data. In: *Proceedings Of ISPRS Joint Workshop on Visualisation and exploration of Geospatial data*, Stuttgart, Germany.
- Haala, N. and Brenner, C., 1997. Generation of 3-d city models from airborne laser scanning data. *EARSEL Workshop on LIDAR Remote Sensing of Land and Sea*, Tallinn, Estonia.
- Haala, N. and Brenner, C., 1999. Extraction of buildings and trees in the urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing* 54, pp. 130–137.
- Habib, A., 2009. Accuracy, quality assurance and quality control of lidar data. In: *Topographic Laser Ranging and Scanning*, CRC Press, pp. 269–294.
- Hinneburg, A. and Gabriel, H.-H., 2007. DENCLUE 2.0: fast clustering based on kernel density estimation. In: *Proceedings of the 7th international conference on Intelligent data analysis, IDA'07*, Springer-Verlag, Berlin, Heidelberg, pp. 70–80.
- Hofmann, A. D., Maas, H. and Streilein, A., 2003. Derivation of roof types by cluster analysis in parameter spaces of airborne laserscanner point clouds. In: *ISPRS Commission III WG3, Workshop, 3-D reconstruction from airborne laser-scanner and InSAR data*, IAPRS International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences Vol.34, Part 3/W13, pp. 112–117.
- Hubert, L. and Arabie, P., 1985. Comparing partitions. *Journal of Classification* 2, pp. 193–218. <http://dx.doi.org/10.1007/BF01908075>.
- Rottensteiner, F., 2003. Automatic generation of high-quality building models from lidar data. *Computer Graphics and Applications*, IEEE 23(6), pp. 42 – 50.
- Rottensteiner, F., Trinder, J., Clode, S. and Kubik, K., 2005. Automated delineation of roof planes from LiDAR data. In: *Laser Scanning 2005*, 221–226, ISPRS WG III/3, III/4, V/3 Workshop.