# HIERARCHICAL CLUSTERED OUTLIER DETECTION IN LASER SCANNER POINT CLOUDS

**S. Sotoodeh**

Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland
Soheil.Sotoodeh@geod.baug.ethz.ch

**Commission V/3**

**KEY WORDS:** Point cloud, Laser Scanner, Outlier detection, EMST, Gabriel graph, Clustering

## ABSTRACT:

Cleaning laser scanner point clouds from erroneous measurements (outliers) is one of the most time consuming tasks that has to be done before modeling. There are algorithms for outlier detection in different applications that provide automation to some extent but most of the algorithms either are not suited to be used in arbitrary 3 dimensional data sets or they deal only with single outliers or small scale clusters. Nevertheless dense point clouds measured by laser scanners may contain surface discontinuities, noise and diffrent local densities due to the object geometry and the distance of the object to the scanner; Consequently the scale of outliers may vary and they may appear as single or clusters. In this paper we have proposed a clustering algorithm that approaches in two steps with the minimum user interaction and input parameters while it can cop with different scale outliers. In the first step the algorithm deals with large outliers (those which are very far away from main clusters) and the second step cops with small scale outliers. Since the algorithm is based on clustering and uses both geometry and topology of the points it can detect outlier clusters in addition to single ones. We have evaluated the algorithm on a simulated data and have shown the result on some real terrestrial point clouds. The results explain the potential of the approach to cop with arbitrary point clouds and different scale erroneous measurements.

## 1 INTRODUCTION

Simple, efficient and direct capturing of 3D information are the main reasons for the fast growing popularity of laser scanners. Although the generated point clouds are direct and dense measurement of objects, the appearance of single or cluster outliers cause serious problems for the next modelling steps. Therefore, a pre-process is required to detect and remove outliers. However, the number of points in the generated point cloud is in the order of million points, so (semi) automatic approaches are necessary.

Outlier detection in point clouds is not a trivial task since there are: geometrical discontinuities caused by occlusions in silhouette boundaries, no prior knowledge of the statistical distribution of points, the existence of noise, and different local point densities. The typical outlier detection approaches are classified as distribution-based, depth-based, distance-based, density-based and clustering approaches (Papadimitriou et al., 2002).

In the previous work (Sotoodeh, 2006), we have introduced an outlier detection algorithm for laser scanner point clouds, which is categorized in density-based approaches, and have investigated the advantages and the deficiencies of the algorithm in different data sets. The algorithm needs a predefined minimum density for inlier clusters and a threshold to distinguish outliers from inlier. There it is shown that even though the algorithm is capable to detect single and small clustered outliers but it simply does not detect clustered outliers that are denser than the predefined cluster density (large $\beta$-error). Also we have tried the algorithm in an iterative manner however it removes a large amount of the inlier and consequently results in a bigger value of $\alpha$-error.

In this paper we have presented a new algorithm that applies more sophisticated information of the point cloud to detect single and clustered outliers with a minimum user interaction. It uses two proximity graphs and performs in two steps. In addition to the algorithm description, the results of applying the algorithm to a real close-range data is reported in this article. Also some implementation issues are discussed.

This article contains a brief review of several outlier detection approaches in section 2. Section 3 presents the algorithm and some implementation issues. Results of applying the algorithm on different data sets is presented and discussed in Section 4 and the last Section concludes the article by discussing the achievements.

## 2 RELATED WORK

While an extensive amount of research has been presented in literature for outlier detection it is still a critical problem in laser scanner point clouds. The proposed approaches have weak potential to perform well with surface discontinuities, they need some priory knowledge of the statistical distribution of the samples (Hawkins, 1980, Vanicek and Krakiwsky, 1982) or they are sensitive to noise and different local densities (Breunig et al., 2000). Nevertheless, the mentioned criteria are typical cases in laser scanner point clouds.

According to (Papadimitriou et al., 2002) outlier detection approaches are classified into the distribution-based (Hawkins, 1980), depth-based (Johnson et al., 1998), clustering approaches (Jain et al., 1999), distance-based (Knorr et al., 2000) and density-based (Breunig et al., 2000). Distribution-based approaches deploy some standard stochastic distribution model (Normal, Poisson, etc.) and flag as outliers those objects that deviate from the model according to a significant level (Vanicek and Krakiwsky, 1982, Rousseeuw and Leroy, 1987, Barnett and Lewis, 1994). However, for arbitrary data sets without any prior knowledge of the distribution of points, determination of the suitable distribution model which fits to the data set (if any) needs to perform expensive tests (in laser point clouds the distribution of points varies according to the distance of objects to laser scanner and the object geometry). Local surface fitting approaches, for instance moving least squares, is also used for outlier detection. The algorithms perform well if the point cloud is dense and obtained from a smooth surface. However, discontinuities or high curvature areas would get severe smoothing effect. The description of these algorithms and their application is beyond the scope of this article.

The depth-based approach is based on computational geometry and computes different layers of k-dimensional convex hulls (Johnson et al., 1998). Objects in the outer layer are detected as outliers. However, it is a well-known fact that the algorithms employed cannot cope with large, arbitrary data sets in 3 dimensions. The above two approaches for outlier detection are not appropriate for large, arbitrary data sets (Papadimitriou et al., 2002). Nevertheless, this is often the case with laser point clouds.

The distance-based approach was originally proposed by (Knorr et al., 2000). An object in a data set $P$ is a distance-based outlier if at least a fraction $b$ of the objects in the object set is further than $r$ from it. This outlier definition is based on a single, global criterion determined by the parameters $r$ and $b$. This can lead to problems when the data set has both dense and sparse regions (Breunig et al., 2000).

The density-based approach was proposed by (Breunig et al., 2000) for KDD (Knowledge Discovery in Database) applications and (Sotoodeh, 2006) adopted the algorithm for application in laser scanner point clouds. It relies on a local outlier factor (LOF) of each object, which depends on the local density of its neighborhood. The neighborhood is defined by the distance to the Mintsth nearest neighbor. The MinPts is a predefined value, which corresponds to the minimum number of points in the calculation of density. The algorithm is not only independent of the prior knowledge of the scanned objects, the distribution or density of sampled points but also does not suffer from the different local point densities. It is capable to detect single and small clustered outliers. Nevertheless it does not detect clustered outliers that are denser than the predefined cluster density (large $\beta$-error).

Many clustering algorithms detect outliers as by-products (Jain et al., 1999). From the viewpoint of a clustering algorithm, outliers are objects not located in the clusters of dataset. These algorithms, in general, consider outliers from a more global perspective, which also has some major drawbacks (Breunig et al., 2000). Clustering algorithms, also called as classification methods, are performing by two main approaches: supervised and unsupervised. In the supervised approach the algorithm needs some representatives of different classes the supervisor expects. Providing such samples differs in various laser data set and so makes the approach very dependent on the scanned objects.

In the unsupervised case, the goal is to cluster the input data in such a way as to provide clusters $C_k, k = 1, ..., K$ which correspond to some underlying (interesting or useful) unobserved class labels. A fundamental difficulty in clustering is determining $K$, the number of clusters. Once $K$ is determined, one proceeds to group the observations. One may approach clustering from a density estimation viewpoint. For instance, a common approach is to model the density as a mixture of $K$ components (again, choosing $K$ can be difficult) and then use these components to determine clusters. A related method is $k$-means clustering. The idea is to cluster the data into clusters centered on k centers. The centers are initialized arbitrarily, and points are assigned to the cluster associated with their closest center. The centers are then recomputed using the assigned points, and this continues until convergence. Besides the problem of selecting the value of $K$, the $k$-means algorithm suffers from sensitivity to the initial cluster centers. For this reason, some practitioners advise trying several initializations, with various methods for selecting or combining the result clusters. Others suggest various methods for selection of initial centers (Marchette, 2004).

The minimum spanning tree can be used for clustering, using a local criterion for defining clusters. This idea is described in some detail in (Zahn, 1971). The idea is to break (remove edges from) the minimum spanning tree at edges that are "inconsistent". This results in a collection of connected graphs, one for each cluster. Many definitions of inconsistent are possible. One could compute the standard deviation of the edge lengths incident on a vertex and eliminate edges which are large relative to this scale. However since this cutting is based on a global criterion, the clustering result would be rough and outliers close to the object surface cannot be detected. This is described in more detail in Section 3.

(Sithole, 2005), has also applied minimum spanning tree to segment airborne laser scanner (ALS) data. The algorithm is scan line based and performs in different directions. The author has reported well performance of the algorithm in different ALS data set to separate terrain, trees, house roofs and bridges (segmentation). The outliers are detected as points that are not in the predefined classes. It has a fast run time performance and runs in case there are overlapping point clouds. However the extension of the algorithm to close-range data, in case either there is no information about scan lines or if the point cloud is a combination of different scan positions (topologically 3D data from object surfaces), does not seem trivial and limits the application to ALS datasets.

## 3 HIERARCHICAL OUTLIER DETECTION (HOD) ALGORITHM

According to the general definition of outliers form (Hawkins, 1980), "Observations that deviate so much from other observations as to arouse suspicion that it was generated by a different mechanism", an outlier in a dense point cloud can be identified using its sampling interval deviation from the others. In laser scanner point clouds the sampling interval is not a fixed value since the sampling is performed based on two fixed angular resolution and objects might have different distance to the measurement instrument and so outliers might appear in various scales in a scan; Therefore applying a global and then a local outlier detection should provide useful results. Based on this observation we have developed an algorithm that runs in two phases. The first phase tries to capture some statistical information of a global sampling interval, while the second phase provides a local criteria to cluster the point cloud. Flowchart of the algorithm is depicted in Figure 1.

First, a rough global approximation of the sampling intervals is estimated over the Euclidean Minimum Spanning Tree (EMST) edges. Then the tree edges that are not in a predefined confidential interval are pruned. The result is a rough clustering of the point cloud. In the next step, each cluster is treated separately. For points in each cluster a graph, so called Gabriel Graph (GG), is generated. Edges of GG are used for estimating the sampling interval statistics in each cluster. Then graph edges that are not in a predefined confidential interval are pruned. This gives the final clustering in which single outliers are removed as a by product. The clustered outliers are also removed if they have less point density than a predefined value.

Initially the algorithm computes the Delaunay triangulation of the point cloud. The underlying topology of the Delaunay graph is the base for the generation of the next graphs. In the first phase of the clustering, EMST of the point cloud is generated and the edges of the tree are pruned based on the statistical analysis of the edge lengths. This gives a rough clustering of the point cloud and might disconnect some big clustered outliers that their distance to

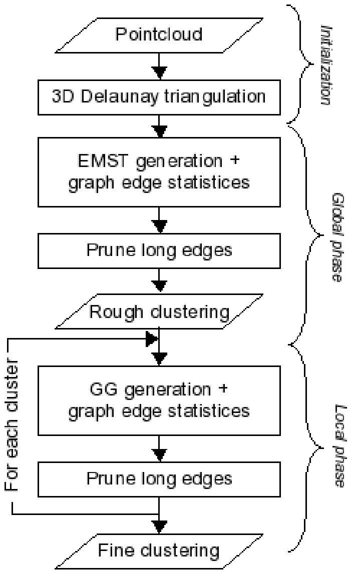

Figure 1: Flowchart of the hierarchical outlier detection algorithm

the other clusters are large. Clusters that are denser than a predefined threshold are kept and the rest are removed. Second phase starts with the generation of GG for the point clouds of each cluster from the last phase. Having pruned the long edges of each GG according to the statistics computed over edges of that GG, a finer clustering of the point cloud is obtained. Removing clusters less denser than a predefined value removes the final outliers and cleans the data. In another viewpoint, the algorithm in the first stage removes relatively large scale erroneous measurements and in the second phase it detects and removes the outliers that might not be as large as the first ones but according to the scanned object surfaces they are considered as wrong measurements. In the following sections, the above process is described in more details.

### 3.1 Global phase (rough clustering)

In the first step we use edges of EMST to obtain a global sampling interval measure. The Euclidean minimum spanning tree or EMST is a minimum spanning tree of a set of points in $R^n$, where the weight of the edge between each pair of points is the Euclidean distance between those two points. In simpler terms, an EMST connects a set of points using edges such that the total length of all the edges is minimized and any point can be reached from any other by following the edges.

This definition gives a clue that edges of EMST contain some global information about the sampling interval, since they span the points by a global minimum edge weight (distance). Additionally in case of some clusters apart from each other, EMST connects them by single edges that are logically longer than the other edges of the tree (Figure 2b).

Having assumed that sampling intervals obey a normal distribution, an edge of the tree is statistically long if its distance to the median of the all edge lengths is longer than the distance corresponding to a predefined confidential interval. Median is used since it is statistically less sensitive to outliers. Removing the long edges of the tree according to such a threshold results in some sub trees that each corresponds to a cluster of points (Figure 2c).

Since today laser scanners provide dense point clouds of objects, splitted clusters that are less dense than a threshold are most probably outliers. In our implementation the minimum inlier cluster density (the threshold) is a user defined single value that might be different for various scanning resolutions and object size.

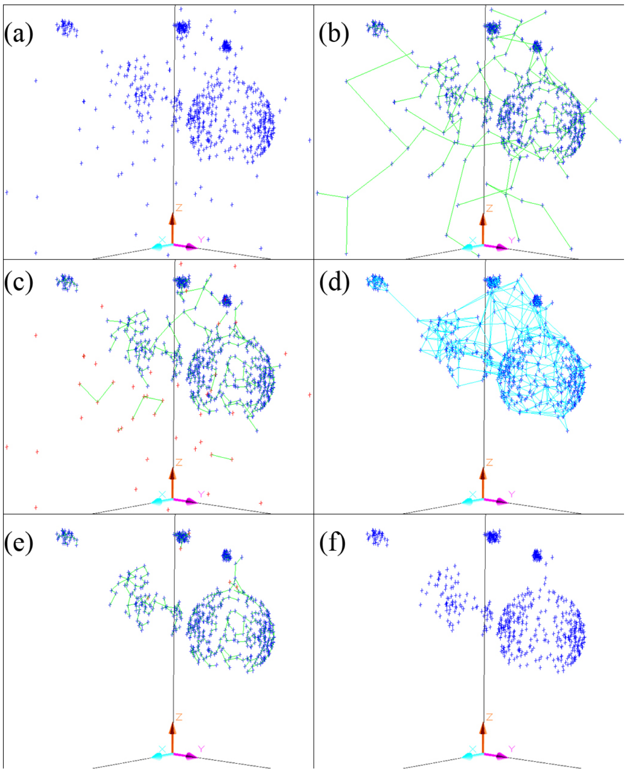

Figure 2: Proposed algorithm steps in a simulated data. Steps (a) to (c) and (d) to (f) illustrate the first and the second phases of the algorithm respectively. (a)input data set (b)EMST of the point set (c) pruned EMST by 99% confidential interval (d) GG of the clusters of the first phase (e) pruned GG by 95% confidential interval (f) ultimate result which is cleaned out of the outliers.

In this stage the algorithm has cleaned outliers according to a global criteria that is performing well in the scale of the whole scan but might not be suitable to remove local outliers. So we need a local and more rich measure of sampling intervals. The next stage describes an approach to reach this goal.

### 3.2 Local phase (fine clustering)

Since EMST provides a rough skeleton of the scanned object, the estimated sampling interval is also not so precise. Applying a denser structure (graph) that has more edges on the underlying scanned surface provides a denser sample of the edges and consequently the estimation of the parameters of the related population is more reliable. Gabriel Graph is such a structure.

Gabriel graphs, also known as Least Squares Adjacency Graphs, were introduced by (Gabriel and Sokal, 1969) and named after their originator. GG has originally been defined for 2D and has been used for geographic variation of data, but the definition is generalized to higher dimensions in a straightforward way (Veltkamp, 1994). It also has widely been applied in the analysis of labeled data (Aupetit, 2003) and widely in boundary/surface reconstruction algorithms; (Veltkamp, 1994, Attene and Spagnuolo, 2000, Adamy et al., 2000) to name a few. A Gabriel Graph on a point set $P$ in $R^n$ is defined to be the graph on $P$ in which two points are connected if the largest open ball passing through the two points is empty. In a three dimensional Euclidean space two points make an edge if the largest sphere passing through these two points contains no other point. On the other hand since

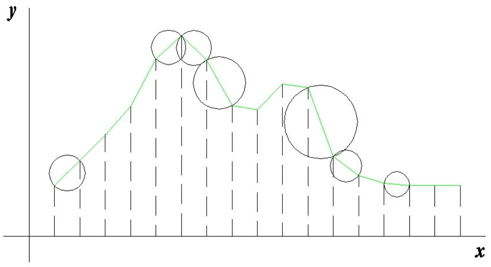

Figure 3: Gabriel graph edges and sampling intervals of a sampled curve in a plane. Two points are connected by GG edges if the largest circle passing through the points is empty (Only some circles are shown in the figure).

GG is a sub graph of each Delaunay triangulation of the point set, the edges of the GG are also edges of each Delaunay triangulation and inherit their properties (Marchette, 2004).

According to the definition, the graph contains edges that resemble the sampling intervals in three dimensions and the structure is quit like a wireframe of the scanned object surface (Figure 2d). Figure 3 illustrates the Gabriel Graph for a sampled curve in a plane. It shows how the edges of GG are similar to the sampling intervals.

Based on the above property the proposed algorithm performs the second phase. For each cluster obtained in the previous stage, GG is computed and its edges considered as the samples of the sampling distance in that particular cluster. Like the first step, the median value of the edge lengths is assumed as the estimation of the sampling distance with a standard deviation equal to the standard deviation of the edge lengths. Considering the predefined confidential interval, long edges of the graph are cut. It results in sub graphs each indicating a cluster. Clusters that have a density less than the predefined cluster size are considered as outliers (Figure 2e).

### 3.3 Implementation

Although the algorithm seems straight forward, computation of EMST and GG needs some considerations. The simplest algorithm to find an EMST, given n points, by constructing the complete graph on n vertices requires $O(n^2)$ time. The same approach constructs GG in $O(n^3)$ in 3 dimensions. Having noticed that EMST and GG are the subgraphs of every Delaunay triangulation of a point set even in 3 dimensions, applying Delaunay triangulation structure reduces the complexity to $O(nlogn)$ for each. Thus, we first compute the 3D Delaunay triangulation of the point set and use that structure for computing the EMST and then GG for each cluster resulting from the first phase. CGAL[1] is used as a geometric core library and for the Delaunay triangulation computations. Boost Graph library[2] is also employed for the EMST computations.

## 4 RESULTS AND DISCUSSION

To assess the explained algorithm, it was examined on a simulated data and some terrestrial laser scanner point clouds, with dense clusters and most of typical outliers. Below the result of all tests are reported.

### 4.1 Simulated data

Figures 2a-f show the algorithm sequence on a simulated data containing 656 points. Reference outlier and inlier are separated manually and the result of the algorithm is compared with the reference data. Table 1 shows some statistics, the number of outlier/inlier points and the first and second error types, of the result in the two phases. At the first phase points and clusters that are too far from the main clusters are detected while the second phase deals with local outliers. High $\beta$-error value at the first phase explanes that there are still some outliers among the intermediate cleaned data that are not detected. Having run the second phase remained outliers are detected and removed (the lower value of $\beta$-error). Of course the second phase increases the $\alpha$-error too (some correct points are detected as outlier) however this is a trade off one has to consider between decreasing $\beta$-error and increasing $\alpha$-error.

The HOD algorithm (phase-1)

| Reference | | Inlier | Outlier | |
|---|---|---|---|---|
| | Inlier | 569 | 3 | 572 |
| | Outlier | 9 | 75 | 84 |
| | | 578 | 78 | 656 |

| |
|---|
| $\alpha$-error |
| correct outlier |
| correct inlier |
| $\beta$-error |

| | | | |
|---|---|---|---|
| $\alpha$-error | 0.45% | $\beta$-error | 1.37% |

The HOD algorithm (phase-2)

| Reference | | Inlier | Outlier | |
|---|---|---|---|---|
| | Inlier | 565 | 6 | 571 |
| | Outlier | 3 | 4 | 7 |
| | | 568 | 10 | 578 |

| | | | |
|---|---|---|---|
| $\alpha$-error | 1.04% | $\beta$-error | 0.52% |

Table 1: Result of the proposed algorithm on the simulated data, phase one (upper table) and phase two (lower table).

### 4.2 Terrestrial case

Point clouds from Sternwarte[3] building, which was measured by Faro[4] LS880 laser scanner, used as the terrestrial test data set. Figure 4 left column, illustrates the original laser scanner data in different scan positions with different object facets. The right column of the figure shows each data set after has been cleaned by the algorithm. 99% and 95% confidential intervals are used for the global and local clustering phases respectively. The minimum inlier cluster density is considered as 100 points, according to the object size, distance of the scanner to the object and sampling resolution. Comparing the data set before and after outlier detection clearly shows the importance of the process and how the proposed algorithm performed. Close look at the results shows that not only the algorithm detected single outliers, but also clustered outliers with different densities have been detected.

The figure shows direct result of the algorithm on the data set. However in some cases it might happen that some cluster of outliers denser than the minimum inlier size exist in the data set

[1] http://www.cgal.org

[2] http://www.boost.org

[3] ETH Sternwarte in Zurich, an astronomical observatory planned and constructed by the two ETH-Professors Gottfried Semper (Architecture) and Rudolf Wolf (Astronomy and Mathematics) from 1859 to 1864

[4] http://www.faro.com

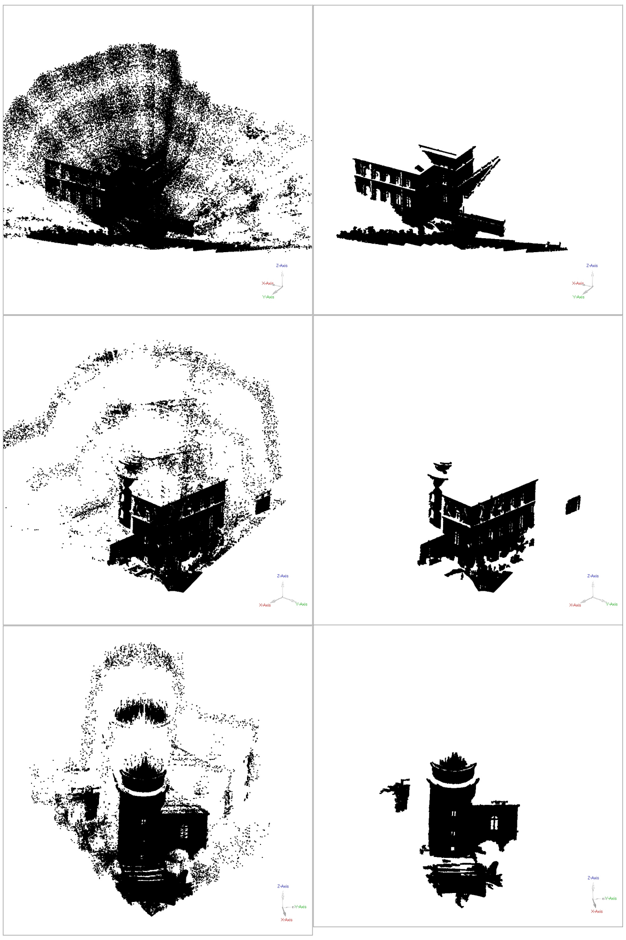

Figure 4: Results of applying the proposed outlier detection algorithm on some scans of the Sternwarte building which are captured by Faro laser scanner. Left column shows 3 different raw point clouds and the right column shows the cleaned point clouds after applying the algorithm.

which the algorithm consider them as inlier. This happens specially in case there are some real objects on the scene further than the main object that has to be measured. In that case detecting those objects as outliers is beyond the potential of the algorithm and needs some further information other than the point cloud itself. User interaction to determine if the cluster is an outlier or an object is required. The result of the algorithm seems quite handy again; User just needs to select a cluster to remove the whole outlier cluster and comparing to the case that the user has to remove the points of the outlier cluster separately, the user saves time for editing.

## 5 CONCLUSIONS AND FUTURE WORK

Detecting outliers in laser scanner point cloud using a hierarchical algorithm is proposed and investigated in this paper. The algorithm approaches in two stages. In the first stage it removes relatively large scale erroneous measurements and in the second phase it detects and removes the outliers that might not be as large as the first ones but according to the scanned object surfaces they are considered as wrong measurements. The algorithm has unconstrained behavior to the preliminary knowledge of the scanned scene and it dose not suffer from the varying density of the points. The algorithm efficiency is assessed by a test on a simulated point cloud, which contained single and clustered outliers. The assessment is done with respect to a manual separation of outlier/inlier points. The $\alpha$-error and $\beta$-error (type I and II errors) are estimated and the results show that most of the detected outliers are really outliers according to the definition of the outliers (Hawkins, 1980). In addition some examples in terrestrial laser scanner point clouds are presented and the behavior of the algorithm on the data sets are shown and discussed. Results show that the algorithm detects single and even clustered outliers almost without user interaction. Also, in case that the user editing is required, the result of the algorithm provides easier editing procedure due to the selection of point clusters rather than individual points. Test of the algorithm on airborn laser scanner data set is another challenge that the author is currently working on.

## ACKNOWLEDGMENTS

This research was partially supported by the Ministry of Science, Research and Technology (MSRT) of IRAN. The author appreciates the cooperation of Faro company, who provided the author a laser scanner for measurements.

## REFERENCES

Adamy, U., Giesen, J. and John, M., 2000. New techniques for topologically correct surface reconstruction. IEEE Visualization 2000 pp. 373–380.

Attene, M. and Spagnuolo, M., 2000. Automatic surface reconstruction from point sets in space. Computer Graphics Forum (Procs. of EUROGRAPHICS '00) 19(3), pp. 457–465.

Aupetit, M., 2003. High-dimensional labeled data analysis with gabriel graphs. ESANN'2003 proceedings - European Symposium on Artificial Neural Networks (ISBN 2-930307-03-X), pp. 21–26.

Barnett, V. and Lewis, T., 1994. Outliers in Statistical Data. John Wiley and Sons, Inc., Hoboken, New Jersey.

Breunig, M., Kriegel, H., Ng, R. and Sander, J., 2000. Lof: Identifying density-based local outliers. In Proc. ACM SIGMOD Conf p. 93104.

Gabriel, K. R. and Sokal, R. R., 1969. A new statistical approach to geographic variation analysis. Systematic Zoology 18, pp. 259–278.

Hawkins, D., 1980. Identification of Outliers. Chapman and Hall, London.

Jain, A., Murty, M. and Flynn, P., 1999. Data clustering: A review. acm computing surveys. ACM Computing Surveys 31(3), pp. 264323.

Johnson, T., Kwok, I. and Ng, R., 1998. Fast computation of 2-dimensional depth contours. Proc. KDD 1998 p. 224228.

Knorr, E., Ng, R. and Tucakov, V., 2000. Distance-based outliers: Algorithms and applications. VLDB Journal 8, pp. 237253.

Marchette, D. J., 2004. Random Graphs for Statistical Pattern Recognition. John Wiley and Sons, Inc., Hoboken, New Jersey, ISBN 0471221767.

Papadimitriou, S., Hiroyuki, K., Gibbons, P. B. and Faloutsos, C., 2002. Loci: Fast outlier detection using the local correlation integral. Intel Corporation, IRP-TR-02-09.

Rousseeuw, P. and Leroy, A., 1987. Robust Regression and Outlier Detection. John Wiley and Sons, Inc., Hoboken, New Jersey.

Sithole, G., 2005. Segmentation and Classification of Airborne Laser Scanner Data. Ph.D. Thesis. Publications on Geodesy, 59. Publication of Netherlands Geodetic Commision, ISBN 90 6132 292 8.

Sotoodeh, S., 2006. Outlier detection in laser scanner point clouds. International Society of Photogrammetry and Remote Sensing.

Vanicek, P. and Krakiwsky, E., 1982. Geodesy: The Concepts. University of New Brunswick, Canada, ISBN 0-444-86149-1.

Veltkamp, R., 1994. Closed Object Boundaries from Scattered Points. Lecture Notes in Computer Science 885, Springer, ISBN 3-540-58808-6.

Zahn, C. T., 1971. Graph theoritical methods for detecting and describing gestalt clusters. IEEE transactions on Computers 20(1), pp. 68–86.