CLUSTERING BASED PLANAR ROOF EXTRACTION FROM LIDAR DATA

Aparajithan Sampath **Jie Shan**
Geomatics Engineering, School of Civil Engineering
Purdue University, 550 Stadium Mall Drive
West Lafayette, IN 47907, USA, Phone: +1-765-494-2168, Fax: +1-765-496-1105
{asampath, jshan}@ecn.purdue.edu

ABSTRACT

An approach to generate 3-D models of buildings from lidar data collected from an urban setting is presented. The present research focuses on extracting roof structures from a point cloud of a building using a combination of data-mining techniques. To extract the roof structure, an assumption of planarity has been made, i.e. it is assumed that the roof can be modeled by a set of planar segments. The task then is to map each of the building point to a planar segment. We present a method that first separates points lying on or near breaklines, by which we mean points that are near the intersection of two planar surfaces or points that are near step-edges. Treating these points to be ambiguous, (i.e. they belong to more than one plane), we separate them from points that are exclusively planar. We then use neighborhood functions to determine what we call planar patches, and their direction cosines using their eigenvalue and eigenvector characteristics. Then an unsupervised data clustering technique to cluster these planar patches into one single plane is described. Most clustering techniques require that the number of clusters be known. In our case these clusters represent roof planes. We present a way to determine the number of roof planes (clusters) by using an iterative combination of k-means and density based clustering methods.

INTRODUCTION

Airborne laser scanning technique is one of the most efficient ways of collecting three-dimensional (3-D) spatial information for large areas. As a result of this scanning process the dataset obtained consists of thousands of points known as a point cloud. Even though we get direct 3-D data, extracting features of interest is not a trivial task. In this paper, we will present and demonstrate a technique for extracting roof planes of buildings.

In general, the techniques to reconstruct buildings can be put under two categories. One technique is to model buildings with a set of parameters, whose values are then determined through the reconstruction procedure (Maas and Vosselman, 1999). In this approach the selection of an appropriate model is usually carried out by human operator in an interactive manner. The second technique is data driven, i.e., only generic assumption such as planar surface is made on the buildings to be reconstructed. Finding the component planes is essentially the key issue in reconstructing the buildings. The topic in this study falls into the second technique. Rottensteiner and Briese (2003) generate a digital surface model and determine a few “homogenous pixels”, i.e., pixels which are most likely to be planar. Connected homogenous pixels are used as initial seeds to generate planar regions. Planar facets are then used to determine lines of intersection of planes and step edges. This method is improved in (Rottensteiner et al, 2005) by introducing statistical tests to minimize the dependence on thresholds for all stages, including detection and classification of roof planes, and breaklines and step edges. Fransens et al (2003) present an algorithm to reconstruct piecewise planar approximations, to which a hierarchical principal component analysis is used to assess planarity in a top-down manner. Planar patches are extracted, merged and grown, followed by a triangulation procedure conforming to the underlying point cloud features. Peternell and Steiner (2004) describe a method where the data is separated into grids, and for each grid and its eight neighbors surface normals are determined. Then grids with similar surface normals are connected using a connected component analysis. Al-Harthy and Bethel (2004) use a technique of moving windows, which is similar to the above technique. They determine the slope in x, slope in y and the Z intercept for points in each grid placed in the data space. Then a region growing approach is used to extract planar segments. Vosselmann et al (2004) review the techniques to extract different surfaces, including planar, cylindrical and spherical surfaces from lidar point clouds. Brenner (2005) summarizes various building reconstruction techniques using image, lidar and map data that have been suggested by different authors.

The methodology presented in this paper is similar to Fransens et al (2003). We address the problem of planar surface segmentation based on pattern classification theory. As the first step, the points on breaklines, i.e., the intersection of two planar surfaces, are detected based on eigenvalue and eigenvector evaluation. Such breakline points are excluded from the subsequent clustering calculation to avoid ambiguity. For the remaining data set, we create a number of planar patches using neighboring points. Eigenvalue and eigenvector functions are used to estimate the surface normals or direction cosines for each planar patch. An iterative procedure of combined density based clustering and K-means clustering is then proposed to generate planar surfaces from the planar patches. Such a procedure makes no assumption about the nature of the planar patches and the number of clusters. As the results of this clustering process, lidar points lying on each planar surface are determined. This paper presents the theory and discusses the implementation details along with successful examples.

BREAKLINE DETECTION

It is observed that all the lidar points that lie on a single planar surface can be described using two independent variables other than three Cartesian coordinates. Such independent variables are related to the eigen-space and certain transformations must be employed. Consider a group of points P in R^3. Let $\bar{P}$ be the mean vector, the covariance matrix $\sum$ is then calculated. For this covariance matrix $\sum$, we determine its eigenvalues $\lambda_1, \lambda_2, \lambda_3$ where $\lambda_1 \le \lambda_2 \le \lambda_3$ and their corresponding eigenvectors $\Lambda_1, \Lambda_2, \Lambda_3$. It can be noted that the direction the eigenvector Λ_1 corresponding to the minimum eigenvalue λ_1 will be in the direction of minimum variance. If the point set P consists of points that lie on a plane, the eigenvalue λ_1 should be very close to zero and the eigenvector Λ_1 should be in the direction perpendicular to the plane, i.e., the normal of the plane. On the other hand, if λ_1 is not negligible, it can be concluded that the point set P does not lie on one plane. If only buildings with planar roof are considered, such point set actually lies on the intersection of two planes, i.e., breaklines. The above principle is summarized as: if $[\lambda_1 / (\lambda_1 + \lambda_2 + \lambda_3)] < \varepsilon$, then the point set P is planar; otherwise it is on the breaklines.

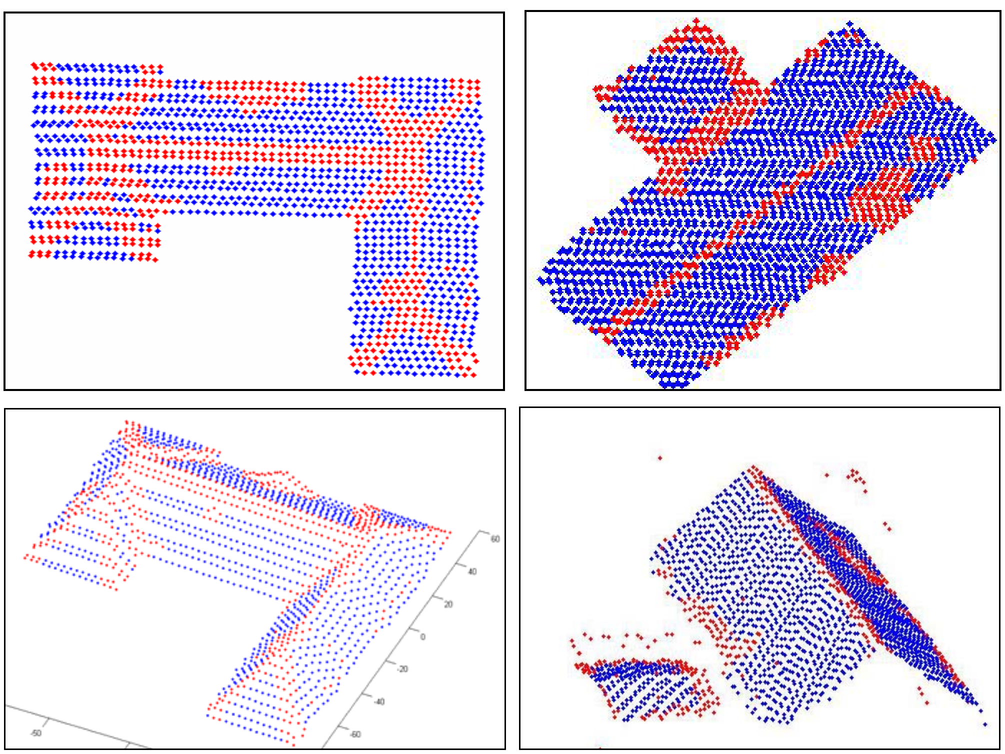

Figure 1. Breakline points in red detected from eigenvalue analysis (left, building 1; right, building 2)

FORMING PLANAR PATCHES

We call a point as lying on a plane or a breakline by considering it along with its neighboring points. Using the above principle, the point is accorded a planar or a non-planar (breakline) status. This test is carried for all points individually, by considering a given point along with its neighborhood. As such, we assume that roofs are a series of planar segments. Because of the discrete nature of the dataset, points that lie close to breaklines are also considered to be lying on breaklines. It should be noted that this approach is also able to detect all other non-planar points, such as 'blunders' returned from trees etc. Figure 1 shows the detected breakline points in red for two buildings.

Once the breakline points are detected, we are left with a dataset that consists of points on one of the several roof planes. The remaining dataset is then subdivided into a number of "planar patches". Each planar patch consists of a small number (typically between 5 and 10) of neighboring points. Since lidar scans are linear, the neighborhood should have sufficient extension in two dimensions to avoid only a set of collinear points being selected. As explained in the above section, the covariance matrix for this set of points is determined and the eigenvector and the eigenvalues of this covariance matrix are also determined. The eigenvector corresponding to the smallest eigenvalue gives the direction of the normal to our planar patch. Once we have the surface normals (direction cosines), we can use clustering algorithms discussed above to bunch up these planar patches into planes.

CLUSTERING PLANAR PATCHES

We now have a series of planar patches that divide the roof. Many of these patches belong to the same plane. Thus the next step is to combine these patches into one plane based on the patch's normal vectors and its distance to the origin as our classification axes. The goal of this section is to group together those patches that can coalesce to a single plane. To accomplish this task, we make use unsupervised clustering techniques.

Data clustering is a technique in data mining, machine learning and pattern classification. It essentially divides the given data into different clusters such that data in each cluster shares a common trait. Clustering algorithms can be broadly classified into hierarchical and partitioning algorithms (Jain et al 1999; Berkhin, 2002). A hierarchical clustering method produces a nested series of partitions. It produces a dendrogram, which is a tree type structure. At the top level of the dendrogram, all data is grouped as the same; and at the bottom, all data is grouped as separate.

On the other hand, the partitioning algorithms produce partitions of data based on a similarity (or dissimilarity) measure. It selects an initial set of partitions and iteratively computes the final partition (clusters). As one of the common and simple algorithms, K-means is used for clustering in this study. It can be described as follows: For the K-means algorithm to select K number of clusters in a dataset

1. Initially K cluster centers are chosen randomly among the data points.
2. Each data point is assigned to its nearest cluster center.
3. The cluster centers are recomputed using the current clusters.
4. Go to step 2 if the convergence criterion is not met, which is usually a decrease in squared error (the distance of each data point to its cluster center).

The major disadvantage of K-means approach is that the number of clusters should also be known. In this study, the aim is to cluster together similar planar patches based on its normal vector and distance to the origin. In our case, the number of clusters corresponds to the number of roof planes in a building and this information is not immediately known. Another disadvantage of K-means method is that the initial choice of cluster centers is important; otherwise the algorithm will terminate at a local minimum of the squared errors. Therefore, such shortcomings need to be properly considered in our application.

The methodology proposed by (Yager and Filev, 1994) is adopted here to estimate the number and initial location of cluster centers (roof planes). This method is an example of density based clustering. The data space is gridded and the density of each grid is computed based on the distance of grid center to the data points. A grid with many data points nearby will have a high potential value. The grid with the highest potential value is chosen as the first cluster center. Also, once the first cluster center (grid) is chosen, the potential of the nearby grids is reduced based on their distance from the cluster center. This is to make sure that two grids that are close together do not become different clusters. The next cluster center is then chosen from the remaining grids with the highest potential. Chiu (1994) further developed this idea by doing away with grids and using actual data point as cluster centers. Each data point is given a potential based on its neighboring points, and the point with the highest potential is considered as a cluster center. Chiu's study calculates the potential as $P_i = \sum_{j=1}^{n} e^{-\frac{4}{r^2}\left(X_i - X_j\right)^2}$ for each point X_i, where r is the radius (distance from X_i) beyond which points are not considered for potential calculation.

Our experience with this method suggests that the value of the radius is crucial. It is found that having smaller radii increases the number of clusters and larger radii decreases the number of clusters. An important thing to note here is that the feature vector for the clustering algorithm consists of the three direction cosines $\cos(\alpha), \cos(\beta), \cos(\gamma)$ of the planar patches. It is difficult to pick proper radius values that can effectively define good neighborhoods in this domain. To overcome the problem, the density based clustering method is iteratively implemented, starting from small radii, which gives a larger number of clusters, to larger radii, which gives less number of clusters. The cluster centers and the number of clusters generated in each iteration are used as input to K-means clustering algorithm, which produces clusters with new cluster centers.

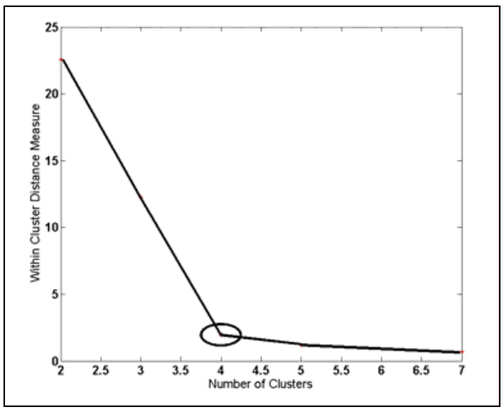

Figure 2. Determining the number of clusters

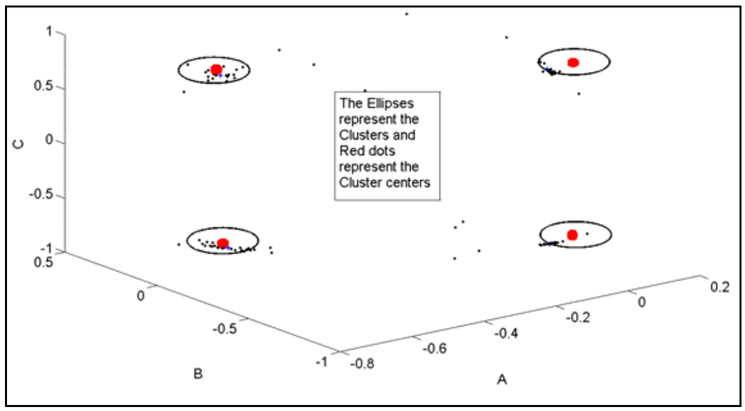

Figure 3. Normal vector clusters with the red dots as cluster centers

As the radius for density clustering is decreased from a very large to a small value (which indirectly increases the number of clusters), the sum of distances from each point to center keeps getting smaller, till it reaches a flat region. Figure 2 illustrates this fact. As the number of clusters increase, each cluster becomes more compact, and the distance of the cluster point to the cluster center decreases. The cluster distances decrease sharply and then stabilize. The shape of the curve looks like an elbow joint. The point where the cluster distance stops decreasing sharply (the joint in the elbow) is taken as the actual number of clusters. A further understanding of such a phenomenon can be obtained in Tibshirani et al (2001).

This process simultaneously gives us the number of clusters and the cluster centers. The cluster centers essentially mean that the normal vectors of the different roof planes are known. A distribution of the normal vectors and their cluster centers is shown in Figure 3. We now have clusters of planar patches. Each cluster consists of a number of patches that are all parallel to each other, and many of them will be coplanar too. To separate the parallel patches within a cluster, we examine the distance of each patch from the origin, using the distance equation $D = N_x X_c + N_y Y_c + N_z Z_c / \sqrt{N_x^2 + N_y^2 + N_z^2}$ where (X_c, Y_c, Y_c) is the center of the patch and (N_x, N_y, N_z) are its normal vectors. A connectivity check is used to separate coplanar patches that are not spatially adjacent. Finally, the breakline points detected at the beginning are tested against each plane discriminant $N_{xi}X + N_{yi}Y + N_{zi}Z + D_i$, where i=1,2,….n for all n planes, to determine the plane or planes it may belong to.

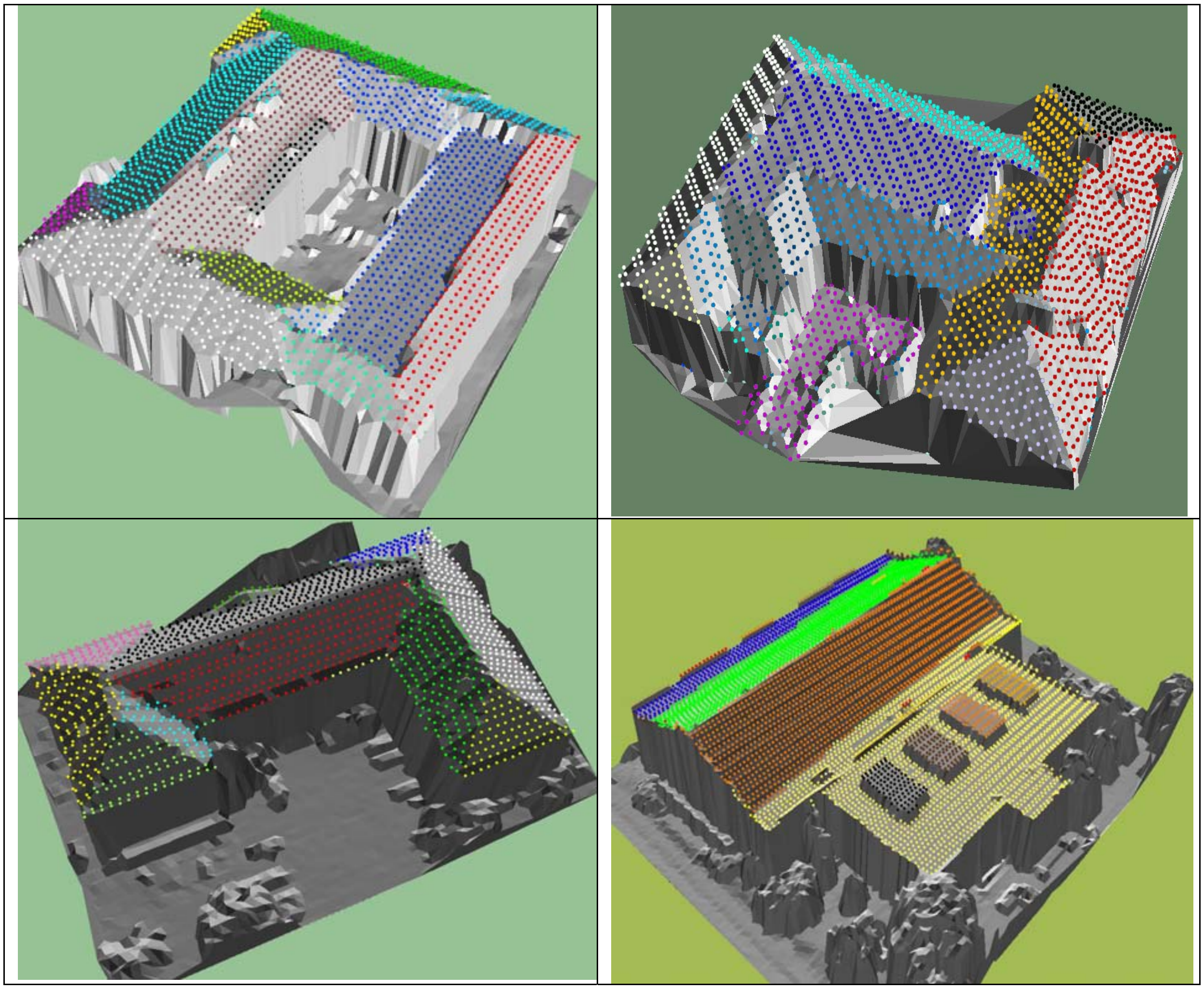

Figure 4. Color coded planar roof points

DISCUSSIONS

Color coded roof points are shown by clustered roof planes in Figure 4. The results indicate that almost of all points are classified correctly to their respective planes. The use of the equation of the various planes as discriminant functions can cause some misclassification errors. For instance, a particular plane segment might intersect another plane segment when their equations are solved, and such points that satisfy both these equations can be misclassified. Use of neighborhood analysis such as comparing the classification of a point with its neighborhood points reduces such errors.

We have tried to minimize the use of thresholds as much as possible. The threshold for the ratio $\lambda_1 / (\lambda_1 + \lambda_2 + \lambda_3)$ is set to be 0.005 based on experiments with many buildings. Decreasing or increasing this threshold will cause more or less points to be classified as breakline points. The number of points that are included as breaklines (more specifically, the thickness) does not cause a great deal of concern, if the planes of the roofs are reasonably big. This is because only a few points are needed to define a plane, and we can assume that a reasonably big plane shall have a sufficient number of lidar returns. In this case, it is possible to easily extract plane parameters, and the breakline points can be assigned to their proper planes based on the equations of the derived planes, as discussed in the previous section.

The presence of smaller planes in the roof is a cause for concern. In general, the size of the planar patches should be selected no larger than the smallest roof entity. The risk with smaller patches is that there might not be enough points on each patch to define a plane or the points might be nearly collinear.

The size of the patches also plays an important role in the clustering process. In general, it is preferable to have a large number of small patches rather than a small number of large patches. This helps the clustering algorithm to be robust. A large patch and a small patch carry equal weight in the clustering process, and having larger number of patches means we have more data points in the feature (normal vector) domain for clustering.

Since the clustering process needs the number of clusters (planes) and an initial approximation for the planes as input, we go through an iterative process by combining a density based approach with the K-means clustering approach to autonomously determine these parameters. The radius for the density approach is initially kept high. This means that the initial clusters will probably be composed of more than one smaller cluster. This causes the distance of the cluster components to the centroid to be high. As we decrease the radius, the number of clusters increase, and the clusters become more compact. This causes the centroidal distance to decrease sharply till we reach the optimal number of clusters. This iterative procedure eliminates the use of thresholds and generates the number of roof planes and the roof plane parameters without any assumptions regarding the building, its orientation or the direction of the breaklines.

In summary, we have demonstrated a method to segment the lidar points from roof of buildings into different roof planes. The only assumption that has been made is that the roofs are planar. We demonstrate a method of determining points that lie along the breaklines, i.e. along intersection of two (or more) roof planes, by using eigenvalue and eigenvector analysis. We also show that points that are reflected off trees are also categorized as breaklines. Therefore, this is also a method of removing points that are from trees, rather than the building top. The non-breakline points are then divided to planar patches characterized by its normal vector and distance to origin. To group the patches into planar surfaces, an iterative combination of density based clustering, which generates cluster centers and the number of clusters, with K-means clustering are investigated, which yields satisfactory and promising results of moderate complex buildings.

REFERENCES

Alharty, A., and Bethel, J., “Detailed building reconstruction from airborne laser data using a moving surface method”, IAPRS XXXV - B3, pp. 213-218, 2004.

Berkhin, P., “Survey of Clustering Data Mining Techniques.” In: Accrue Software, 2002.

Brenner C, 2005. Building reconstruction from images and laser scanning, International Journal of Applied Earth Observation and Geoinformation, Vol 6, pages 187-198.

Chiu, S., "Fuzzy Model Identification Based on Cluster Estimation," Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, Sept. 1994.

Fransens, Jan, Philippe Bekaert and Frank Van Reeth, 2003. A Hierarchical PCA-based Piecewise Planar Surface Reconstruction Algorithm, submitted to SIAM, accessible at www.racine.eu.com/imgLibrary/siam2003.pdf

Jain A.K, Murthy, M.N. Flynn, P.J. "Data Clustering: A Review ". ACM Computing Surveys, Vol. 31, No. 3, September 1999

Maas, H., Vosselman, G., “Two algorithms for extracting building models from raw laser altimetry data”, ISPRS Journal of Photogrammetry & Remote Sensing vol. 54, pp 153–163, 1999.

Peternell M., Steiner T., “Reconstruction of piecewise planar objects from point clouds”, Computer-Aided Design, Vol. 334, No.36, pp. 333–342, 2004

Rottensteiner, F. and Briese, C. “Automatic Generation Of Building Models From Lidar Data And The Integration Of Aerial Images”, ISPRS, Vol. XXXIV, Dresden, 2003

Rottensteiner, F., Trinder, J Clode, S, Kubik, K., “Automated Delineation of Roof Planes from Lidar Data”, ISPRS WG III/3, V/3 Workshop “Laser Scanning 2005”, Enschede, The Netherlands, Sept 12-14 2005.

Tibshirani, R., Walther, G., & Hastie, T., “Estimating the number of clusters in a data set via the gap statistic.” Journal of the Royal Statistical Society: Series B (Statistical Methodology) Volume 63, Page 411 2001.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T. (2004) Recognising structure in laser scanner point clouds IAPRS, vol. 46, part 8/W2, Freiburg, Germany, October 4-6, pp. 33-38.

R. Yager and D. Filev. “Approximate clustering via the mountain method,” IEEE Transactions on Systems, Man & Cybernetics, VOL. 24, NO. 8, August 1994