

PAIRWISE LINKAGE FOR POINT CLOUD SEGMENTATION

Xiaohu Lu, Jian Yao,* Jingjie Tu, Kai Li, Li Li, Yahui Liu

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China
(xiaohu.lu,jian.yao,kaili.li,liuyahui)@whu.edu.cn
<http://cvrs.whu.edu.cn/>

Commission III, WG III/2

KEY WORDS: Pairwise Linkage, Clustering, Point Cloud Segmentation, Slice Merging

ABSTRACT:

In this paper, we first present a novel hierarchical clustering algorithm named Pairwise Linkage (P-Linkage), which can be used for clustering any dimensional data, and then effectively apply it on 3D unstructured point cloud segmentation. The P-Linkage clustering algorithm first calculates a feature value for each data point, for example, the density for 2D data points and the flatness for 3D point clouds. Then for each data point a pairwise linkage is created between itself and its closest neighboring point with a greater feature value than its own. The initial clusters can further be discovered by searching along the linkages in a simple way. After that, a cluster merging procedure is applied to obtain the finally refined clustering result, which can be designed for specialized applications. Based on the P-Linkage clustering, we develop an efficient segmentation algorithm for 3D unstructured point clouds, in which the flatness of the estimated surface of a 3D point is used as its feature value. For each initial cluster a slice is created, then a novel and robust slice merging method is proposed to get the final segmentation result. The proposed P-Linkage clustering and 3D point cloud segmentation algorithms require only one input parameter in advance. Experimental results on different dimensional synthetic data from 2D to 4D sufficiently demonstrate the efficiency and robustness of the proposed P-Linkage clustering algorithm and a large amount of experimental results on the Vehicle-Mounted, Aerial and Stationary Laser Scanner point clouds illustrate the robustness and efficiency of our proposed 3D point cloud segmentation algorithm.

1. INTRODUCTION

Segmentation is one of the most important pre-processing step for automatic processing of point clouds. It is a process of classifying and labeling data points into a number of separate groups or regions, each corresponding to the specific shape of a surface of an object. The cluster analysis which classifies elements into categories according to their similarities has been applied in many kinds of fields, such as data mining, astronomy, pattern recognition, and can also be applied on the segmentation of 3D point clouds.

1.1 Point Cloud Segmentation

Segmentation in 3D point clouds obtained from laser scanners is not trivial, because the three dimensional point data are usually incomplete, sparsely distributed, and unorganized, also there is no prior knowledge about the statistical distribution of the points, and the densities of points vary with the point distribution. Many methods have been developed to improve the quality of segmentation in 3D point clouds that can be classified into three main categories: edge/border based, region growing based and hybrid.

The edge/border based methods attempt to detect discontinuities in the surfaces that form the closed boundaries, and then points are grouped within the identified boundaries and connected edges. These methods usually apply on the depth map where the edges are defined as the points where the changes of the local surface properties exceed a given threshold. The local surface properties mostly used are surface normals, gradients, principal curvatures, or higher order derivatives (Sappa and Devy, 2001, Wani and Arabnia, 2003). However, due to noise caused by laser scanners themselves or spatially uneven point distributions in 3D space,

such methods often detect disconnected edges which makes it difficult for them to identify closed segments (Castillo et al., 2013) without a filling or interpretation procedure.

The region growing based approaches deal with segmentation by detecting continuous surfaces that have homogeneity geometrical properties. In the segmentation of unstructured 3D point clouds, these methods firstly choose a seed point from which to grow a region, and then local neighbors of the seed point are combined with it if they have similarities in terms of surface point properties such as orientation and curvature (Rabbani et al., 2006, Jagannathan and Miller, 2007). There are also algorithms which take a sub window (Xiao et al., 2013) or a line segment (Harati et al., 2007) as the growth unit. (Woo et al., 2002) proposed an octree-based 3D-grid method to handle large amount of unstructured point clouds. The smoothly connected regions are the key points of the region growing based methods. Surface normal and curvatures constraints were widely used to find the smoothly connected areas (Klasing et al., 2009, Belton and Lichti, 2006). In general, the region growing based methods are more robust to noise than the edge-based ones because of the using of global information (Liu and Xiong, 2008). However, these methods are sensitive to the location of initial seed regions and inaccurate estimations of the normals and curvatures of points near region boundaries can cause inaccurate segmentation results, and also outliers can result in over- and under-segmentation.

The hybrid approaches use both edge/border-based and region-growing-based methods to overcome limitations in the respective approaches (Vieira and Shimada, 2005, Lavoué et al., 2005). (Benkő and Várady, 2004) proposed a hybrid approach for the segmentation of engineering objects, which detects sharp edges and small blends using an edge-based approach in the first step and then finds smooth regions after filtering out sharp edges and small blends. However, the success of these hybrid methods de-

*Corresponding author

depends on the success of either or both of the underlying methods.

1.2 Clustering

The cluster analysis, which aims at classifying elements into categories on the basis of their similarities, has been applied in many kinds of fields, such as data mining, astronomy, and pattern recognition. In the last several decades, thousands of algorithms have been proposed to try to find a better solution for this problem in a simple but philosophical way. In general, these algorithms can be divided into two categories: partitioning and hierarchical methods. The partitioning clustering algorithms usually classify each data point to different clusters via a certain similarity measurement. The traditional algorithms K-Means (MacQueen et al., 1967) and CLARANS (Ng and Han, 1994) belong to this category. The hierarchical methods usually create a hierarchical decomposition of a dataset by iteratively splitting the dataset into smaller subsets until each subset consists of only one object, for example, the single-linkage (SLink) method and its variants (Sibson, 1973).

1.3 Clustering in Point Cloud Segmentation

The clustering algorithms which classify elements into categories on the basis of their similarities can also be applied on the segmentation of 3D point clouds. The widely used K-Means algorithm (MacQueen et al., 1967), which can divide the data points into K (a predefined parameter that gives the number of clusters), was applied in (Lavoué et al., 2005) to classify the point clouds into 5 clusters according to their curvatures. The shortcoming of the K-Means clustering algorithm is that it needs to know the number of clusters beforehand, which can't be predefined in many cases. To overcome this shortcoming, the mean shift algorithm (Comaniciu and Meer, 2002), which is a general nonparametric technique to cluster scattered data, was employed on the point cloud segmentation (Yamauchi et al., 2005b, Yamauchi et al., 2005a, Zhang et al., 2008). In the works of (Yamauchi et al., 2005b, Yamauchi et al., 2005a), the mean shift algorithm was employed to integrate the mesh normals and the Gaussian curvatures, respectively. In the work of (Liu and Xiong, 2008), the normal orientation was converted into the Gaussian Sphere, and a novel cell mean shift algorithm was proposed to identify planar, parabolic, hyperbolic or elliptic surfaces in a parameter-free way. However, most of the point cloud segmentation methods based on clustering can only discover small amount segmentations, which can be employed on some industry applications but may fail on the vehicle-mounted and aerial laser scanner point clouds which contains thousands of surfaces in large indoor/outdoor scenes.

1.4 Objectives and Motivation

In this paper, we aim to develop a simple, efficient point cloud segmentation algorithm which can be applied on a large amount of unstructured Vehicle-Mounted, Aerial and Stationary Laser Scanner point clouds by employing the clustering algorithm on point cloud segmentation. To achieve this goal, we introduce two algorithms:

P-Linkage Clustering: Based on the assumption that: a data point should be in the same cluster with its **closest neighboring point** (CNP) which is more likely to be a cluster center, we propose a novel hierarchical clustering method named Pairwise Linkage (P-Linkage) which can discover the clusters in a simple and efficient way. Firstly, a pairwise linkage procedure is applied to link each data point to its CNP on the data-point level. Then the initial clusters can be discovered by searching along the pairwise linkages starting from the points **with local-maximal densities**.

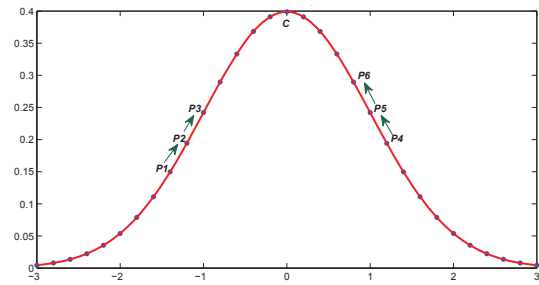


Figure 1: An illustration of the pairwise linkage on a 2D Gaussian curve. Derived from the non-maximum suppression, the P-Linkage compares each data point to its neighbors and forms the linkages from $p_1 \rightarrow p_2 \rightarrow p_3 \dots \rightarrow c$.

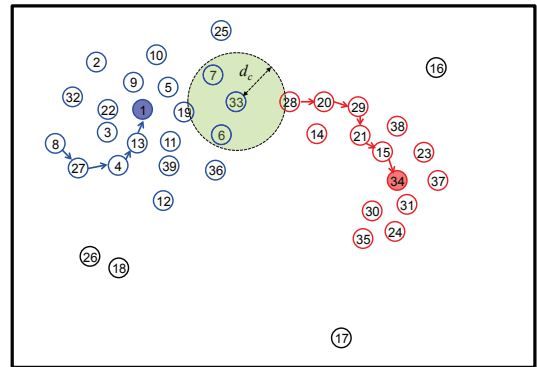


Figure 2: An illustration of the hierarchical clustering procedure. The data points p_1 and p_{34} are the cluster centers, the symbol \rightarrow indicates the pairwise linkage, and the big circle in green denotes the neighborhood set of a data point with a cutoff distance d_c .

The proposed clustering method is not iterative and needs only one step for general cases, and also a cluster merging method is proposed for specific cases.

Point Cloud Segmentation: Based on the proposed P-Linkage clustering, we develop a simple and efficient point cloud segmentation algorithm which needs only **one parameter** and can be applied on a large amount of unstructured Vehicle-Mounted, Aerial and Stationary Laser Scanner point clouds. The P-Linkage clustering in point cloud segmentation takes **the flatness of the estimated surface of a 3D point as its feature value** and forms the initial clusters via point data collection along the linkages. For each **initial cluster we create a slice**. All the slices are merged in a simple or efficiently strategy to get the final segmentation result. The proposed point cloud segmentation algorithm needs only one parameter to balance the segmentation results of planar and surface structures.

The remainder of this paper is organized as follows. The proposed P-Linkage clustering algorithm is detailedly described in Section 2. The point cloud segmentation algorithm by employing the P-Linkage clustering on 3D unstructured point clouds is introduced in Section 3. Experimental results on different kinds of synthetic and real data are presented in Section 4 followed by the conclusions drawn in Section 5.

2. PAIRWISE LINKAGE

The key conception of the P-Linkage clustering method is that: **a data point p_i should be in the same cluster with its closest neighbor p_j that is more likely to be a cluster center, and this relationship between p_i and p_j is called a pairwise linkage**. This conception is derived from the idea of non-maximum suppression

(NMS) (Canny, 1986, Neubeck and Van Gool, 2006), in which one data point is only needed to compare with its neighbors and will be suppressed if it is not local-maximal. Figure 1 shows an illustration of the NMS, from which we can see that \mathbf{p}_1 is suppressed by \mathbf{p}_2 and the same suppression occurs on \mathbf{p}_2 when it is compared to \mathbf{p}_3 , which result in a link $\mathbf{p}_1 \rightarrow \mathbf{p}_2 \rightarrow \mathbf{p}_3$. In this way, all the data points on the curve are finally linked to the cluster center \mathbf{c} , which is the local-maximal one, just via comparing to their neighboring points. **In fact, the P-Linkage clustering makes up the gap between the local to the global information of the data points, which makes it more robust than the local-based clustering methods and more efficient than the global-based ones.** In the following subsections, we will introduce the pairwise linkage algorithm on the clustering of 2D data points, which takes the density of a data point as its feature value to build linkages.

Cutoff Distance: The cutoff distance d_c , as shown in Figure 2, is a global parameter to demarcate the neighborhood set of a data point \mathbf{p}_i from other data points. In the recent work of (Rodriguez and Laio, 2014), the value of d_c was set as the value at the 1% – 2%-th of all the distances between any two data points, denoted as the set \mathcal{D} , which were sorted in ascent order. However it is not appropriate to set the cutoff distance d_c in this way because d_c is an indicator of the distribution of the neighboring points, and it should be derived from the local neighborhood instead of \mathcal{D} . Thus, we propose a simple method to determine the value of d_c , which is described as follows. For each data point \mathbf{p}_i , the distance between \mathbf{p}_i and its closest neighbor is recorded in \mathcal{D}_{cn} , and then d_c is computed as:

$$d_c = scale \times median(\mathcal{D}_{cn}), \quad (1)$$

where *scale* is a customized parameter which means the cutoff distance d_c is *scale* times the value of the median value of the set \mathcal{D}_{cn} . In this way, d_c represents much more neighborhood distribution information than setting it 1% – 2%-th of \mathcal{D} . Only the data points whose distances to \mathbf{p}_i are smaller than d_c are considered as the neighborhood set of \mathbf{p}_i , which is denoted as \mathcal{I}_i , as the green circle shown in Figure 2.

Density: (Rodriguez and Laio, 2014) defined the density of a data point \mathbf{p}_i as the number of data points of its neighbors, which is discrete-value and thus is not suitable for our application requiring continuous values for densities. In our proposed method, the **density ρ_i of a data point \mathbf{p}_i is calculated by applying a Gaussian Kernel on all the data points as follows:**

$$\rho_i = \sum_{j \in [1, N], j \neq i} \exp(-(d_{ij}/d_c)^2), \quad (2)$$

where N denotes the number of all the data points and d_{ij} is the distance between two points \mathbf{p}_i and \mathbf{p}_j .

Pairwise Linkage: With the densities of all the data points, the pairwise linkage can be recovered in a non-iterative way, which is performed as follows. For a data point \mathbf{p}_i whose neighborhood set is \mathcal{I}_i , we traverse each point in \mathcal{I}_i and find the closest data point \mathbf{p}_j whose density is greater than that of \mathbf{p}_i , and then we consider the data point \mathbf{p}_i should be in the same cluster as \mathbf{p}_j and record the linkage between the data points \mathbf{p}_i and \mathbf{p}_j . If the density of \mathbf{p}_i is local-maximal, which means that there exists no data point in \mathcal{I}_i whose density is greater than that of \mathbf{p}_i , we consider \mathbf{p}_i as a cluster center. The result of the pairwise linkage procedure is comprised of a lookup table \mathbb{T} recording the linkage relationship and a set \mathcal{C}_{center} recording all the cluster centers.

Hierarchical Clustering: The hierarchical clustering is a top-down procedure, which is similar to that of the divisive clustering algorithm. For each cluster center \mathbf{c}_i in \mathcal{C}_{center} , we start

searching the lookup table \mathbb{T} from \mathbf{c}_i in a depth-first or breadth-first way to gather all the data points that are directly or indirectly connected with \mathbf{c}_i , which generates a cluster whose center is \mathbf{c}_i . The whole hierarchical clustering finds the final clusters \mathcal{C} . Figure 2 shows an illustration of the hierarchical clustering procedure. From Figure 2, we can observe that \mathbf{p}_1 is the cluster center due to its local maximal density and there are four pairwise linkages between $(\mathbf{p}_1, \mathbf{p}_{13})$, $(\mathbf{p}_{13}, \mathbf{p}_4)$, $(\mathbf{p}_4, \mathbf{p}_{27})$, and $(\mathbf{p}_{27}, \mathbf{p}_8)$. Thus the hierarchical clustering is performed as $\mathbf{p}_1 \rightarrow \mathbf{p}_{13} \rightarrow \mathbf{p}_4 \rightarrow \mathbf{p}_{27} \rightarrow \mathbf{p}_8$. **By this way, the clustering information is propagated from the dense data points to the sparse ones, which is similar to the heat propagation.**

Cluster Merging: When the data points are Gaussian-distributed, as shown in Figure 1, the hierarchical clustering via pairwise linkage can find the global cluster centers and recover the clusters quite well, but may fail in fragmented clustering results when there exist one or **multiple local maximum(s)**. To deal with all the conditions of data point distribution, a customized cluster merging strategy is proposed with the following three steps. Firstly, for each cluster \mathcal{C}_p , the average density μ_p and the standard deviation σ_p of all the data points in \mathcal{C}_p are calculated. Secondly, the adjacent clusters for each cluster \mathcal{C}_p are collected by searching for the border data points between adjacent clusters. For each data point \mathbf{p}_i in \mathcal{C}_p , its neighborhood set is denoted as \mathcal{I}_i . If a data point \mathbf{p}_j in \mathcal{I}_i belongs to another cluster \mathcal{C}_q , these two clusters are considered to be a pair of adjacent clusters, \mathbf{p}_i and \mathbf{p}_j are recorded as the adjacent points between \mathcal{C}_p and \mathcal{C}_q , respectively. Thirdly, for each adjacent cluster pair \mathcal{C}_p and \mathcal{C}_q , the average densities of the adjacent points of \mathcal{C}_p and \mathcal{C}_q are denoted as $\bar{\rho}_p$ and $\bar{\rho}_q$, respectively. These two adjacent clusters will be merged if the following conditions are met:

$$\bar{\rho}_p > \mu_q - \sigma_q \text{ and } \bar{\rho}_q > \mu_p - \sigma_p. \quad (3)$$

The cluster merging is conducted iteratively, which means that all the clusters that are directly or indirectly adjacent to the start cluster are merged.

Outliers: In the previous work presented by (Ester et al., 1996), the outlier points are the ones whose densities are smaller than a certain threshold. By this way, the low density data points may be classified as outliers. In the work of (Rodriguez and Laio, 2014) the outlier points are considered as those whose densities are small than the highest density in the border region of a cluster, which means that all the data points in the border region of a cluster are discarded as outliers. In our work, we consider the outliers on the cluster-level. If a data point \mathbf{p}_i whose density is local-maximal but smaller than the median density, $median(\rho)$, of all the data points, all the data points in the same cluster with \mathbf{p}_i are considered as outliers.

Figure 2 shows an illustration of some basic ideas of the proposed P-Linkage method, from which we can observe that there are two clusters in blue and red, respectively. For each data point, the pairwise linkage is formed by searching for the closest neighboring point whose density is greater than its own. For example, \mathbf{p}_8 is first linked to \mathbf{p}_{27} , \mathbf{p}_{27} is then linked to \mathbf{p}_4 , \mathbf{p}_4 is further linked to \mathbf{p}_{13} , and \mathbf{p}_{13} is finally linked to \mathbf{p}_1 with the greatest density in its neighborhood. In this way, the complete linkage is found as $\mathbf{p}_8 \rightarrow \mathbf{p}_{27} \rightarrow \mathbf{p}_4 \rightarrow \mathbf{p}_{13} \rightarrow \mathbf{p}_1$, and thus all of these five data points are classified into the same cluster whose cluster center is \mathbf{p}_1 . The same procedure occurs for all the data points in blue as a separate cluster. Similarly the red cluster can be formed by this way. As to the data points on the boundary between two clusters, the pairwise linkage can still be applied. Taking the data point \mathbf{p}_{33} for example, \mathbf{p}_{19} , \mathbf{p}_7 , \mathbf{p}_6 , and \mathbf{p}_{28} are its neighboring points, \mathbf{p}_6 is its closest neighboring point (CNP), and thus \mathbf{p}_{33}

is classified into the blue cluster, which is quite reasonable. The four data points in black, \mathbf{p}_{26} , \mathbf{p}_{18} , \mathbf{p}_{17} , and \mathbf{p}_{16} , are classified as outliers because there exist no CNP in their neighborhood, and the densities of their cluster centers are not high enough neither.

As a summary, the proposed clustering method can discover the clusters and cluster centers in only one step in general cases without the merging procedure. For each data point \mathbf{p}_i with a density ρ_i , we find its closest neighboring point $CNP(\mathbf{p}_i)$ whose density is greater than that of \mathbf{p}_i , and classify the point \mathbf{p}_i to the same cluster as $CNP(\mathbf{p}_i)$. If the density ρ_i of the data point \mathbf{p}_i is local-maximal and greater than the average density $\bar{\rho}$, we consider \mathbf{p}_i as a cluster center. Algorithm 1 describes the complete procedure in details of the proposed P-Linkage clustering method.

Algorithm 1 Hierarchical Clustering by Pairwise Linkage

Require: The density of each data point; the cutoff distance d_c .

Ensure: The clusters \mathcal{C} ; their cluster centers \mathcal{C}_{center} .

```

1:  $\rho_i$ : the density of a data point  $\mathbf{p}_i$ 
2:  $\bar{\rho}$ : the average density of all the data points
3:  $\mathcal{I}_i$ : the neighborhood set of a data point  $\mathbf{p}_i$ 
4:  $\mathbb{T}$ : the lookup table recording all the pairwise linkages
5: for each data point  $\mathbf{p}_i$  do
6:   Set  $LocalMaximal \leftarrow \text{TRUE}$ 
7:   Set  $d_{\min} \leftarrow \infty$  and  $CNP(\mathbf{p}_i) \leftarrow \emptyset$ 
8:   for each neighboring point  $\mathbf{p}_j$  in  $\mathcal{I}_i$  do
9:     Set  $d_{ij} \leftarrow$  the distance between  $\mathbf{p}_i$  and  $\mathbf{p}_j$ 
10:    if  $\rho_j > \rho_i$  and  $d_{ij} < d_{\min}$  then
11:      Set  $LocalMaximal \leftarrow \text{FALSE}$ 
12:      Set  $CNP(\mathbf{p}_i) \leftarrow \mathbf{p}_j$ ,  $d_{\min} \leftarrow d_{ij}$ 
13:    end if
14:  end for
15:  if NOT  $LocalMaximal$  then
16:    Record the linkage between  $\mathbf{p}_i$  and  $CNP(\mathbf{p}_i)$  into  $\mathbb{T}$ 
17:  else if  $LocalMaximal$  and  $\rho_i > \text{median}(\rho)$  then
18:    Insert the data point  $\mathbf{p}_i$  into  $\mathcal{C}_{center}$ 
19:  end if
20: end for
21: Collect the clusters  $\mathcal{C}$  by searching data points in the lookup
    table  $\mathbb{T}$  from each data point in  $\mathcal{C}_{center}$ 

```

3. P-LINKAGE FOR POINT CLOUD SEGMENTATION

The segmentation of point clouds can also be formulated as a clustering problem because the data points on a small surface often share the similar normal value. Thus we can employ the proposed P-Linkage clustering method on the segmentation of point clouds, which differs from that on the 2D data points in three aspects: (1) the neighborhood is based on the K nearest neighbors (KNN) instead of the fixed distance neighbors; (2) the feature value is the flatness of the estimated surface instead of the density of neighbors; (3) the distance of two data point is measured as the deviation of their normal orientations instead of their Euclidean distance. In the following subsections we will explain the P-Linkage based point cloud segmentation algorithm in details.

Normal Estimation: The normal for each point is estimated by fitting a plane to some neighboring points. This neighborhood can be specified in two different methods: K nearest neighbors (KNN) based and Fixed distance neighbors (FDN) based. For each data point, the KNN based methods select the K points from the point clouds having the minimum distance to it as its neighborhood, which is usually achieved by applying space partitioning strategies like the k-d tree (Arya et al., 1998). The FDN

based methods (Toth et al., 2004) select all the points within a distance to each point, and thus the number of neighbors changes according to the density of the point clouds. Compared to KNN, the number of neighbors of FND is less in the areas of low density area, which may result in inaccurate estimation of the normals. In this paper, we employ the KNN method to find the neighbors of each data point and estimate the normal of the neighboring surface via the Principal Component Analysis (PCA). The procedure contains three following steps. Firstly, we build a k-d tree by applying the ANN library (Mount and Arya, 2010). For each data point \mathbf{p}_i , its K nearest neighbors (KNN) is found and recorded as \mathcal{I}_i which is sorted in ascending order according to their distances to \mathbf{p}_i . Secondly, for each data point \mathbf{p}_i , the covariance matrix is formed by the first $K/2$ data points in its KNN set \mathcal{I}_i as follows:

$$\Sigma = \frac{1}{K/2} \sum_{i=1}^{K/2} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T, \quad (4)$$

where Σ denotes the 3×3 covariance matrix and $\bar{\mathbf{p}}$ represents the mean vector of the first $K/2$ data points in \mathcal{I}_i . Then the standard eigenvalue equation:

$$\lambda \mathbf{V} = \Sigma \mathbf{V} \quad (5)$$

can be solved using Singular Value Decomposition (SVD), where \mathbf{V} is the matrix of eigenvectors (Principal Components, PCs) and λ is the matrix of eigenvalues. The eigenvectors \mathbf{v}_2 , \mathbf{v}_1 , and \mathbf{v}_0 in \mathbf{V} are defined according to the corresponding eigenvalues sorted in descending order, i.e., $\lambda_2 > \lambda_1 > \lambda_0$. The first two PCs \mathbf{v}_2 and \mathbf{v}_1 form an orthogonal basis which indicate the two dimensions of highest variability that defines the best fitted plane of the neighboring points in \mathcal{I}_i , the third PC \mathbf{v}_0 is orthogonal to the first two PCs, and approximates the normal of the fitted plane. λ_0 estimates how much the points deviate from the tangent plane which can evaluate the quality of a plane fitting, and the smaller the value of λ_0 the better the quality of the plane fitting.

For each data point, we first find its K nearest neighbors and calculate its eigenvectors via the first $K/2$ neighbors via PCA, and then take the eigenvector \mathbf{v}_0 as the normal and the eigenvalue λ_0 as the flatness of the estimated plane. After that, the **Maximum Consistency with Minimum Distance** (MCMD) algorithm (Nurunnabi et al., 2015) is employed to find the inliers and outliers, which is conducted as follows. First, the orthogonal distances $\{d_o^k\}_{k=1}^K$ for the K nearest neighbors of a data point \mathbf{p}_i to its estimated plane are calculated, which are collected as a set $\mathcal{N}_{OD} = \{d_o^k\}_{k=1}^K$. Then, the **Median Absolute Deviation** (MAD) is calculated as follows:

$$MAD = a \times \text{median}_{d_o^k \in \mathcal{N}_{OD}} |d_o^k - \text{median}(\mathcal{N}_{OD})|, \quad (6)$$

where $\text{median}(\mathcal{N}_{OD})$ is the median value of \mathcal{N}_{OD} and $a = 1.4826$ is set constant. The inliers, also known as the **Consistent Set** (CS), are those data points whose R_z scores:

$$R_z = \frac{|d_o^k - \text{median}(\mathcal{N}_{OD})|}{MAD} \quad (7)$$

are less than a constant threshold 2.5 (Nurunnabi et al., 2015). Thus for each data point \mathbf{p}_i , we obtain its normal $\mathbf{n}(\mathbf{p}_i)$, flatness $\lambda(\mathbf{p}_i)$ and Consistent Set $CS(\mathbf{p}_i)$.

Linkage Building: With the normals, flatnesses and CS s of all the data points, the pairwise linkage can be recovered in a non-iterative way, which is performed as follows. For each data point \mathbf{p}_i we search in its CS to find out the neighbors whose flatnesses are smaller than that of \mathbf{p}_i and choose the one among them whose normal has the minimum deviation to that of \mathbf{p}_i as $CNP(\mathbf{p}_i)$. If there exists $CNP(\mathbf{p}_i)$, a pairwise linkage between $CNP(\mathbf{p}_i)$ and \mathbf{p}_i is created and recorded into a lookup table \mathbb{T} . If the flatness

$\lambda(\mathbf{p}_i)$ of \mathbf{p}_i is the minimum one in its neighborhood and $\lambda(\mathbf{p}_i)$ is smaller than the following threshold:

$$th_\lambda = \bar{\lambda} + \sigma_\lambda, \quad (8)$$

where $\bar{\lambda}$ is the average value of the flatnesses of all the N data points, $\sigma_\lambda = \sqrt{\sum_{i=1}^N (\lambda(\mathbf{p}_i) - \bar{\lambda})^2 / N}$ is standard deviation of all the flatnesses, thus we take \mathbf{p}_i as a cluster center, and insert it into the list of cluster centers \mathcal{C}_{center} .

Slice Creating: To create the surface slices, the clusters \mathcal{C} are firstly formed by searching along the lookup table \mathbb{T} from each cluster center in \mathcal{C}_{center} to collect the data points that are directly or indirectly connected with it. The clusters whose numbers of data points are smaller than 10 will be removed as outliers. Then for each cluster \mathcal{C}_p a slice is created by plane fitting via the MCS method proposed by (Nurunnabi et al., 2015), which is an iterative procedure with the iteration number calculated as:

$$I_t = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^{h_0})}, \quad (9)$$

where h_0 is the size of the minimal subset of the data points in \mathcal{C}_p which equals to 3 for plane fitting, P is the probability of the event that there exists at least one case in all the I_t iterations that the random chosen h_0 minimal subset is outlier free, and ϵ is the outlier rate in \mathcal{C}_p which was set 50% for general cases. Then for each iteration in the MCS method, the following steps are performed: (1) First, h_0 data points are chosen randomly. (2) For the h_0 -subset, a plane is fitted via the PCA, and the orthogonal distance for all the data points in \mathcal{C}_p are calculated and recorded in \mathcal{N}_{OD} . (3) Then \mathcal{N}_{OD} is sorted in ascending order and the first h (h equals to half the size of \mathcal{C}_p) data points are chosen to form the h -subset. (4) Finally, the PCA is applied again on the h -subset to fit for a plane whose flatness λ_0 is added into the list of previous flatnesses, defined as the set $\mathcal{S}(\lambda_0)$. After the iterations, the $\mathcal{S}(\lambda_0)$ is sorted in ascending order, and the plane corresponding to the smallest flatness is chosen to be the best fitted plane of \mathcal{C}_p . Then the MCMD outlier removal method is applied to find out the inliers, also known as the Consistent Set (CS), to the best fitted plane in \mathcal{C}_p . Thus for each slice S_p , we obtain its normal $\mathbf{n}(S_p)$, flatness $\lambda(S_p)$ and Consistent Set $CS(S_p)$ in the same way as each data point.

Slice Merging: To obtain complete planar and curved surfaces which are quite common in the indoor and industry applications, a normal and efficient slice merging method is proposed, which is similar to the Cluster Merging procedure introduced in the P-Linkage clustering and contains the following steps. First, we search for the adjacent slices for each one, two slices S_p and S_q are considered adjacently if the following condition is satisfied:

$$\begin{aligned} \exists \mathbf{p}_i \in CS(S_p) \text{ and } \mathbf{p}_j \in CS(S_q), \\ \text{where } \mathbf{p}_i \in CS(\mathbf{p}_j) \text{ and } \mathbf{p}_j \in CS(\mathbf{p}_i). \end{aligned} \quad (10)$$

Then, for a slice S_p and one of its adjacent slice S_q , they will be merged if the following condition is satisfied:

$$\arccos |\mathbf{n}(S_p)^\top \cdot \mathbf{n}(S_q)| < \theta, \quad (11)$$

where $\mathbf{n}(S_p)$ and $\mathbf{n}(S_q)$ are the normals of S_p and S_q , respectively, and θ is the threshold of the angle deviation. The greater θ is, the more curving the surface can be. The slice merging is conducted iteratively, which means that all the slices that are directly or indirectly adjacent to the start slice will be processed.

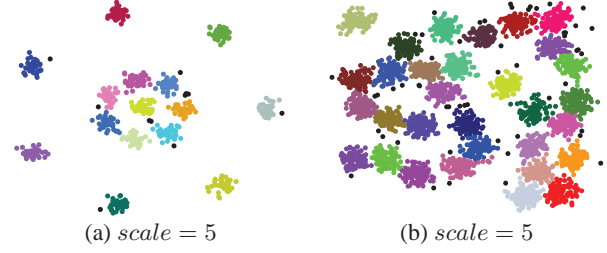


Figure 3: Clustering results of the proposed method on the Gaussian distributed test data.

4. EXPERIMENTAL RESULTS

4.1 Evaluation on Clustering

The only customized parameter the P-Linkage requires is the *scale* used to determine the value of d_c . We set *scale* = 5 for general cases, which will be proved in the following experiments to be a good choice. The merging procedure, which is specifically designed for the uniform distribution data, is not in use in general case but can be customized to get a better result if the initial clustering result contains too much fragments.

To evaluate the proposed method on the Gaussian distribution data, we chose two test data in the “Shape sets”¹, which contain 15 and 31 clusters, respectively. Figure 3 shows the clustering results on them, where the black points stand for the outliers, and the points in other different colors belong to different clusters. We can see that, in both data sets, all the clusters were correctly discovered, and the outliers extracted by the proposed method were all reasonable to be picked out. In both cases, the value of *scale* was set as 5.

To further evaluate the proposed method on the uniformly distribution data, the two other test data in the “Shape sets” were tested. Figure 4 shows the clustering results of our method. In Figure 4(a) all the 7 clusters were discovered and distinguished well. While in Figure 4(b), the left-down data points in purple were all classified into a single cluster, the reason is that the *scale* = 5 is too large for this case. By reducing *scale* = 2 and applying the merging procedure, a better clustering result can be achieved in Figure 4(c), however in this condition more data points were discarded as outliers (points in black) because there are not enough data points in its neighborhood thus the densities are smaller than the density threshold to be a new cluster.

We also tested the P-Linkage method on multi-dimensional data sets to see its robustness. Figure 5 shows the clustering result on a simulated 3D data which is composed of three subsets in Gaussian distribution. Each subset contains 200 data points. The number of correct classified data points for each subset is 175 (87.5%), 195 (97.5%) and 191 (95.5%), respectively. In all the three 2D views in Figure 5, we can see that all the three subsets were divided well.

4.2 Evaluation on Point Cloud Segmentation

Unlike the traditional K-Means algorithm and the approach proposed by (Rodriguez and Laio, 2014), the proposed P-Linkage clustering can be employed directly on the applications like point cloud segmentation where there exist a huge amount of clusters of data points in complex scenes. To test the robustness of the proposed point cloud segmentation method, we applied it on Vehicle-mounted, Aerial and Stationary Laser Scanner point clouds. In all

¹<http://cs.uef.fi/sipu/datasets/c>

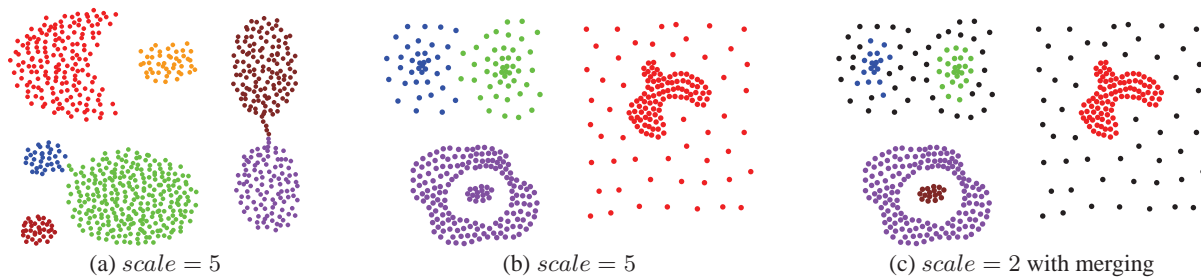


Figure 4: Clustering results of the proposed method on the uniformly distribution test data.

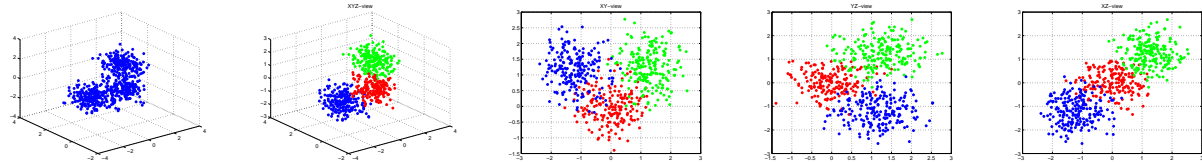


Figure 5: Clustering results of the proposed method on the 3D (XYZ) simulated test data: the original data points and the clustering result projected onto the XY plane, the YZ plane, and the XZ plane from left to right.

these tests, only the parameter θ was adjusted to get the best segmentation results without any filtering or diluting.

Vehicle-mounted: The segmentation of dense vehicle-mounted laser scanner points is a challenging task due to the existence of varied kinds of road furnitures which contain signs and light poles, road barricades, billboards, the ground and vehicles. In this work, two vehicle-mounted datasets were tested as shown in Figure 6(a) captured from an urban street of 355 meters long with 217k points and in Figure 6(b) captured from a small partial of a city in details containing 120k points. From Figure 6(a) containing road, buildings, street lamps, vehicles and trees, we observe that the road surface was clustered into a complete one and separated entirely from other objects. Also, the building facades were discovered quite well. From Figure 6(b), we observe that most of the building facades were segmented well despite that their densities vary in a wide range. Those facades perpendicular to the road (the green slice within the big red ellipse frame) and the small slices connected with other big ones (the two slices within the small red ellipse frames) were recovered quite well. Figure 6(c) shows a detailed look of the segmentation result with the red quadrangles representing different slices, which means that the segmentation result of the proposed method can be applied on the extraction of street patches with more specific operations.

Aerial: The first aerial data set tested is composed of 3433k points, which covers an urban area of $5\text{km} \times 5\text{km}$. There are buildings, ground, road and vegetations in this data set. Figure 7(a) shows a partial result of the whole data set, from which we can see that the ground was separated by the road (left bottom, in orange) into two parts, in purple and lemon yellow, respectively, and the ground in purple was segmented into a whole surface despite of the various objects on the ground. The roofs of buildings were segmented into a whole part in general, while some small structures can also be kept. Figure 7(b) shows a detailed view of the segmentation result of the area in red frame in Figure 7(a), from which we can see that the proposed method can preserve the details well. The second tested aerial data set is the ISPRS commission III/4 benchmark on Urban Classification and 3D Building Reconstruction and Semantic Labeling², the results of which are shown in Figure 7(c) and (d). We can see that the objects including the trees and building are generally separated from the ground which is segmented into a single part, and the details of the roofs are preserved well as Figure 7(d) shows.

²<http://www2.isprs.org/commissions/comm3/wg4/tests.html>

More experiments to evaluate the accuracy and correctness of the segmentation method will be presented in the future work. What is noteworthy is that to achieve the segmentation results on both data sets, only the nearest neighbors size K in k -d tree building and the θ in slices merging is adjusted.

Stationary: The tested stationary laser scanner data set consists of 2500×1076 data points, which means it can be unfolded into a 2D image whose column equals 2500 and row equals 1076. Figure 8 shows the segmentation result on a 2D image, from which we can see that all the main surfaces were segmented well. Specifically, we notice that the details of the modulator tubes on the ceiling were preserved quite well. In this case we set $\theta = 20.0^\circ$ due to that there exist many streamlined objects in this indoor scene, which will result in more surfaces and less planes.

5. CONCLUSION

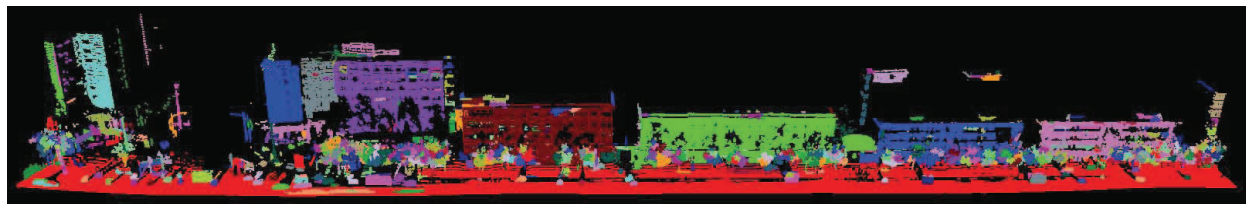
In this paper we propose a novel hierarchical clustering method named P-Linkage to discover the clusters in a simple and efficient way by recovering the pairwise linkages on the data point level. The proposed P-Linkage clustering can be employed directly on the applications which have huge amount of clusters and complex scene. Applying the P-Linkage clustering on point cloud segmentation, we develop an efficient point cloud segmentation algorithm which can handle a huge amount of data points captured from different scenes. Experimental results on different dimensional synthetic data from 2D to 4D sufficiently demonstrate the efficiency and robustness of the proposed P-Linkage clustering algorithm, and a large amount of experimental results on the Vehicle-Mounted, Aerial and Stationary Laser Scanner point clouds sufficiently illustrate the robustness and efficiency of our proposed 3D point cloud segmentation algorithm.

ACKNOWLEDGMENT

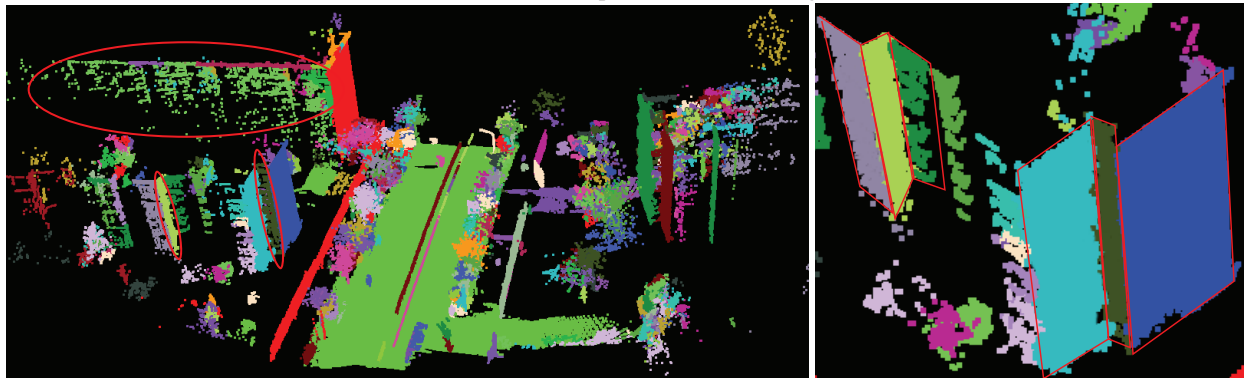
This work was supported by the National Natural Science Foundation of China (Project No. 41571436), the National Basic Research Program of China (Project No. 2012CB719904), the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866), the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), and the South Wisdom Valley Innovative Research Team Program.

REFERENCES

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y., 1998. An optimal algorithm for approximate near-



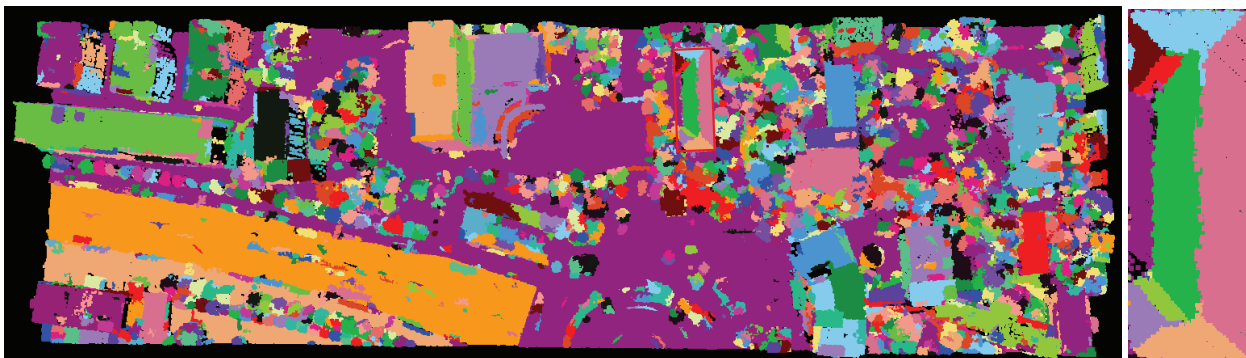
(a) an urban street with 217k points, $\theta = 10.0^\circ$, segments = 753



(b) an urban region with 120k points, $\theta = 10.0^\circ$, segments = 468

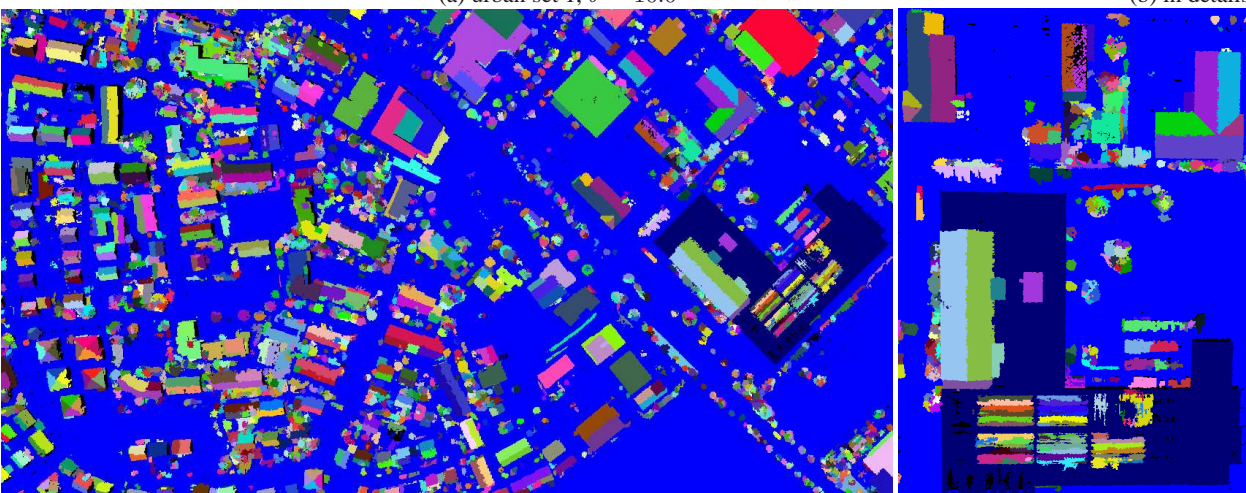
(c) local details of partial regions in (b)

Figure 6: Segmentation results of the proposed method on the vehicle-mounted test data.



(a) urban set 1, $\theta = 10.0^\circ$

(b) in details



(c) urban set 2, $\theta = 10.0^\circ$

(d) in details

Figure 7: Segmentation results of the proposed method on two aerial data sets.

est neighbor searching fixed dimensions. *Journal of the ACM (JACM)* 45(6), pp. 891–923.

Belton, D. and Lichti, D. D., 2006. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36(5), pp. 44–49.

Benkő, P. and Várady, T., 2004. Segmentation methods for smooth point regions of conventional engineering objects. *Computer-Aided Design* 36(6), pp. 511–523.

Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8(6), pp. 679–698.

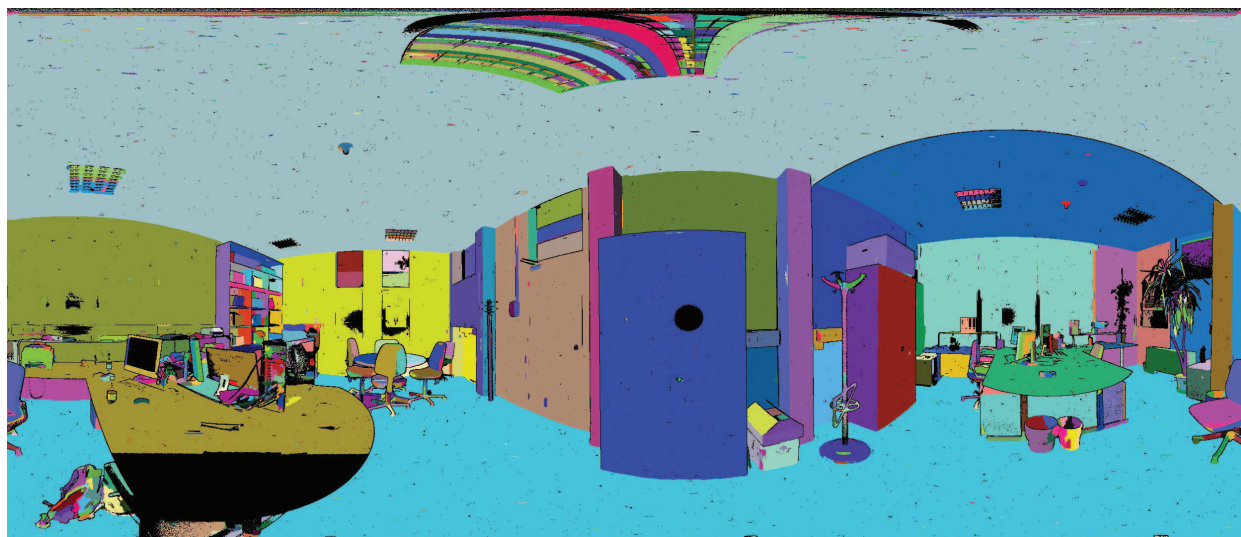


Figure 8: Segmentation results of the proposed method on the stationary indoor test data with $\theta = 20.0^\circ$.

- Castillo, E., Liang, J. and Zhao, H., 2013. Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. In: *Innovations for Shape Analysis*, Springer, pp. 283–299.
- Comaniciu, D. and Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), pp. 603–619.
- Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.
- Harati, A., Gächter, S. and Siegwart, R., 2007. Fast range image segmentation for indoor 3D-SLAM. In: *Intelligent Autonomous Vehicles*, Vol. 6number 1, pp. 475–480.
- Jagannathan, A. and Miller, E. L., 2007. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(12), pp. 2195–2204.
- Klasing, K., Althoff, D., Wollherr, D. and Buss, M., 2009. Comparison of surface normal estimation methods for range sensing applications. In: *IEEE International Conference on Robotics and Automation*, IEEE, pp. 3206–3211.
- Lavoué, G., Dupont, F. and Baskurt, A., 2005. A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design* 37(10), pp. 975–987.
- Liu, Y. and Xiong, Y., 2008. Automatic segmentation of unorganized noisy point clouds based on the gaussian map. *Computer-Aided Design* 40(5), pp. 576–594.
- MacQueen, J. et al., 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*.
- Mount, D. M. and Arya, S., 2010. Ann: A library for approximate nearest neighbor searching. <https://www.cs.umd.edu/~mount/ANN/>.
- Neubeck, A. and Van Gool, L., 2006. Efficient non-maximum suppression. In: *International Conference on Pattern Recognition (ICPR)*, Vol. 3, IEEE, pp. 850–855.
- Ng, R. T. and Han, J., 1994. Efficient and effective clustering methods for spatial data mining. In: *Proceedings of the 20th International Conference on Very Large Data Bases*.
- Nurunnabi, A., West, G. and Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognition* 48, pp. 1404–1419.
- Rabbani, T., van den Heuvel, F. and Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36(5), pp. 248–253.
- Rodriguez, A. and Laio, A., 2014. Clustering by fast search and find of density peaks. *Science* 344(6191), pp. 1492–1496.
- Sappa, A. D. and Devy, M., 2001. Fast range image segmentation by an edge detection strategy. In: *Third International Conference on 3-D Digital Imaging and Modeling*, IEEE, pp. 292–299.
- Sibson, R., 1973. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal* 16(1), pp. 30–34.
- Toth, C. D., O'Rourke, J. and Goodman, J. E., 2004. Handbook of discrete and computational geometry. CRC press.
- Vieira, M. and Shimada, K., 2005. Surface mesh segmentation and smooth surface extraction through region growing. *Computer Aided Geometric Design* 22(8), pp. 771–792.
- Wani, M. A. and Arabnia, H. R., 2003. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. *The Journal of Supercomputing* 25, pp. 43–62.
- Woo, H., Kang, E., Wang, S. and Lee, K. H., 2002. A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture* 42(12), pp. 167–178.
- Xiao, J., Zhang, J., Adler, B., Zhang, H. and Zhang, J., 2013. Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robotics and Autonomous Systems* 61, pp. 1641–1652.
- Yamauchi, H., Gumhold, S., Zayer, R. and Seidel, H.-P., 2005a. Mesh segmentation driven by Gaussian curvature. *The Visual Computer* 21(8-10), pp. 659–668.
- Yamauchi, H., Lee, S., Lee, Y., Ohtake, Y., Belyaev, A. and Seidel, H.-P., 2005b. Feature sensitive mesh segmentation with mean shift. In: *International Conference on Shape Modeling and Applications*, pp. 236–243.
- Zhang, X., Li, G., Xiong, Y. and He, F., 2008. 3D mesh segmentation using mean-shifted curvature. In: *Advances in Geometric Modeling and Processing*, Springer.