

Article

Effective Planar Cluster Detection in Point Clouds Using Histogram-Driven Kd-Like Partition and Shifted Mahalanobis Distance Based Regression

Jakub Walczak ¹, Tadeusz Poreda ² and Adam Wojciechowski ^{1,*}¹ Institute of Information Technology, Lodz University of Technology, 90-924 Łódź, Poland; jakub.walczak@edu.p.lodz.pl² Institute of Mathematics, Lodz University of Technology, 90-924 Łódź, Poland; tadeusz.poreda@p.lodz.pl

* Correspondence: adam.wojciechowski@p.lodz.pl

Received: 28 July 2019; Accepted: 18 October 2019; Published: 23 October 2019



Abstract: Point cloud segmentation for planar surface detection is a valid problem of automatic laser scans analysis. It is widely exploited for many industrial remote sensing tasks, such as LIDAR city scanning, creating inventories of buildings, or object reconstruction. Many current methods rely on robustly calculated covariance and centroid for plane model estimation or global energy optimization. This is coupled with point cloud division strategies, based on uniform or regular space subdivision. These approaches result in many redundant divisions, plane maladjustments caused by outliers, and excessive number of processing iterations. In this paper, a new robust method of point clouds segmentation, based on histogram-driven hierarchical space division, inspired by kd-tree is presented. The proposed partition method produces results with a smaller oversegmentation rate. Moreover, state-of-the-art partitions often lead to nodes of low cardinality, which results in the rejection of many points. In the proposed method, the point rejection rate was reduced. Point cloud subdivision is followed by resilient plane estimation, using Mahalanobis distance with respect to seven cardinal points. These points were established based on eigenvectors of the covariance matrix of the considered point cluster. The proposed method shows high robustness and yields good quality metrics, much faster than a FAST-MCD approach. The overall results indicate improvements in terms of plane precision, plane recall, under-, and the over-segmentation rate with respect to the reference benchmark methods. Plane precision for the S3DIS dataset increased on average by 2.6pp and plane recall- by 3pp. Both over- and under-segmentation rates fell by 3.2pp and 4.3pp.

Keywords: shifted Mahalanobis distance; hd-kd-tree; planes detection; point cloud segmentation; multivariate normal distribution

1. Introduction

Plane detection is a key aspect of a comprehensive point cloud analysis and segmentation. It is particularly important for indoor scenes, most of which can be easily decomposed into planar primitive shapes [1]. Point cloud processing (or its other form- depth map) is widely applied in such areas as reconstruction of human-made items, making inventories of architectural interiors, the documentation of cultural heritage structures [2], roofs detection in outdoor scans [3], or even for driver drowsiness estimation [4] and autonomous robot control [5]. It is also successfully used in civil engineering for deformation analysis [6]. Plane detection might also be applied to LIDAR-based rooftop reconstruction [7,8] and for primitive compression purposes [9], by storing shapes in a form of mathematical formulas rather than dense sets of points. The segmented points represent, among others, planes sampled during the scanning process. The problem of accurate and efficient plane detection, or more precisely, of extracting planar point clusters from an unorganized point cloud, has

been widely discussed in the literature. In this research, attention was paid to the area of model-fitting- based on plane detection methods for indoor scans, since this group of methods is currently the most successful [10–13]. Though thoroughly studied, model-based approaches still face generic problems related the difficulty in modelling the outlying points and performance in point clouds segmentation [14,15]. The attention was paid to indoor scans which similarly to all human-made structures may be usually reliably decomposed into geometrical primitives, particularly planes.

This paper introduces a new model-fitting based method for indoor scans, relying on Mahalanobis distance (MD) and histogram-driven kd-like point cloud division. The hierarchical, well-balanced point cloud subdivision process enables a shallow yet sufficient partition, preventing the adjacent planar point sets from unintended splitting. The proposed strategy of adapting MD for determination of the set of core points, especially around corners and edges, results in a robust plane model fitting, ensuring higher resistance to outlying points and better than the state-of-the-art precision methods in detecting planar clusters.

2. Related Works

2.1. Space Organization

Each point cloud segmentation method, regardless of the category it belongs to, may be characterized in terms of a space subdivision strategy.

Very frequently, one may encounter classical, direct and unorganized, sparse representation whereby each point is considered individually. This approach has become popular mainly for region-growing based methods.

Another method of space organization is uniform grid subdivision. It relies on the construction of a grid of equal, fixed-size cells which are subjected to further processing. This uniform subdivision assures better processing performance for large point clouds and easy determination of the adjacent point subsets. However, the subdivision is only as precise as the cell size, and smaller point clusters need to be manually analysed. This approach can be found, for instance, in studies by Li et al. [10], Xiao et al. [16], and Douillard et al. [17].

Another approach to point cloud organization involves a hierarchical representation. In the majority of cases, octree structures [18–20] may be encountered. This hierarchical approach is superior in many aspects to the sparse and uniform representations. Hierarchical description allows an algorithm to force or prevent further space divisions, driven by the structure of a point cloud. The hierarchical octree structure used in point cloud analysis was introduced by Meagher in 1982 [21]. Assumed to work in $\mathbb{R}^3$ space, a scene is enclosed in a cube, which is further divided into eight child cubes of equal size. However, the octree structure is not perfect. It produces cubes regardless of the distribution of points. In the presence of noise or in the case of uneven point density, the method produces a poorly balanced, very deep tree structure, which brings few benefits from hierarchical space organization. Additionally, the deterministic point cloud partition needs to correctly deal with the border points.

Reference [22] describes a semi-hierarchical subdivision. The authors subdivide the space into supervoxels whose dimensions decrease at each successive level. Such an approach exploits partially hierarchical structure, but a solution is limited to the cell maximum and minimum sizes, fixed at the beginning. Furthermore, the space is in fact much more granular due to rejection of some points and the non-planar supervoxels.

Another hierarchical structure is kd- tree based [23] space partitioning. It is widely used for point cloud registration purposes; namely, for retrieving nearby model points and vicinity determination [24–26] as well as for retrieving of the nearest neighbours. A kd- tree performs an equinumerous point cloud division in orthogonal directions indicated by the coordinate system axes. The way the space is subdivided by the kd-tree keeps balance because of the equinumerous division and intuitively enables detection of larger planar fragments. The division value of the kd-tree is

normally selected as a median along the considered dimension. It was used by Liu et al. in [27], who combined a kd-tree partition with a self-organizing fuzzy k-means clustering. They applied kd-tree to search for the nearest neighbours in order to model point set outliers based on the average point-to-point distance and the standard deviation. The authors made use of the clustering method and Ransom Sampling Consensus (RANSAC) to find appropriate planar patches. In fact, in [27] kd-tree was used for its dedicated purposes—nearest neighbours extraction. However, the authors present neither qualitative nor quantitative results of their studies. The validation dataset was not described either.

Concerning space partitioning, structure of a kd-tree itself might not be an optimal choice, unless division respecting points distribution is conducted. Although equinumerous divisions provides a well-balanced structure, selecting median value to construct division plane is not always a favourable choice since it may lead to many redundant subdivisions.

2.2. Plane Model Fitting

Hitherto mentioned point cloud subdivision methods produce points clusters, which should undergo planar regression. Taking into account the way the regression is processed, three main groups can be distinguished.

The first group involves transforming the data into model parameters space, which is referred to as a Hough transform [28]. In its pure form, in its generalization [29], or variants [30], Hough transform is a time demanding procedure, which does not deal well with noise [31]. Furthermore, the parameters space grid grows exponentially with a number of parameters.

Random Sampling Consensus (RANSAC) is a well known and widely used method for robust model fitting [10,11,32]. This approach is robust when considering the outliers, however, it does not handle noise (disturbances in points which may fit to a model). Furthermore, a fixed value of tolerance for inliers may be difficult or impossible to determine in advance; therefore, as Nurunnabi et al. observed, fixed threshold is one of the major problems of RANSAC based solutions [33].

Several methods relying on an effective estimation of the centroid and the covariance matrix for plane model fitting may be found in the literature. A solution of eigen-decomposition of a covariance matrix Σ , (Equation (1)) supplies three vectors, where the two longest indicate directions in which the plane spreads, whereas the shortest may be thought of as a plane's normal vector.

$$\Sigma = \frac{1}{n} \sum_i^n (\mathbf{p}_i - \mu)(\mathbf{p}_i - \mu)^T \quad (1)$$

where n is a cardinality of a point cloud $\mathcal{D} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_n\} (|\mathcal{D}| = n)$, μ is a centroid of the point set $\mathcal{D}$, defined as in Equation (2).

$$\mu = \frac{1}{n} \sum_i^n \mathbf{p}_i \quad (2)$$

To obtain a full plane equation, a point belonging to that plane has to be selected. In a perfect case, where all points are indeed sampled from a perfect plane, any point may be chosen for that purpose. However, because of the presence of noise in real cases, an average point is calculated as “the most representative” of a plane—centroid $\mu = [x, y, z]$, (Equation (2)). This strategy is usually referred to as the Principal Component Analysis (PCA). However, PCA works under strong assumptions:

1. All points are sampled from the same ideal plane (which means the absence of any outlying points), and
2. Noise is symmetric so that its impact on the solution is minimised.

Whereas the second assumption may be justified with the central limit theorem [34] (which says that a number of random distributions will tend to form a bell curve—which is characterized by a

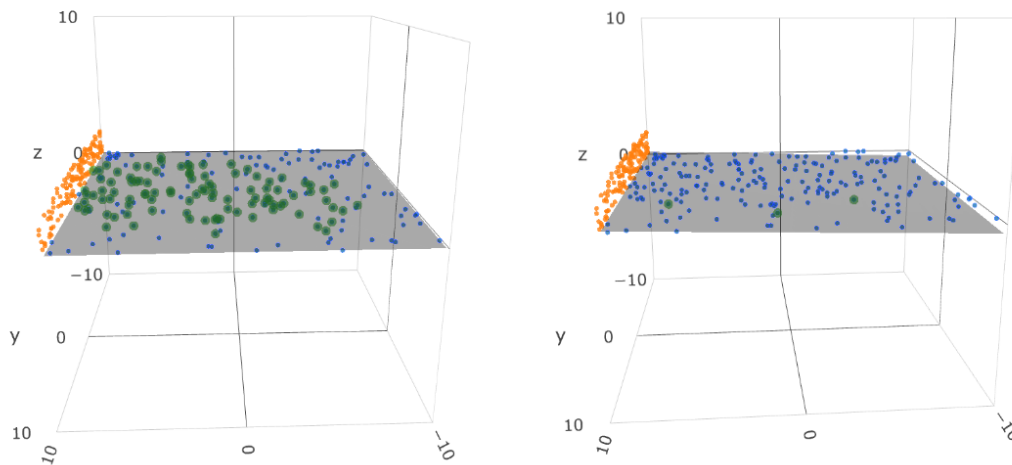
symmetric shape), the first one is difficult to defend since a considered set usually contains outlying points. This results in extra operations performed prior to the analysis.

In order to deal with the outlying elements, Hubert et al. introduced a robust version of PCA (ROBPCA) [35]. The authors determine a set of outliers by means of a modified Stahel–Donoho outlyingness [36,37] using a centroid and a covariance matrix rather than the median and the median absolute deviation.

As the authors pointed out, their method is extremely time demanding even for small datasets. In addition, the method proposed by Hubert et al. searches for global outliers, which as indicated by Nurunnabi et al. in [33] may cause unreliable and unexpected output in the case of point clouds.

To overcome this problem, Nurunnabi et al. proposed to reject outliers locally (per point) to make the vicinity of each point accurate and representative for the plane fitting purpose [33]. In order to achieve this goal, Nurunnabi et al. introduced an algorithm for determining the Maximum Consistency Set (MCS) [33]. It starts with the random selection of three points and fits a plane with PCA-based approach on that subset. Then, having calculated a point-plane distances for all neighbouring points, another subset of points with the smallest values of the Euclidean distances is selected and another plane is fitted. This procedure is carried out iteratively in a manner similar to the concentration step (C-step) in the FAST-MCD algorithm by Rousseeuw and Driessen [38], which minimizes the covariance determinant by iterative selection of a subset of points with the shortest Mahalanobis distance. Final plane parameters are those, which result in the smallest eigenvalue of the covariance matrix of the considered points subset.

Based on parameters of the plane computed for MCS, [33] proposed two methods to identify the outlying points. The first uses the measure of the Robust z-score (Rz-score) [33]. The second relies on the so-called, Robust Mahalanobis Distance (RMD), where rejection of the outliers is performed by screening out the points lying farther than 97.5-th quantile of the Chi-square distribution. Robustness of the methods is an effect of computing both a covariance matrix and a centroid based on the Maximum Consistency Set (a set with removed outliers) rather than the entire collection. Robust Mahalanobis Distance (RMD) is used solely for a slight reduction of the noise in the considered set of points. The authors reported accuracy exceeding both ROBPCA and RANSAC algorithms. In spite of quality improvements and reduction of the calculation time, the method presented in [33] leaves some areas for improvement. The first problem concerns determination of the vicinity of the points. Whereas the authors suggested using the kNN strategy, they did not indicate how to determine a value of k . This is a crucial issue, since the size of the vicinity influences determination of the MCS. Moreover, selection of the points whose vicinity contributes to more than one plane may produce an unexpected Maximum Consistency Set (Figure 1). As may be seen therein, for point clusters representing plane cross-sections, both methods selected points from the expected set. However, in case of the MCS (similarly to RANSAC), three points were detected, which in the case of noise, badly approximate real plane (see estimated plane marked in dark grey in Figure 1). On the other hand, FAST-MCD considered more core points, better approximating estimated plane model (Figure 1), but it takes much more time than MCS.



(a) A set of core points determined with FAST-MCD (b) A set of core points determined with MCS

Figure 1. Example of the vicinity of the edge points. A set of core points (green dots) and an estimated plane (dark grey plane) determined with the MCS or the FAST-MCD method.

An attempt to solve the issue of an adequate neighbourhood choice may be found in the studies of Eckart et al. in [13] or this of Xu et al. [39]. The authors of [13] dealt with point cloud registration rather than plane detection, they presented an interesting way of space partition which favours planar fragments. Eckart et al. employed a top-down recursive point cloud partition, making use of Gaussian Mixture Models (GMM) with an early stopping heuristic, which terminates procedure when the curvature of the points in a node C_λ , (Equation (3)) is beneath a threshold. This procedure adaptively sets proper division scale to produce efficient partition reducing oversegmentation.

$$C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}, \quad \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \quad (3)$$

where λ_1 , λ_2 , and λ_3 are eigenvalues of the covariance matrix of the considered points.

On the other hand, Xu et al. applied octree-based voxelization coupled with region growing. Such voxelization imposes a trade-off between details preservation and processing performance [39].

Yet the other strategy is represented by an algorithm of Dong et al. [22]. Their method processes partition and a hybrid region growing in order to reduce a number of candidate planes. The final stage is a global energy optimization procedure, which in fact is constructed by two nested optimization strategies: simulated annealing and an extension of the α -expansion algorithm introduced in [40]. Dong et al. reported high quality of their method: precision at the level of 90.4%, recall- 91.4% on the S3DIS [41] dataset. Unlike ROBPCA [35] or MCS [33], Dong et al. focused on high granularity of space partition rather than on rejection of the statistical outliers in order to construct robust planar fragments. This, in turn, significantly increases the processing time increase, accompanied by an excessive oversegmentation. Nonetheless, [22] reported the best results validated on the popular and recognized dataset (S3DIS), which makes it a reference method for precision comparison.

In contrast to the current leading approaches, the motivation for the presented strategy is to exploit MD for a selection of the core points rather than rejection of the outliers. Eigenvectors-driven shifts of a default centroid, supported with a selection of the MD core points, should result in a better and faster fitting of the plane model. According to our hypothesis, core points selected semi-deterministically, with the shortest MD, are more efficient than randomly selected model constructed by reducing the noise with the MD or FAST-MCD algorithm.

From the above review of the state-of-the-art, just a few authors [10,22] considered space partitioning simultaneously respecting point set characteristics and the fitting of the plane model.

These authors recommended other approaches than presented in this paper. In most cases, just one plane detection stages were thoroughly studied. However, none of the authors explicitly addressed the problem of excessive point cloud oversegmentation. Methods conducting space subdivision find nodes comprising corners and edges difficult to split efficiently. They are treated as many others non-planar, cluttered nodes and subdivided into smaller patches.

Our contribution relies on a better fitting of a plane model within non-planar point clusters, by means of a MD-based, selection of highly planar core points around six, adaptively shifted centroids. Preceding it, kd-tree like histogram-driven space partitioning method (hd-kd-tree), assures planarity-oriented point cloud partition and results in lower over-segmentation, which, in turn, increases time efficiency.

3. Proposed Method

The method of a point cloud planar clusters detection can be divided into four stages (Figure 2):

1. initial point cloud alignment (preprocessing);
2. histogram-driven point cloud partition;
3. planar patches robust refinement;
4. refined planar patches aggregation and post division;

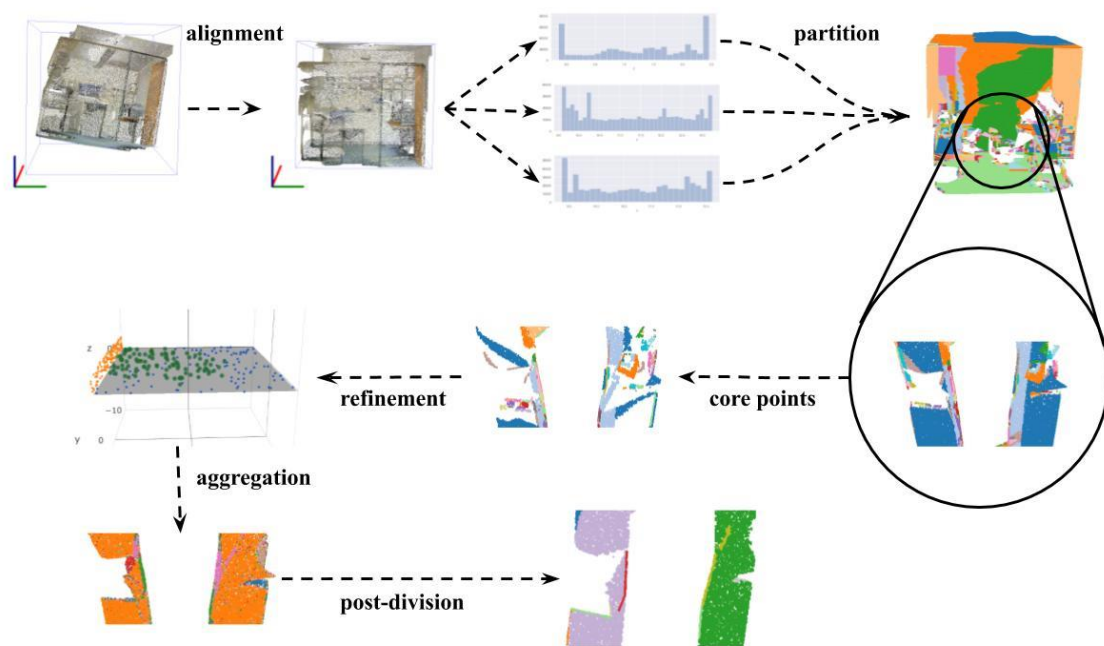


Figure 2. The architecture of the proposed algorithm.

3.1. Initial Point Cloud Alignment (Preprocessing)

Let a set $\mathcal{D}$ of n points $\mathcal{D} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_n\}$ with cardinality ($|\mathcal{D}| = n$) be defined. Each point $\mathbf{p}_i \in \mathcal{D}$ is defined as a tuple of three coordinates $\mathbf{p}_i = (p_i^{(x)}, p_i^{(y)}, p_i^{(z)})$. Often, a point cloud $\mathcal{D}$ may be misaligned with respect to an object inherent coordinating system (Figure 3a) due to several reasons. For instance, imperfect laser device placement during scene capture. If the dataset is oriented better with respect to the axes, it is very likely that at least some of the planes might be extracted without an unnecessary subdivision. Certainly, the benefits of this stage depend strongly on the point cloud, but in the case of interiors, it may be assumed that many elements are mutually orthogonal. Markiewicz used computer vision algorithms, based on features matching for successful point cloud orientation [42]; however, this method needs a reference model with respect to which orientation will be determined.

In the problem considered in this paper, such a reference model is not known in advance. Hence, it was decided to align the point cloud by minimizing the product of its dimensions.

Let a sample point cloud represents the interior of a room. Very often, the set is composed of mutually parallel or orthogonal planes, like walls, floor, ceiling, tables, and others. This knowledge might be used to optimize a way the space is partitioned. In fact, the dataset is supposed to be transformed so that the major planes are aligned with the coordinate system axes. It can be seen from Figure 3a that an indoor misaligned set usually has a larger Axis Aligned Bounding Box (AABB) than the properly oriented point cloud (Figure 3b) [43].

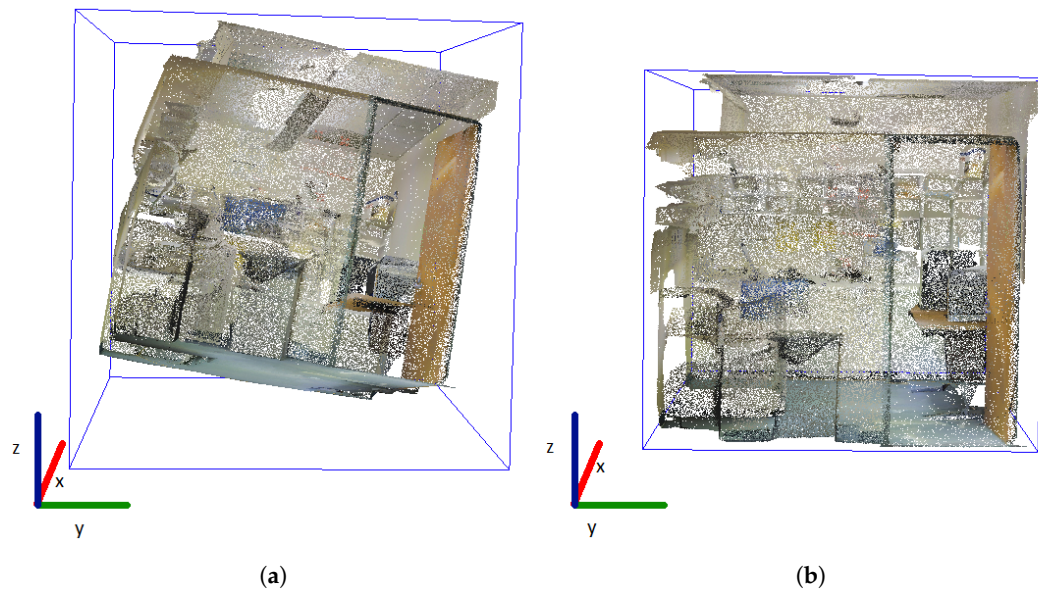


Figure 3. (a) Misaligned point cloud (b) aligned point cloud.

In view of the above, a proper alignment procedure can be formulated as an optimization task, where the objective function takes into account the AABB dimensions (Equation (4)).

$$\hat{R} = \arg \min_{\alpha, \beta, \gamma} (W_x \cdot W_y \cdot W_z) \quad (4)$$

where $\hat{R}$ is the resulting rotation matrix; W_x , W_y , and W_z are dimensions of the AABB ($x_{max} - x_{min}$, $y_{max} - y_{min}$, $z_{max} - z_{min}$) along the x , y and z axes respectively.

Because it is a trivial optimization task, one of the simplest, fast, gradient-free Nelder Mead method [44] was applied. As a result of this procedure, the optimal rotation matrix $\hat{R}$ is calculated. This matrix transforms a point cloud so that the volume of the AABB is minimized and a point cloud is better aligned with the coordinate system axes. This should enable extracting larger planar fragments, especially for indoor scenes.

3.2. Histogram-Driven Point Cloud Partition

The aim of this stage is to subdivide the considered point cloud $\mathcal{D}$ into major, inherently planar groups $\mathcal{O}_j : j \in \{1, 2, 3, \dots, m\}$ for which change of curvature (Equation (3)) is low and one additional group ($\mathcal{O}_{out}$) representing remaining points.

Kd-tree structure seems to be better-suited for extraction of the planar fragments than the octree. Therefore, it was selected for space partitioning. A basic kd-tree recurrently produces equinumerous sets of points resulting in a well-balanced tree. However, in the ordinary kd-tree building procedure, the division plane is supposed to be constructed based on a median value along the successive dimensions. Having obtained better oriented point cloud (in the stage 1 from Section 3.1), subdivision

separation values can be selected making use of histogram peaks of point coordinates in order to reduce the number of redundant subdivisions and to force favourable planar points concentration.

To do so, three histograms $\mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z$ ($\mathcal{H}_{x/y/z}$ for short), describing the distribution of the point coordinates along three axes x, y, z , are defined. It must be noticed that the distribution of points might be different for the three coordinating system directions.

Assuming that numbers of bins for x, y and z axes take values b_x, b_y , and b_z respectively, three sets of corresponding bin border values, B_x, B_y, B_z ($B_{x/y/z}$ for short), are calculated as in Equation (5).

$$B_{x/y/z} = \{\min \mathcal{D}^{(x/y/z)}, \min \mathcal{D}^{(x/y/z)} + 2 \cdot \epsilon_d, \dots, \min \mathcal{D}^{(x/y/z)} + 2 \cdot b_{x/y/z} \cdot \epsilon_d\} \quad (5)$$

where ϵ_d is an empirical coefficient of inlier plane tolerance threshold (Figure 4), discussed later in this section, determining the range of points constituting individual histogram bin.

Histograms $\mathcal{H}_{x/y/z}$ may be formally expressed as shown in Equation (6).

$$\mathcal{H}_{x/y/z}[\text{bin}] = ||\{\mathbf{p}_i | \mathbf{p}_i \in \mathcal{D} \wedge B_{x/y/z}[\text{bin}] \leq \mathbf{p}_i^{(x/y/z)} < B_{x/y/z}[\text{bin} + 1]\}\| \quad (6)$$

where $|| \cdot ||$ represents the cardinality of a point set constituting the selected *bin*.

A crucial value of each histogram is a number of bins. It should be sufficient to differentiate close parallel planes, yet large enough to encompass most of the planes described by the histogram bins. If ϵ_d is selected as an inlier-plane tolerance (Figure 4), then the number of bins b_x, b_y, b_z ($b_{x/y/z}$ for short), considered separately for each dimension x, y, z , might be defined using Equation (7).

$$b_{x/y/z} = \left\lfloor \frac{\max(\mathbf{p}_i^{(x/y/z)} \in \mathcal{D}) - \min(\mathbf{p}_i^{(x/y/z)} \in \mathcal{D})}{2 \cdot \epsilon_d} \right\rfloor \quad (7)$$

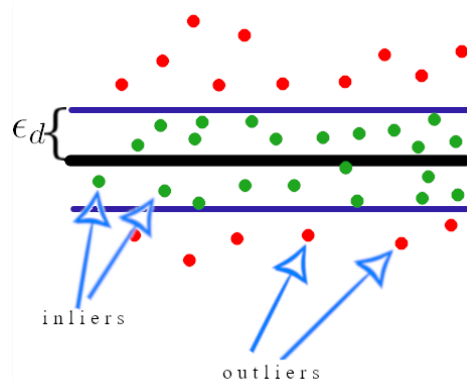


Figure 4. Inlier- plane tolerance.

The symbol $\lfloor \cdot \rfloor$ represents the floor function (integer part). An example of a histogram along the z axis for the selected (Area1/office-19) point cloud is presented in Figure 5.

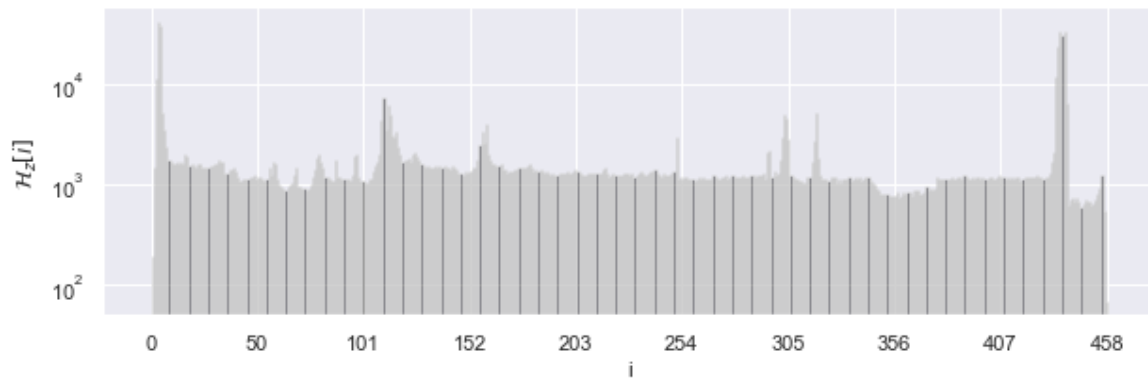


Figure 5. An example of a histogram for the OZ axis.

Histograms should be examined for bin height values as they reveal the concentration of points throughout the whole set $\mathcal{D}$. For instance, considering an interior scene, if the first and the last bin of one of the axes histograms contain the most points, this suggests that the large planar fragments are contained therein. Note that some high values of histograms' bins along other dimensions might also imply the presence of the bigger planes. Therefore, they might be considered as foundations for constructing point cloud division planes. It does not matter if the point cloud is rotated by 90° and a ceiling will be vertical whereas the walls will be horizontal because the semantic analysis is not applied for plane. Furthermore, even in the case of a poorly defined point cloud, the larger part of the planar fragments can be detected. Histogram construction is of linear complexity ($O(n)$).

Peaks of the histograms could indicate regions where larger planes are contained even though real planes might not necessarily be orthogonal to the coordinate system axes. With this in mind, the division planes are selected based on the inner border values ($\mathcal{B}_{x/y/z}$). The highest bin border value, closer to a value representing half of a point set dimension ($\min \mathcal{D}_{x/y/z} + \frac{W_{x/y/z}}{2}$), along the corresponding axis, is selected. Assuming a plane is defined as in Equation (8) and $\mathcal{H}_{x/y/z}[\text{bin}]$ is the highest histogram for the selected $z/y/z$ -coordinate, a division plane is calculated based on the associated value $\mathcal{B}_{x/y/z}[\text{bin}]$ as presented in Equation (9).

$$\pi : \mathbf{n} \cdot \mathbf{p} = -\rho \quad (8)$$

where $\mathbf{n}$ is a unit-length normal vector, $\mathbf{p}$ is an arbitrary point belonging to the plane, and ρ is the plane intercept, defined as a distance between that plane and the origin of the coordinate system.

$$\pi = \begin{cases} \mathbf{n} \cdot \mathbf{p} = -\mathcal{B}_{x/y/z}[\text{bin} + 1] & \text{if } \mathcal{B}_{x/y/z}[\text{bin} + 1] \leq \min \mathcal{D}_{x/y/z} + \frac{W_{x/y/z}}{2} \\ \mathbf{n} \cdot \mathbf{p} = -\mathcal{B}_{x/y/z}[\text{bin}] & \text{otherwise} \end{cases} \quad (9)$$

Division of a point set with a plane (Equation (9)) produces two child nodes containing points being on the left (or exactly on) or right-hand side of that division plane. Planarity check is carried out by the analysis of the covariance matrix of points contained in each node. A good estimator of the planarity is a low curvature change, expressed with Equation (3). This determinant has been widely used by researchers [22,33,45,46].

Each subsequent plane subdivision is calculated based on the highest (or the first highest value if there are more), yet not considered bin, among all three histograms $\mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z$. Subdivision is continued until the stop condition is reached. Even though many planar point clusters in a point set $\mathcal{D}$ may not be orthogonal to the coordinate system axes, applying a histogram for points space partitioning reflects their distribution and may improve the result because of the extraction of larger planar fragments instead of their deterministic division.

There are two stopping conditions for the recurrent subdivision. The first stops recursive calls if a number of points, in a current node, is lower than the minimum ($\hat{n} = 20$ [22])—user-defined minimum

cardinality of a valid node). The second is the planarity condition $C_\lambda < \epsilon_C$ (Equation (3)), where ϵ_C is an assumed threshold, which says how many points are spread along the candidate normal vector ($\epsilon_C = 0.002$ in the presented method)

When the points planarity test in a node was passed, the set of points was appended to the set of patches $\mathcal{O}_j$ satisfying the planarity condition ($C_\lambda < \epsilon_C$). If not, the division plane is selected as indicated earlier in this section—based on the currently highest histogram peak. Original histograms are exploited throughout the whole partitioning process, in spite of the fact that the sub—histograms of the selected nodes (subsets of points histograms) may suggest different subdivision planes.

It may happen that all border values $\mathcal{B}_x, \mathcal{B}_y, \mathcal{B}_z$ were already used to construct division planes but some points remained because of failure of the planarity test. In this situation, the space subdivision follows the ordinary kd-tree partition procedure. It involved selecting as a division plane one based on the median of coordinate values along the current axis.

For clarity, the partition procedure is presented in Algorithm 1.

Algorithm 1 Histogram- driven partition algorithm

```

1:  $j \leftarrow 1$  ▷ output cluster index
2:  $dim \leftarrow 0$  ▷  $dim \in \{0, 1, 2\}$  for  $\{x, y, z\}$  axes
3:  $\mathcal{O}_{out} \leftarrow \emptyset$  ▷ cluster of outlying points
4: function PARTITION( $\mathcal{D}, \mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z, \mathcal{B}_x, \mathcal{B}_y, \mathcal{B}_z, \hat{n}, dim$ )
5:   if  $||\mathcal{D}|| < \hat{n}$  then
6:      $\mathcal{O}_{out} \leftarrow \mathcal{O}_{out} \cup \mathcal{D}$  return
7:   end if
8:   if  $\mathcal{D}$  is planar then
9:      $\mathcal{O}_j \leftarrow \mathcal{D}$ 
10:     $j \leftarrow j + 1$ 
11:    return
12:   end if
13:    $ids[0] \leftarrow \arg \max(\mathcal{H}_x)$ 
14:    $ids[1] \leftarrow \arg \max(\mathcal{H}_y)$ 
15:    $ids[2] \leftarrow \arg \max(\mathcal{H}_z)$ 
16:    $dim \leftarrow \arg \max([\mathcal{H}_x[ids[0]], \mathcal{H}_y[ids[1]], \mathcal{H}_z[ids[2]])$ 
17:    $id \leftarrow ids[dim]$ 
18:   if  $\mathcal{H}_{dim}[id] == -1$  then
19:      $division\_value \leftarrow \text{median}\{\mathbf{p}_i^{(dim)} \mid \mathbf{p}_i \in \mathcal{D}, i \in \{1, 2, 3, \dots, ||\mathcal{D}||\}\}$ 
20:   else
21:     if  $\mathcal{B}_{dim}[id + 1] \leq \min \mathcal{D}_{x/y/z} + \frac{W_{x/y/z}}{2}$  then ▷ Equation (9)
22:        $division\_value \leftarrow \mathcal{B}_{dim}[id + 1]$ 
23:     else
24:        $division\_value \leftarrow \mathcal{B}_{dim}[id]$ 
25:     end if
26:      $\mathcal{H}_{dim}[id] \leftarrow -1$  ▷ mark id-th bin when used
27:   end if
28:    $\mathcal{D}_L^{(1)} \leftarrow \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathcal{D}, \mathbf{p}_i^{(dim)} \leq division\_value\}$ 
29:    $\mathcal{D}_R^{(1)} \leftarrow \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathcal{D}, \mathbf{p}_i^{(dim)} > division\_value\}$ 
30:   PARTITION( $\mathcal{D}_L^{(1)}, \mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z, \mathcal{B}_x, \mathcal{B}_y, \mathcal{B}_z, \hat{n}, (dim + 1) \bmod 3$ )
31:   PARTITION( $\mathcal{D}_R^{(1)}, \mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z, \mathcal{B}_x, \mathcal{B}_y, \mathcal{B}_z, \hat{n}, (dim + 1) \bmod 3$ )
32: end function
  
```

Assuming the histogram is calculated once prior to the subdivision procedure, the procedure complexity is of the order $O(n \cdot \log n)$.

Obviously, the superiority of the proposed method is strictly tied with the presence of the orthogonal and parallel planes. If planes of other orientations exist, they will not be segmented entirely in a single iteration, but they are supposed to be divided into parts according to the assumed threshold of the histogram bins.

3.3. Planar Patches Refinement

The previous stage of the method gives a set of patches, retrieved from the histogram-driven kd-tree (hd-kd-tree) nodes $\mathcal{O}_j : j \in \{1, 2, 3, \dots, m\}$ satisfying the planarity condition, and the group of remaining points $\mathcal{O}_{out}$. The patches are roughly planar because the measure of curvature (Equation (3)) is a relative value rather than an absolute one. Hence, it may happen that the planarity condition is satisfied but some outlying points can still be contained therein (Figure 1). If so, these planar patches need additional processing to fit a plane properly.

Mahalanobis distance (MD) of a point set, whose sample covariance matrix is Σ and sample mean μ , is defined as in Equation (10).

$$MD(\Sigma, \mu, \mathbf{p}_j) = \sqrt{(\mathbf{p}_j - \mu)^T \Sigma^{-1} (\mathbf{p}_j - \mu)} \quad (10)$$

Usually, a standard MD-based rejection of outliers is performed by screening out points lying farther than the 97.5-th quantile of the Chi-square distribution. Since, MD indeed follows Chi-square distribution (χ^2) with the number of degrees of freedom set to 3 (because there are three variables: x, y , and z), outliers are the points being farther than $\sqrt{\chi_{3,0.975}^2} = 3.057$ from the centroid μ [33,35,47]. This works under the assumption of a Gaussian distribution of points, but this assumption may not be valid globally. Nevertheless, at a sufficiently small scale, it might be a good local estimate of a surface, as indicated by Stoyanov et al. in [26].

Even though a covariance matrix and a centroid are robustly estimated with FAST-MCD [35] or MCS [33] procedures, nodes containing points close to corners or edges may yield improper estimation where the crucial issue is to find points which actually belong to the main plane. Even though the randomness of MCS and FAST-MCD forces a result of the plane fitting to tend to a global optimum but to ensure it, many iterations may be required. Hence, in the proposed method, the stochastic space exploration, which is applied in the MCS and FAST-MCD procedures, was replaced by performing C-step with respect to, so called, cardinal points ψ_i . Eigenvectors become predominant directions of alternative centroid candidate (cardinal points) locations as they represent inherent distribution of the point variations. The cardinal points are calculated by shifting the default centroid of a points cluster along its positive and negative eigenvectors by distances resulting from an analysis of the variance. This leads to a fast and accurate result of plane fitting.

The first cardinal point (ψ_1) is a default centroid μ , which is a centre of the mass of a point set. However, the C-step calculated with respect to μ may converge to core points distributed across several planes in case of their unfavourable points distribution. Hence, C-step should be carried out with respect to relocated centroids (cardinal points) to verify convergence around other possible centroids.

The most promising directions, which indicate possible centroid shifts, are those indicated by the eigenvectors of the covariance matrix Σ . Even though Σ may be disturbed by outliers, eigenvectors would point either towards these outliers or towards candidate centroids of a set of core inliers (see Figure 6). As a result, one of the shifted centroids (cardinal points) might be better than the default one. Hence, the shifts are defined as appropriately scaled eigenvectors e_1, e_2 and e_3 of the covariance matrix Σ and their opposites ($-e_1, -e_2$ and $-e_3$). This results in seven cardinal points $\psi_{1,\dots,7}$ (in a 3D case there are six points shifted along positive and negative eigenvectors plus the default centroid).

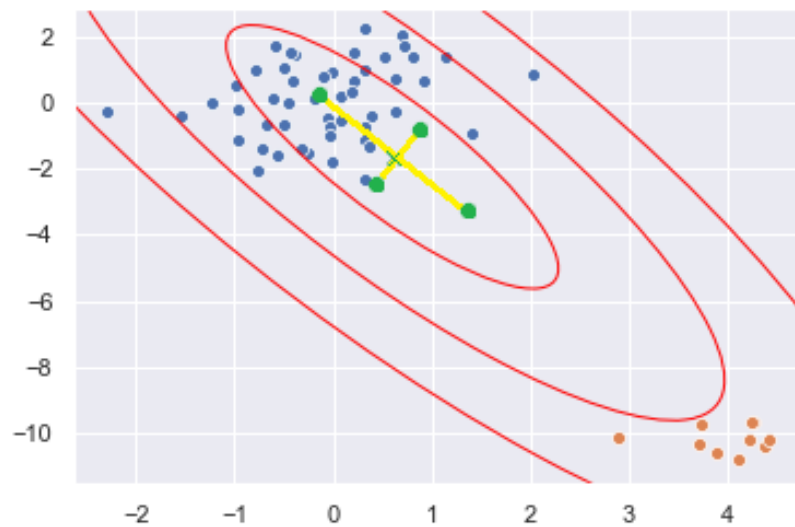


Figure 6. Cardinal directions of shifts (yellow lines) of the centroid (green cross) in 2D.

Next, the issue of a proper shift range has to be solved. It should be radical change of a default centroid position in order to force C-step to verify also boundary regions of the sub-clusters. Let a portion $h = 50\%$, which provides the highest breakdown point [48] be used for C-step. Then, knowing that eigenvalues are variances along the eigenvectors [13], the other half of a set—according to a univariate Gauss distribution—is contained outside the ranges $[-0.68 \cdot \sqrt{\lambda_1}, 0.68 \cdot \sqrt{\lambda_1}]$, $[-0.68 \cdot \sqrt{\lambda_2}, 0.68 \cdot \sqrt{\lambda_2}]$, $[-0.68 \cdot \sqrt{\lambda_3}, 0.68 \cdot \sqrt{\lambda_3}]$ along the corresponding eigenvectors e_1, e_2 , and e_3 . Hence, seven cardinal points may be defined as shown in Table 1.

Table 1. Seven cardinal points ψ_i . μ is the centroid of the contaminated dataset; e_1, e_2 and e_3 are dataset eigenvectors corresponding to λ_1, λ_2 and λ_3 eigenvalues.

Cardinal Point	Adapted Centroids
ψ_1	μ
ψ_2	$\mu + 0.68\sqrt{\lambda_1} \cdot e_1$
ψ_3	$\mu - 0.68\sqrt{\lambda_1} \cdot e_1$
ψ_4	$\mu + 0.68\sqrt{\lambda_2} \cdot e_2$
ψ_5	$\mu - 0.68\sqrt{\lambda_2} \cdot e_2$
ψ_6	$\mu + 0.68\sqrt{\lambda_3} \cdot e_3$
ψ_7	$\mu - 0.68\sqrt{\lambda_3} \cdot e_3$

Having obtained seven cardinal points, C-step (concentration step) of FAST-MCD is performed to produce covariance with a minimum determinant with the assumed portion of $h\%$ of points ($h = 50\%$) with respect to the seven cardinal points $\psi_1, \dots, \psi_7$. In such a way, seven candidate subsets of points are obtained. The resulting core points set is that of which covariance determinant is minimal. The entire procedure making use of a shifted Mahalobis distance (SMD) for core points set determination is presented in the Algorithm 2. After a core portion of the inliers set is determined, ordinary PCA is performed to fit a plane to that set. As an output of this stage, a set of m efficiently fitted planes π_j , corresponding to the planar clusters O_j , are provided to the final points aggregation stage (Section 3.4).

Algorithm 2 Shifted Mahalanobis Distance (SMD) based core points determination of a set $\mathcal{O}_j$

```

1: function SMD( $\mathcal{O}_j, h, \text{MAX\_IT} = 5$ )
2:    $h \leftarrow \frac{||\mathcal{O}_j||}{2}$  ▷ usually 50% [48]
3:    $\Sigma \leftarrow \text{covariance}(\mathcal{O}_j)$ 
4:    $\mu \leftarrow \text{mean}(\mathcal{O}_j)$ 
5:    $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \leftarrow \text{eigendecomposition}(\Sigma)$ 
6:    $[\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7] \leftarrow \text{Table 1}$ 
7:    $\text{min\_det} \leftarrow \infty$ 
8:    $\mathcal{U} \leftarrow \emptyset$ 
9:   for  $\text{card\_pt}$  in  $[\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7]$  do
10:      $\text{portion} \leftarrow \text{choose } h \text{ points randomly from } \mathcal{O}_j$ 
11:      $\text{old\_det} \leftarrow \infty$ 
12:     for  $i < \text{MAX\_IT}$  do ▷ C-step
13:        $\Sigma \leftarrow \text{covariance}(\text{portion})$ 
14:       if  $|\Sigma| < \text{min\_det}$  then
15:          $\mathcal{U} \leftarrow \text{portion}$ 
16:          $\text{min\_det} \leftarrow |\Sigma|$ 
17:       end if
18:       if  $\text{abs}(|\Sigma| - \text{old\_det}) < \epsilon_\Sigma$  then
19:         break
20:       end if
21:        $\text{old\_det} \leftarrow |\Sigma|$ 
22:        $\text{dists} \leftarrow \text{MD}^2(\Sigma, \text{card\_pt}, \mathcal{O}_j)$  ▷ Equation (10) for the entire set  $\mathcal{O}_j$ 
23:        $\text{portion} \leftarrow \text{choose } h \text{ points with the smallest Mahalanobis distances in } \text{dists}$ 
24:     end for
25:   end for
26:   return  $\mathcal{U}$  ▷ core subset of points— $h\%$  of  $\mathcal{O}_j$ 
27: end function

```

3.4. Point Aggregation

The successive stage of the proposed method aims at assigning points p_i across the entire point cloud $\mathcal{D}$ to appropriate planes (π_j) determined in Section 3.3. The stage begins with calculating core points for the local vicinity of each point p_i making use of Algorithm 2. Based on those core points, the normal vector is estimated by fitting a plane to them. Vicinity of a point p_i , from which core points are selected, was set to the 7 nearest neighbours according to the method presented in [49].

Next, the global procedure of point-plane assignment is performed for the entire set $\mathcal{D}$. This is done by checking an angular deviation between the refined planes normal vectors $\mathbf{n}_{\pi_j}$ (Section 3.3) and each points' normal vectors ($\mathbf{n}_{p_i}$) estimated based on their vicinity. Points whose normal vector deviates not more than assumed ϵ_θ are assigned to the given plane (Equation (11)).

$$\arccos(\mathbf{n}_{\pi_j} \cdot \mathbf{n}_{p_i}) \leq \epsilon_\theta \quad (11)$$

where $\mathbf{n}_{\pi_j}$ is the robustly estimated normal vector of a j -th plane (π_j) and $\mathbf{n}_{p_i}$ is the robustly estimated normal vector of a point p_i .

while applying angular deviation threshold, many parallel planes could be joined together. To overcome this problem, density-based clustering was employed (for instance, k-means based density clustering [50] or HDBSCAN/DBSCAN [51]). HDBSCAN was used in the presented approach

because of its superior efficiency, reported by [51]. As an output of the method, a set of detected planar points clusters, assigned globally to robustly selected planes π_j , is obtained.

4. Methodology

4.1. Datasets

The notion of a *benchmark dataset* for planes detection task is not well established in the literature. Almost every reported method used a different dataset, including artificially generated ones. In [22] and [52], S3DIS [41] dataset was used. However, it contains points labelled with respect to objects' adherence (i.e., a chair, a table, a lamp) rather than the individual planes. On the other hand, Li et al. [10] made use of a laser scan (Room-1) from Rooms UZH Irchel dataset [53], in spite of the fact that it does not contain any labelled data. The S3DIS dataset shows much sparser point clouds density than in the case of rooms from UZH Irchel dataset. These datasets differ significantly in terms of accuracy, noise, scan shadows, cardinality, and scene complexity. Therefore, it was decided to use representatives from both of them to verify the proposed method on point clouds of varying nature.

The present study uses point clouds of the S3DIS dataset and the Room-1 point set [53] used by Li et al. [10] (Table 2).

Because for the Room-1 dataset no ground truth segmentation was provided, it was labelled manually. The ground truth segmentation of the S3DIS dataset, in turn, was manually modified to represent individual planes. An example of six point clouds from S3DIS and Room-1 datasets are presented in Table 2.

Table 2. Visualization of randomly selected six point clouds from tested S3DIS dataset plus additional Room-1 used by Li et al..

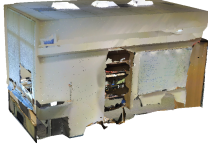
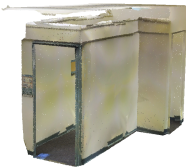
Point Cloud Name	Cardinality ($ \mathcal{D} $)	Image
Room-1	11,050,391	
Area6/office-19	515,366	
Area1/office-19	848,534	

Table 2. Cont.

Point Cloud Name	Cardinality ($ \mathcal{D} $)	Image
Area1/hallway-3	369,279	
Area4/conferenceRoom-2	1,653,935	
Area5/WC-1	719,348	
Area4/hallway-13	883,137	

4.2. Experiments

Experiments were conducted in two stages. At first, the space partition methods: PCP [54], octree [18–20], kd-tree [23,27], VCCS [22], and the proposed hd-kd-tree were examined. Secondly, planar clusters detection was assessed.

For the space partition juxtaposition, four values were presented: the division tree spreadness (number of all nodes in division tree), the final number of groups, the number of points, which remains after the partition process was accomplished, and the partition time. All space partition methods were tested on the S3DIS dataset with the same setup (Table 3).

Table 3. Setup for partition methods tests.

Method	Cardinality Threshold ($\hat{n}$)	Curvature Threshold (C_λ)	Min r	Max r	Δr
octree	25	0.002	-	-	-
kd-tree	25	0.002	-	-	-
pcp	25	0.002	-	-	-
VCCS	25	0.002	$5 \cdot \Omega$	$50 \cdot \Omega$	0.79433
hd-kd-tree	25	0.002	-	-	-

A decision to use quality measures applied by Dong et al. and Li et al. for planar cluster detection was made to clearly demonstrate the superiority of the proposed method. Hence, ordinary plane precision (Equation (12)), plane recall (Equation (13)) and over- as well as under- segmentation rates (Equations (14) and eqrefeq:usr) were used as the validity measures of the entire procedure.

$$PP = \frac{N_C}{N_S} \quad (12)$$

$$PR = \frac{N_C}{N_G} \quad (13)$$

where N_C stands for a number of correctly segmented planar clusters (with maximum overlapping strategy, 80%), N_S represents the total number of planar clusters obtained as the algorithm output, and N_G is a number of ground truth planar clusters.

$$OSR = \frac{N_O}{N_G} \quad (14)$$

$$USR = \frac{N_U}{N_G} \quad (15)$$

where N_U is a number of resulting planar clusters that overlap the multiple ground truth planar clusters. N_O , in turn, is a number of ground truth planar clusters overlapping more than one resulting planar cluster.

5. Results and Discussion

5.1. Space Partitioning Results

All methods used the same parameters values (see Table 3). For VCCS, remaining parameters (Min r , Max r , Δr) were taken from the studies of Dong et al. who used the method for the S3DIS dataset [22]. The symbol Ω represents an average distance between a point and its closest neighbourhood.

The results of the experiments are presented in Figure 7.

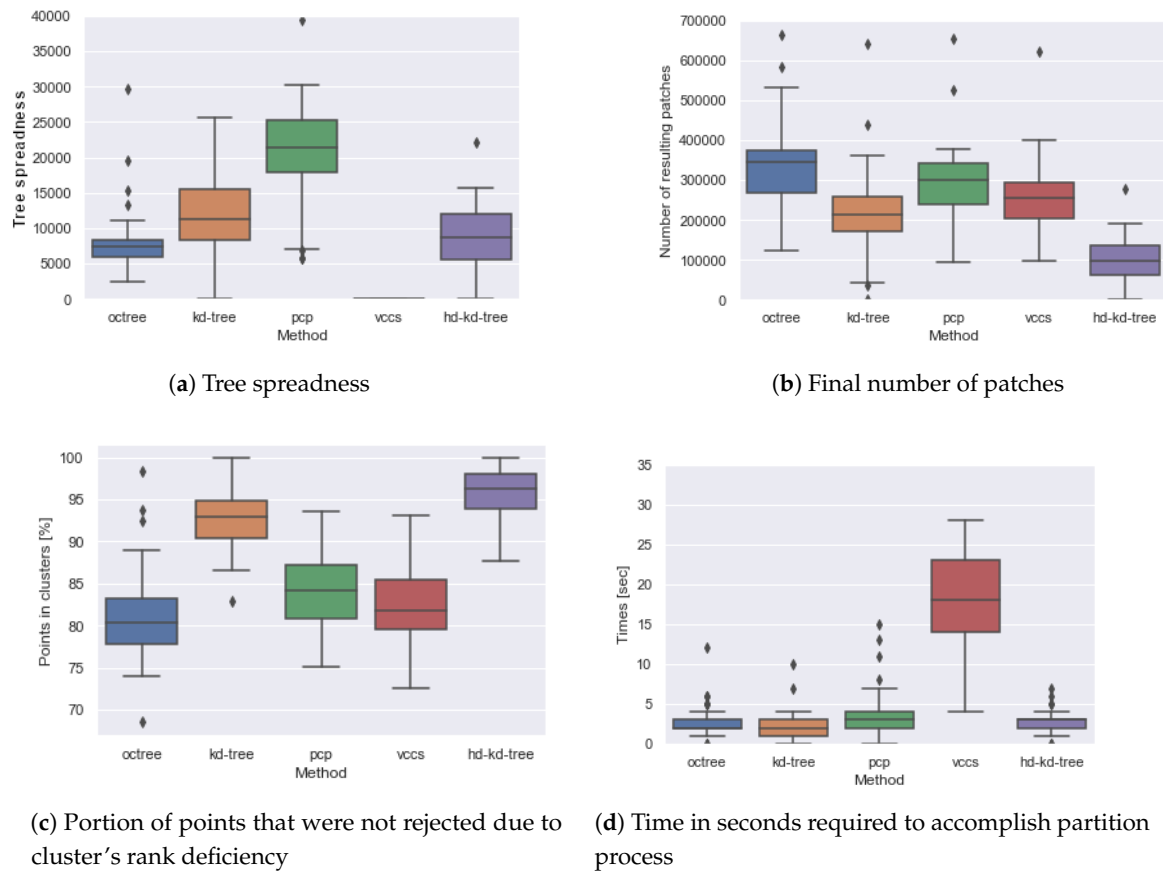


Figure 7. Juxtaposition of tree spreadness (the number of all nodes in tree), final number of clusters, portion of preserved points, and times for benchmark space partition algorithms and the proposed hd-kd-tree.

Figure 7a shows that the proposed method (hd-kd-tree) yields, on average, similar tree spreadness as the octree (close to 7500 nodes); however, with a higher variance. Both kd-tree and, especially PCP, produce much more spread trees. In case of VCCS, tree structure is always very flat (up to 10) due to constraints put on the partition (minimum supervoxel size- Min r, and maximum supervoxel size- Max r). However, in spite of the flat structure of the VCCS tree, Figure 7b shows that this tree has a very wide structure and produces many more clusters (260,000) than the hd-kd-tree strategy (100,000). The octree partition results in even more clusters than VCCS. Figure 7c clearly shows that the proposed method preserves significantly more points (on average, more than 95%) than the other methods. In this comparison, also ordinary kd-tree is superior versus octree, PCP, or VCCS. Figure 7d, in turn, shows time required to perform partition. All methods, except VCCS, show similar time demand (2.5 s). VCCS usually requires substantially more time (17.0 s).

5.2. Planar Patches Extraction Results

This section presents the results of the proposed method in comparison with the outcome of the approach of Dong et al. [22] and Li et al. [10], which are the most competitive state-of-the-art procedures.

The approach of Dong et al. partially exploits the advantages of a hierarchical partition; however, due to constraints put on the cell sizes at each level, it still introduces many redundant subdivisions.

Time complexity of these partition algorithms is presented below. The way Dong et al. divided the space suggests at least $O(n^2)$ time complexity because of the application of the region growing k-means algorithm, assuming supervoxels sizes to be constant and fixed in advance. The proposed partition algorithm has a complexity of the order $O(n \log n)$.

An outcome of the proposed method, for the test point clouds presented in Table 2, is shown in Figure 8. Note, that most of the planes were correctly detected both inside and outside the room.

Tables 4 and 5 present the results and comparison with those obtained by Dong et al. (Table 4) and by Li et al. (Table 5). The results of the SMD-based method results and the method of Li et al. [10] results are compared in Table 5. The results were obtained for Room-1 dataset [53].

In terms of PP and PR metrics, the experiments reveal that the SMD-based method outperforms (by 2.6pp, 3.0pp) the reliably documented method of Dong et al. Moreover, the newly developed points cloud histogram-driven partition is characterized by much lower over- and under-segmentation rates (by 3.2pp, 4.3pp). In comparison to the method of Li et al., the proposed method reaches higher plane precision (by 1.5pp) and slightly higher plane recall (by 0.2%). OSR and USR were not reported in [10].

Table 4. Comparison of the proposed method (SMD) vs. Dong et al. [22].

Method	Dataset	PP (%)	PR (%)	OSR (%)	USR (%)
SMD based	average S3DIS	93.0	94.4	4.4	3.9
Dong et al.	average S3DIS	90.4	91.4	7.6	8.2

Table 5. Comparison of the proposed method (SMD) vs. Li et al. [10].

Method	Dataset	PP (%)	PR (%)	OSR (%)	USR (%)
SMD based	Room-1	86.5	98.5	9.7	3.7
Li et al.	Room-1	85.0	98.3	-	-

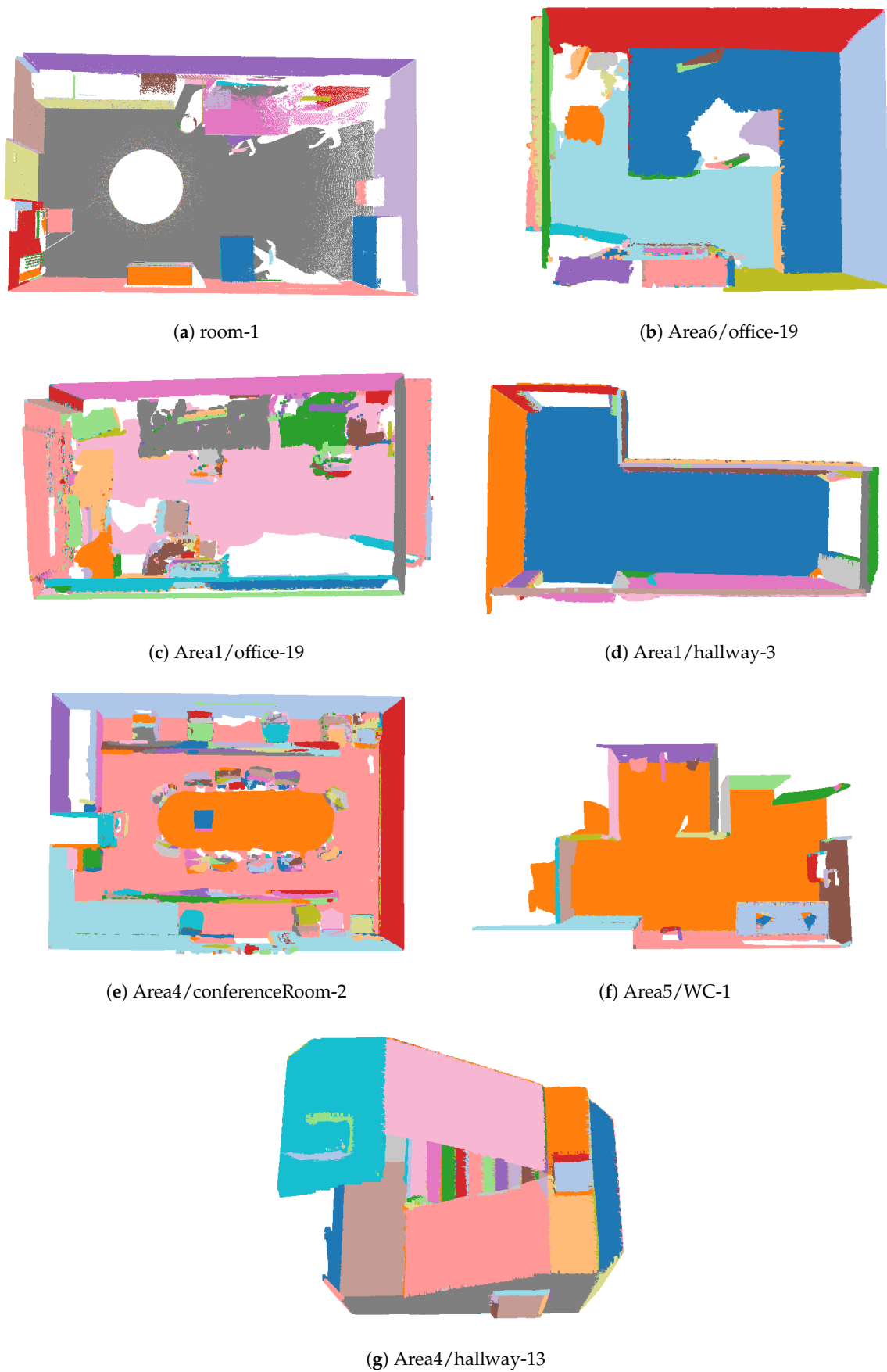


Figure 8. Preview of the results of the proposed method for seven exemplary validation sets.

The limitation of the proposed method concerns non-planar object detection that is decomposed into planar sets, like a trash bin (Figure 9), but such objects should be considered individually with other methods.

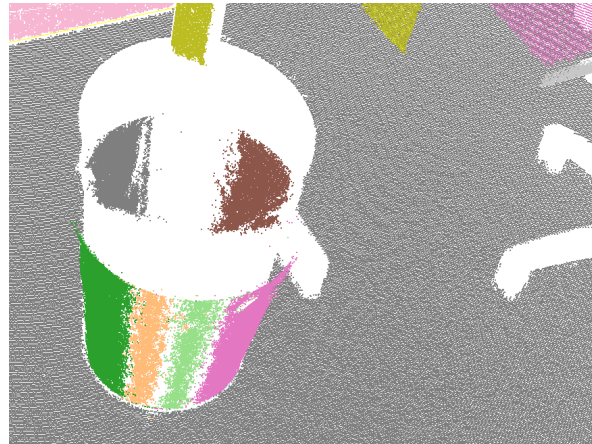


Figure 9. The curved bin over- segmented into the set of nearly planar fragments.

The presented results clearly confirm the contribution stated at the beginning of this paper. Robust plane refinement, based on SMD coupled with hd-kd-tree points cloud partition and its simple aggregation, yields better results than the current state-of-the-art methods. SMD method tends to converge to actual planar fragments, like FAST-MCD and MCS (Figure 10), yet within a much shorter time (Table 6). Partition with hd-kd-tree subdivides the space in such a way that fewer points are rejected compared to other methods (Figure 8) and the output is supplied in reasonable time virtually equal to the octree, which had the best time so far.

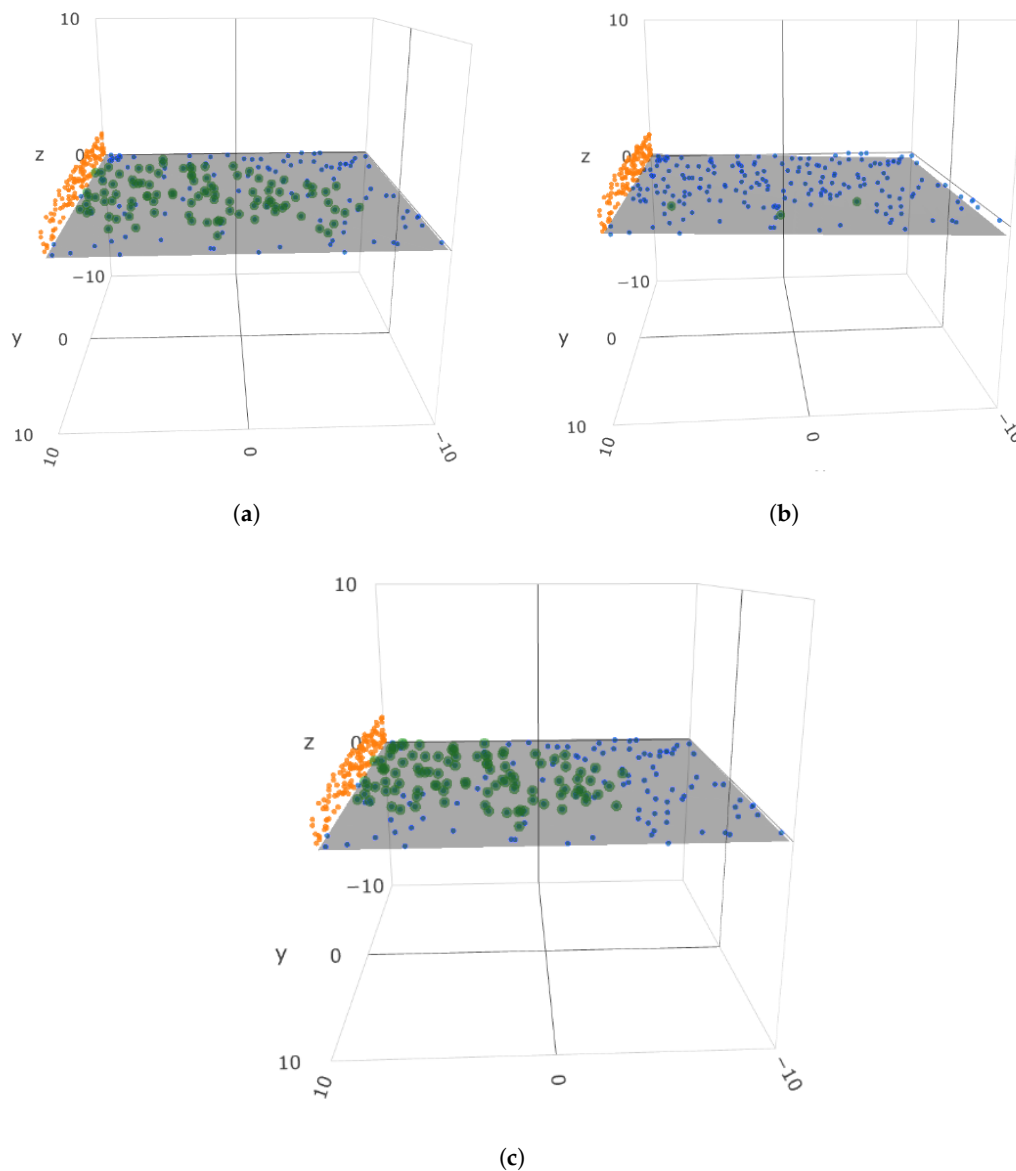


Figure 10. Comparison of (a) FAST-MCD, (b) MCS, and (c) SMD methods of core points determination (core points marked with green) for exemplary, roughly planar points cluster (blue points) containing a fragment of other planar group (orange points). Fitted plane is depicted in dark gray.

Table 6. Time comparison for FAST-MCD, MCS and the proposed SMD method for core points selection for points presented in Figure 10.

Method	FAST-MCD	MCS	SMD
Time (ms)	19.8 ± 1.0	4.1 ± 0.1	6.4 ± 0.45

6. Conclusions

The experiments described in this paper were focused on searching for an effective method of a point cloud subdivision process and efficient clustering of the set of planar points. As a result, a new efficient method exploiting histogram-driven kd-tree structure was introduced. It processes recurrent and orthogonal point cloud subdivision. A point cloud was divided efficiently into a set of planar point fragments. The experiments have clearly shown that the proposed partition method is superior over

the benchmark partition approaches presented in the literature, in terms of the number of resulting clusters and portion of the preserved points. It extracts a lower number of larger planar clusters than other state-of-the-art methods.

The conducted experiments have revealed that semi-deterministic strategy relying on the node core points selection, based on Shifted Mahalanobis Distance (SMD), enables precise, reliable, robust, and fast estimation of the plane parameters.

The proposed method was verified for sparse and noised datasets like these from the S3DIS database. The method correctly detects planes with the average plane precision (PP) score of 93.0% and the average plane recall (PR) at the level of 94.4%. It exceeds the examined reference methods by 2.6pp and 3.0pp respectively. Both over- and under- segmentation rates were relatively low and better than in the state-of-the-art methods (for the proposed method they fell to 4.4% and 3.9% respectively). The over- segmentation occurs mainly for curved objects that could not be approximated well by a single plane (like a trash bin) and should be treated separately.

Selective experiments to determine efficiency of the method were also conducted for the dense point cloud- Room-1. Plane precision reached 86.4% whereas plane recall increased to 98.5%.

In the case of a laser scan, the over-segmentation rate is higher than in the case of sparse datasets because of disturbances in point distribution in areas where the laser beam falls under very small or very large angle (ceiling above the laser head or floor just below the laser head). Li et al. did not report OSR nor USR rates.

Conducted surveys have demonstrated that application of the SMD-based plane fitting procedure allows for the selection of clusters of core points that are more representative than the other methods and the resulting planes better fit the considered clusters. Furthermore, application of a histogram-driven kd-tree partition yields a more balanced partition than other state-of-the-art partition procedures. This research also reveals a noticeable potential for further generalization, especially for non-planar, semantic model-based detection in point clouds segmentation tasks.

Author Contributions: Conceptualization, J.W., T.P. and A.W.; data curation, J.W.; methodology, J.W. and A.W.; software, J.W.; validation, J.W.; formal analysis, A.W.; investigation, J.W.; writing, J.W. and A.W.; visualisation, J.W. and T.P.; supervision, A.W.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hug, C. Extracting Artificial Surface Objects from Airborne Laser Scanner Data. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*; Gruen, A., Baltsavias, E.P., Henricsson, O., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; pp. 203–212, doi:10.1007/978-3-0348-8906-3_20. [[CrossRef](#)]
2. Kedzierski, M.; Fryskowska, A.; Wierzbicki, D.; Dabrowska, M.; Grochala, A. Impact of the method of registering Terrestrial Laser Scanning data on the quality of documenting cultural heritage structures. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, XL-5/W7, 245–248, doi:10.5194/isprsarchives-XL-5-W7-245-2015. [[CrossRef](#)]
3. Ramiya, A.M.; Nidamanuri, R.R.; Krishnan, R. Segmentation based building detection approach from LiDAR point cloud. *Egypt. J. Remote Sens. Space Sci.* **2017**, 20, 71–77. [[CrossRef](#)]
4. Forczmański, P.; Kutelski, K. Driver Drowsiness Estimation by Means of Face Depth Map Analysis. In *International Multi-Conference on Advanced Computer Systems*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 396–407.
5. Lipinski, P.; Lichy, K.; Santorek, J. Empirical research of autonomous robot control system. In Proceedings of the IEEE 13th CSIT 2018, Lviv, Ukraine, 11–14 September 2018; pp. 108–111.
6. Ziolkowski, P.; Szulwic, J.; Miskiewicz, M. Deformation Analysis of a Composite Bridge during Proof Loading Using Point Cloud Processing. *Sensors* **2018**, 18, 4332, doi:10.3390/s18124332. [[CrossRef](#)]

7. Chen, D.; Wang, R.; Peethambaran, J. Topologically Aware Building Rooftop Reconstruction From Airborne Laser Scanning Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7032–7052, doi:10.1109/TGRS.2017.2738439. [\[CrossRef\]](#)
8. Zhang, C.; He, Y.; Fraser, C.S. Spectral Clustering of Straight-Line Segments for Roof Plane Extraction From Airborne LiDAR Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 267–271, doi:10.1109/LGRS.2017.2785380. [\[CrossRef\]](#)
9. Vaskevicius, N.; Birk, A.; Pathak, K.; Schwertfeger, S. Efficient Representation in Three-Dimensional Environment Modeling for Planetary Robotic Exploration. *Adv. Robot.* **2010**, *24*, 1169–1197, doi:10.1163/016918610X501291. [\[CrossRef\]](#)
10. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. *Remote Sens.* **2017**, *9*, 433, doi:10.3390/rs9050433. [\[CrossRef\]](#)
11. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the Weighted RANSAC Approaches for Building Roof Plane Segmentation from LiDAR Point Clouds. *Remote Sens.* **2016**, *8*, 5, doi:10.3390/rs8010005. [\[CrossRef\]](#)
12. Ni, H.; Lin, X.; Ning, X.; Zhang, J. Edge detection and feature line tracing in 3d-point clouds by analyzing geometric properties of neighborhoods. *Remote Sens.* **2016**, *8*, 710. [\[CrossRef\]](#)
13. Eckart, B.; Kim, K.; Kautz, J. Fast and Accurate Point Cloud Registration using Trees of Gaussian Mixtures. *arXiv* **2018**, arXiv:1807.02587.
14. Kaiser, A.; Ybanez Zepeda, J.A.; Boubekeur, T. A survey of simple geometric primitives detection methods for captured 3d data. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2019; Volume 38, pp. 167–196.
15. Lazarek, J.; Pryczek, M. A Review on Point Cloud Semantic Segmentation Methods. *J. Appl. Comput. Sci.* **2018**, *26*, 99–105.
16. Xiao, J.; Zhang, J.; Adler, B.; Zhang, H.; Zhang, J. Three-dimensional Point Cloud Plane Segmentation in Both Structured and Unstructured Environments. *Robot. Auton. Syst.* **2013**, *61*, 1641–1652, doi:10.1016/j.robot.2013.07.001. [\[CrossRef\]](#)
17. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LIDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2798–2805, doi:10.1109/ICRA.2011.5979818. [\[CrossRef\]](#)
18. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100, doi:10.1016/j.isprsjprs.2015.01.011. [\[CrossRef\]](#)
19. Su, Y.T.; Bethel, J.; Hu, S. Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 59–74, doi:10.1016/j.isprsjprs.2016.01.001. [\[CrossRef\]](#)
20. Wang, M.; Tseng, Y.H. Automatic Segmentation of Lidar Data into Coplanar Point Clusters Using an Octree-Based Split-and-Merge Algorithm. *Photogramm. Eng. Remote Sens.* **2010**, *76*, 407–420, doi:10.14358/PERS.76.4.407. [\[CrossRef\]](#)
21. Meagher, D. Geometric modeling using octree encoding. *Comput. Graph. Image Process.* **1982**, *19*, 129–147, doi:10.1016/0146-664X(82)90104-6. [\[CrossRef\]](#)
22. Dong, Z.; Yang, B.; Hu, P.; Scherer, S. An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *137*, 112–133, doi:10.1016/j.isprsjprs.2018.01.013. [\[CrossRef\]](#)
23. Bentley, J.L. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* **1975**, *18*, 509–517, doi:10.1145/361002.361007. [\[CrossRef\]](#)
24. Granger, S.; Pennec, X. Multi-scale EM-ICP: A fast and robust approach for surface registration. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002; pp. 418–432.
25. Phillips, J.M.; Liu, R.; Tomasi, C. Outlier robust ICP for minimizing fractional RMSD. In Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007), Montreal, QC, Canada, 21–23 August 2007; pp. 427–434.

26. Stoyanov, T.; Magnusson, M.; Andreasson, H.; Lilienthal, A.J. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *Int. J. Robot. Res.* **2012**, *31*, 1377–1393. [\[CrossRef\]](#)
27. Liu, X.; Zhang, X.; Cheng, S.; Nguyen, T.B. A Novel Algorithm for Planar Extracting of 3D Point Clouds. In Proceedings of the International Conference on Internet Multimedia Computing and Service, Xi'an, China, 19–21 August 2016; ACM: New York, NY, USA, 2016; pp. 142–145, doi:10.1145/3007669.3007746. [\[CrossRef\]](#)
28. Hough, P.V.C. Method and Means for Recognizing Complex Patterns. U.S. Patent 3,069,654, 18 December 1962.
29. Ballard, D.H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–121, doi:10.1016/0031-3203(81)90009-1. [\[CrossRef\]](#)
30. Limberger, F.A.; Oliveira, M.M. Real-time detection of planar regions in unorganized point clouds. *Pattern Recognit.* **2015**, *48*, 2043–2053, doi:10.1016/j.patcog.2014.12.020. [\[CrossRef\]](#)
31. University of California, Merced. *Introduction to Computer Vision Fitting and Alignment*; University of California, Merced: Merced, CA, USA, 2015.
32. Qian, X.; Ye, C. NCC-RANSAC: A fast plane extraction method for navigating a smart cane for the visually impaired. In Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, USA, 17–21 August 2013; pp. 261–267, doi:10.1109/CoASE.2013.6653929. [\[CrossRef\]](#)
33. Nurunnabi, A.; West, G.; Belton, D. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognit.* **2015**, *48*, 1404–1419, doi:10.1016/j.patcog.2014.10.014. [\[CrossRef\]](#)
34. Charles, B. 4.5—Image Noise Models. In *Handbook of Image and Video Processing*, 2nd ed.; Bovik, A., Ed.; Communications, Networking and Multimedia, Academic Press: Burlington, NJ, USA, 2005; pp. 397–409, doi:10.1016/B978-012119792-6/50087-5. [\[CrossRef\]](#)
35. Hubert, M.; Rousseeuw, P.J.; Branden, K.V. ROBPCA: A New Approach to Robust Principal Component Analysis. *Technometrics* **2005**, *47*, 64–79, doi:10.1198/004017004000000563. [\[CrossRef\]](#)
36. Stahel, W. Robust Estimation: Infinitesimal Optimality and Covariance Matrix Estimators. Ph.D. Thesis, ETH, Zurich, Switzerland, 1981.
37. Donoho, D.L. *Breakdown Properties of Multivariate Location Estimators*; Technical Report; Harvard University: Boston, MA, USA, 1982.
38. Rousseeuw, P.J.; Driessen, K.V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **1999**, *41*, 212–223. [\[CrossRef\]](#)
39. Xu, Y.; Yao, W.; Hoegner, L.; Stilla, U. Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing. *Remote Sens. Lett.* **2017**, *8*, 1062–1071. [\[CrossRef\]](#)
40. Delong, A.; Osokin, A.; Isack, H.N.; Boykov, Y. Fast approximate energy minimization with label costs. *Int. J. Comput. Vis.* **2012**, *96*, 1–27. [\[CrossRef\]](#)
41. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
42. Markiewicz, J.S. The use of computer vision algorithms for automatic orientation of terrestrial laser scanning data. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B3*, 315–322, doi:10.5194/isprs-archives-XLI-B3-315-2016. [\[CrossRef\]](#)
43. Walczak, J.; Wojciechowski, A. Clustering Quality Measures for Point Cloud Segmentation Tasks In Proceedings of the International Conference on Computer Vision and Graphics, Warsaw, Poland, 14–16 September 2018; pp. 173–186, doi:10.1007/978-3-030-00692-1_16. [\[CrossRef\]](#)
44. Nelder, J.A.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308–313, doi:10.1093/comjnl/7.4.308. [\[CrossRef\]](#)
45. Nurunnabi, A.; Belton, D.; West, G. Robust Segmentation for Large Volumes of Laser Scanning Three-Dimensional Point Cloud Data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4790–4805, doi:10.1109/TGRS.2016.2551546. [\[CrossRef\]](#)
46. Blomley, R.; Weinmann, M.; Leitloff, J.; Jutzi, B. Shape distribution features for point cloud analysis — a geometric histogram approach on multiple scales. *ISPRS Ann.* **2014**, *II-3*, 9–16, doi:10.5194/isprsannals-II-3-9-2014. [\[CrossRef\]](#)

47. Leys, C.; Klein, O.; Dominicy, Y.; Ley, C. Detecting multivariate outliers: Use a robust variant of the Mahalanobis distance. *J. Exp. Soc. Psychol.* **2018**, *74*, 150–156, doi:10.1016/j.jesp.2017.09.011. [[CrossRef](#)]
48. Hubert, M.; Rousseeuw, P.J.; Van Aelst, S. High-breakdown robust multivariate methods. *Stat. Sci.* **2008**, *23*, 92–119. [[CrossRef](#)]
49. Weinmann, M.; Urban, S.; Hinz, S.; Jutzi, B.; Mallet, C. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Comput. Graph.* **2015**, *49*, 47–57. [[CrossRef](#)]
50. Bai, L.; Cheng, X.; Liang, J.; Shen, H.; Guo, Y. Fast density clustering strategies based on the k-means algorithm. *Pattern Recognit.* **2017**, *71*, 375–386, doi:10.1016/j.patcog.2017.06.023. [[CrossRef](#)]
51. Campello, R.J.G.B.; Moulavi, D.; Sander, J. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*; Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 160–172.
52. Landrieu, L.; Martin, S. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
53. Rooms UZH Irchel Dataset. Available online: <http://www.ifi.uzh.ch/en/vmml/research/datasets.html> (accessed on 6 March 2018).
54. Department of Computer Science & Engineering in the University of Washington. *Lecture 15: Principal Component Partition*; University of Washington College of Engineering: Seattle, WA, USA, 1999.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).