2021 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. June 15-18, 2021

Towards Robust and Efficient Plane Detection from 3D Point Cloud

Sina Sharif Mansouri[1], Christoforos Kanellakis[1], Farhad Pourkamali-Anaraki[2], and George Nikolakopoulos[1]

Abstract— **This article proposes a robust and scalable clustering method for 3D point-cloud plane segmentation with applications in Micro Aerial Vehicles (MAVs), such as Simultaneous Localization and Mapping (SLAM), collision avoidance, and object detection. Our approach builds on the sparse subspace clustering framework, which seeks a collection of subspaces that fit the data. Since subspace clustering requires solving a global sparse representation problem and forming a similarity graph, its high computational complexity is known to be a significant drawback, and performance is sensitive to a few hyperparameters. To tackle these challenges, our method has two key ingredients. We use randomized sampling to accelerate subspace clustering by solving a reduced optimization problem. We also analyze the obtained segmentation for quality assurance and performing a post-processing process to resolve two forms of model mismatch. We present numerical experiments to demonstrate the benefits and merits of our method.**

Index Terms— **Subspace clustering, MAVs, robotics, plane segmentation, computational efficiency**

I. Introduction

Recent advances in sensing devices and the increase in computational power have led to the extensive use of depth cameras and 3D sensors in various robotics applications, such as Simultaneous Localization and Mapping (SLAM) [1], [2], urban areas and infrastructure 3D reconstructions with Micro Aerial Vehicles (MAVs) [3], [4], 3D object detection [5], path planning, and collision avoidance for the MAVs [6], [7], [8]. However, the obtained 3D point clouds from sensing devices are usually large-scale, noisy, redundant, and do not provide sufficient scene semantics. Thus, it is necessary to reduce the data size and extract meaningful information.

Detecting planar segments in a point cloud has played an essential role in providing useful representations in autonomous MAVs. In indoor environments, for instance, ceilings, floors, walls, and surfaces are planar. Also, in outdoor scenes, the ground is typically piecewise planar. The horizontal planes are used as a support for other objects (such as pick and place of objects with manipulators) or traversable terrain for robots. Vertical planes are usually considered as obstacles or walls, which are critical for autonomous navigation of aerial robots. The obtained information regarding planes can be also directly used in SLAM. However, an inaccurate plane segmentation will have dramatic consequences on the quality of a MAVs mission. Examples include drift in localization, collision of the platform, imprecise path planning, or dropping objects in incorrect surface areas.

This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 869379 illuMINEation.

[1]The authors are with the Robotics Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden.

[2]The author is with Department of Computer Science, University of Massachusetts Lowell, MA, USA.

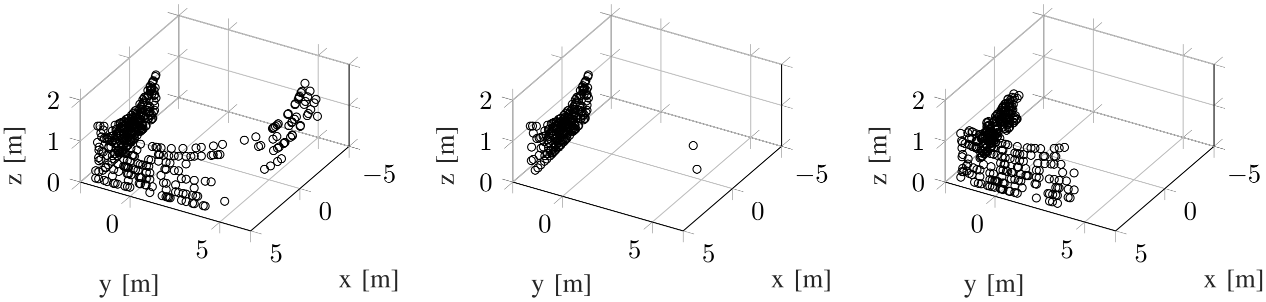
While the task of plane detection has received significant attention over the last years, most research relies on organized point clouds, such as RGB-D images [9], where the neighbor information can be used. However, extracting planes from unorganized point clouds is more challenging because of the cloud size variations, which means that the neighbor information cannot be immediately used [10].

In this paper, we propose a novel method for the task of plane segmentation in unorganized point clouds based on subspace clustering, a generalization of Principal Component Analysis (PCA). Subspace clustering assumes the Union of Subspaces (UoS) model for analyzing a set of unlabeled data points, where the goal is to recover subspaces by assigning each point to its corresponding subspace. Among various techniques [11], the state-of-the-art subspace clustering uses the self-expressiveness property [12]. That is, each data point in a UoS model can be expressed as a linear combination of other points from its own subspace.

After solving a sparse representation problem for all data points, one forms a similarity graph and partitions the original data points using the normalized cut algorithm [13]. While the high computational complexity of this technique is known to be a significant challenge, we bring the focus into another critical issue in this article. Unfortunately, false connections in similarity graphs may lead to imprecise clustering of a small portion of the data. Existing clustering quality metrics are not useful in our problem of interest because the erroneous assignment of even a few points does not allow us to characterize worst-case results regarding the estimated subspaces.

A. Background and Motivation

In the literature, plane segmentation has been extensively investigated, and we present a brief overview. These methods can be divided into three main categories. (a) Point clustering, based on similarities between the measurements such as distance and angle between surface normal [14]. In [15], the points are clustered into super voxels, and based on an adjacency graph the regions are clustered. The authors in [16] proposed to cluster data points with similar normals. (b) Region growing, where the method chooses seed points or regions, and cluster the points based on that information. In [17], an initial set of candidates are produced, then extracted a regular arrangement of planes with their relations. The authors in [18] proposed fast segmentation of 3D point clouds for autonomous vehicles. This method extracts a set of seed points based on low height values



(a) 3D point cloud (b) segmentation (a few outliers) (c) segmentation (connected planes)

Fig. 1: Illustrating two types of clustering imperfections when applying subspace clustering to a 3D point cloud in (a). We see a small set of points lying outside the plane in (b) and the two connected planes are identified as one cluster in (c). Existing subspace clustering methods are not able to identify these erroneous cases without the knowledge of ground-truth labels.

to estimate the ground surface and then extract the points close to initial ground plane. (c) Random sample consensus (RANSAC)-based plane fitting where points from the point cloud are sampled, and planar models are fitted to them. In [19], a RANSAC method was proposed for performing normal coherence check on points and removed the data points whose normal directions were contradictory to the fitted plane.

While most research has focused on designing scalable subspace clustering methods, a critical question is left unanswered regarding the accuracy of representing each data point in a UoS model. To be formal, ensuring that each data point selects others points from its own subspace requires tuning a regularization parameter, and also depends on other factors such as the number of samples in each subspace (e.g., see [20]). Hence, false connections in the similarity graph constructed from such sparse representations may lead to imprecise assignments, which is problematic for plane segmentation from 3D point clouds. For example, in Figure 1, we exhibit two forms of clustering imperfections that we observed by running subspace clustering multiple times on a 3D point cloud. We will present a post-processing mechanism to determine whether each obtained segmentation is valid. When we find out a segmentation does not reflect the structure of a plane, we propose to remove outliers and partition connected planes to improve the performance of subspace clustering.

B. Contributions

To improve both scalability and robustness of subspace clustering, our main contributions are twofold. First, we show how randomized sampling reduces the computational cost of subspace clustering. Second, to the best of our knowledge, our method is the first line of work in the context of subspace clustering that aims to measure the quality of clustering without the knowledge of ground-truth labels, which are typically assumed to be a priori known in the previous research. Moreover, our method takes the first step towards designing post-processing mechanisms to improve the quality of assignments by removing outliers and partitioning connected subspaces. The proposed method is valuable in robotics since it can be used in closed-loop for controlling robot actions, as ignoring inaccurate plane segmentation is a better policy than robot actions, which may fail a mission with dramatic consequences.

C. Outline

The rest of the article is structured as follows. Section II presents our proposed methodology for identifying the number of planes and evaluation metrics. Then, in Section III, we evaluate the proposed method on various simulated and real-world data sets. Finally, Section IV concludes this article by summarizing our findings and offering some future research directions to improve our framework further.

II. Segmentation of 3D Point Cloud

This section explains the proposed subspace clustering method along with our new post-processing strategy. Suppose that the obtained 3D point cloud consists of n points, i.e., $\mathcal{P} = \{(x_i, y_i, z_i)\}_{i=1}^n$, then we will form a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{3\times n}$ by representing each data point as a column vector. As mentioned before, these points are drawn from the UoS model in $\mathbb{R}^3$, so we can express each point as a sparse linear combination of all the other data points.

Subspace clustering techniques aim to partition the data into multiple clusters and fit each partition or cluster with a low-dimensional subspace, such as a 2D surface in our problem of interest. An effective approach [11] to subspace clustering relies on solving the following optimization problem:

$$\min_{\mathbf{c}_j \in \mathbb{R}^n} \|\mathbf{c}_j\|_1 + \frac{\lambda}{2}\|\mathbf{x}_j - \sum_{i\neq j} c_{ij}\mathbf{x}_i\|_2^2 \ \text{ s.t. } \ \mathbf{c}_j^T\mathbf{1}_n = 1, \quad (1)$$

where $\|\cdot\|_q$ represents the ℓ_q norm for vectors, λ is a regularization parameter, $\mathbf{c}_j = [c_{1j}, \ldots, c_{nj}]^T$ is the coefficient vector corresponding to $\mathbf{x}_j$, and $\mathbf{1}_n$ is the vector of all ones. The constraint $i \neq j$ avoids the trivial solution of expressing $\mathbf{x}_j$ via itself, and the constraint $\mathbf{c}_j^T\mathbf{1}_n = 1$ allows us to extract the general case of affine subspaces. This is crucial because subspaces do not necessarily pass through the origin.

The above optimization problem can be cast in a more concise form for the entire data set, i.e., $\mathbf{x}_j,\ j = 1, \ldots, n$:

$$\min_{\mathbf{C}\in\mathbb{R}^{n\times n}} \|\mathbf{C}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 \tag{2a}$$

$$\text{s.t. } \operatorname{diag}(\mathbf{C}) = \mathbf{0}_n,\ \ \mathbf{C}^T\mathbf{1}_n = \mathbf{1}_n, \tag{2b}$$

where $\|\mathbf{C}\|_1 = \sum_{i,j}|c_{ij}|$, $\|\mathbf{C}\|_F^2 = \sum_{i,j} c_{ij}^2$, and $\operatorname{diag}(\mathbf{C})$ is the vector of the diagonal elements of $\mathbf{C}$. After solving the above optimization problem and finding the coefficient matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_n]$, the next step is to find the segmentation of the data into multiple subspaces. Towards this goal, subspace clustering methods build a weighted graph with n nodes corresponding to the n original points, and its similarity matrix is defined as $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$. That final step involves applying the normalized cut algorithm [13] to $\mathbf{W}$ for partitioning.

To be formal, we form the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix. Then, the K eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K \in \mathbb{R}^n$ corresponding to the K smallest eigenvalues of $\mathbf{L}$ are found (K represents the number of clusters). The last step of spectral clustering is to perform *K-means* clustering [21] on the rows of the matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K] \in \mathbb{R}^{n\times K}$ to cluster the original data (one can normalize the rows before clustering). The optimization problem in (2) can be solved using the Alternating Direction Method of Multipliers (ADMM) [22], which scales poorly with the data size. The computational cost of existing implementations is cubic or quadratic in terms of the number of data points [23]. Moreover, one has to form various n-dimensional square matrices (e.g., $\mathbf{C}$ and $\mathbf{W}$) that will lead to high memory usage. Hence, existing subspace clustering methods are not appropriate for our problem of interest due to time constraints. We address these problems by introducing a scalable subspace clustering technique.

The high computational cost of constructing the coefficient matrix originates from computing a regularized representation of every single data point with respect to the whole dataset. Thus, the first step of our approach involves forming two subsets of the original data by uniform sampling without replacement. Although we can use more sophisticated non-uniform sampling techniques, such as [20], we decide to use uniform sampling because our post-processing step will allow us to improve the clustering performance if required. We also eliminate the constraint $\mathbf{C}^T\mathbf{1}_n = \mathbf{1}_n$ by mapping the original data from $\mathbb{R}^3$ to $\mathbb{R}^4$. This trick is known as homogeneous embedding [24], where we add a new coordinate which is 1 (or another constant) to each point, i.e., $\mathcal{P} = \{(x_i, y_i, z_i, 1)\}_{i=1}^n$.

Given two sampling parameters $0 < \kappa_1 < \kappa_2 < 0.5$, we create two sets of indices $\mathcal{I}_1$ and $\mathcal{I}_2$ with $n_1 = \lfloor \kappa_1 n \rfloor$ and $n_2 = \lfloor \kappa_2 n \rfloor$ elements from $\{1, \ldots, n\}$ selected uniformly at random, respectively. Then, we seek to solve the following sparse representation problem for each $\mathbf{x}_j,\ j \in \mathcal{I}_2$:

$$\min_{\mathbf{c}_j\in\mathbb{R}^{n_1}} \|\mathbf{c}_j\|_1 + \frac{\lambda}{2}\|\mathbf{x}_j - \sum_{i\in\mathcal{I}_1} c_{ij}\mathbf{x}_i\|_2^2. \tag{3}$$

The above optimization problem does not return a trivial solution because $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$, and it can be solved efficiently using the SPArse Modeling Software (SPAMS) package [25].

After solving the new optimization problem, we should apply the normalized cut algorithm to the obtained coefficient matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_{n_2}]$. However, the matrix $\mathbf{C}$ is not square anymore because $n_1 \neq n_2$, and we often want n_1 to be much smaller than n_2 to reduce the computational cost.

To tackle this problem, we *implicitly* form the similarity matrix $\mathbf{W} = \widetilde{\mathbf{C}}^T\widetilde{\mathbf{C}} \in \mathbb{R}^{n_2\times n_2}$, where $\widetilde{\mathbf{C}} = |\mathbf{C}|$. Next, we present an efficient approach to perform normalized cut using the new similarity matrix. First, note that the degree matrix can be computed without forming any $n \times n$ matrix. In particular, we find the i-th element of $\mathbf{D}$ as follows:

$$\sum_{j=1}^{n_2} w_{ij} = \sum_{j=1}^{n_2} \widetilde{\mathbf{c}}_i^T\widetilde{\mathbf{c}}_j = \widetilde{\mathbf{c}}_i^T\boldsymbol{\eta} = \widetilde{\mathbf{c}}_i \cdot \boldsymbol{\eta}, \tag{4}$$

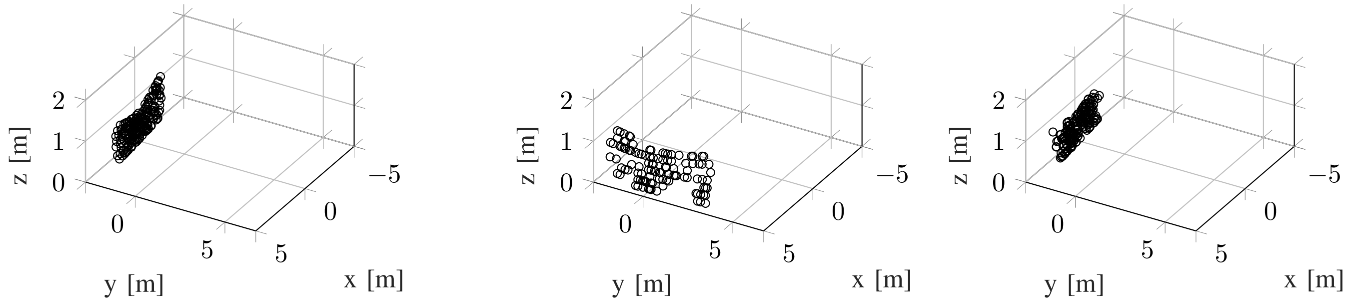
where $\boldsymbol{\eta} = \sum_{j=1}^{n_2}\widetilde{\mathbf{c}}_j \in \mathbb{R}^{n_1}$. Thus, we compute the diagonal degree matrix $\mathbf{D}$ using n_2 scalar products.

The remaining task is to compute the K smallest eigenvectors of the graph Laplacian matrix. We can reduce the computational cost and memory usage of this step by computing the top K eigenvectors of $\mathbf{I}_n - \mathbf{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. Let $\mathbf{U}\boldsymbol{\Sigma}\mathbf{P}^T$ be the singular value decomposition of $\widetilde{\mathbf{C}}\mathbf{D}^{-1/2} \in \mathbb{R}^{n_1\times n_2}$, where $\mathbf{U} \in \mathbb{R}^{n_1\times r}$ (left singular vectors), $\mathbf{P} \in \mathbb{R}^{n_2\times r}$ (right singular vectors), $\boldsymbol{\Sigma}$ contains the singular values, and r is the rank parameter. Then, the top K eigenvectors of $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ are equivalent to the top K right singular vectors of $\widetilde{\mathbf{C}}\mathbf{D}^{-1/2}$ since we have [26]:

$$\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{P}^T)^T(\mathbf{U}\boldsymbol{\Sigma}\mathbf{P}^T) = \mathbf{P}\boldsymbol{\Sigma}^2\mathbf{P}^T. \tag{5}$$

After computing the top eigenvectors and performing *K-means* clustering, we obtain the segmentation of the data points indexed by $\mathcal{I}_2$. Note that a significant advantage of using the normalized cut algorithm for clustering is that we can infer the number of clusters K. We use the multiplicity of singular value 1 in (5) to estimate the number of connected components because we are using the spectral decomposition of $\mathbf{I}_n - \mathbf{L}$, instead of the Laplacian matrix $\mathbf{L}$.

The majority of existing works use the knowledge of ground-truth labels to evaluate the performance of subspace clustering, such as clustering accuracy or normalized mutual information [27]. The evaluation step is critical because there are various parameters that have to be tuned, including the regularization parameter λ, and sampling parameters κ_1 and κ_2. However, in many applications such as autonomous systems, the information concerning labels is not readily available. Thus, we propose a simple technique to determine whether each cluster is valid. Towards this goal, we perform PCA on each cluster by subtracting the mean and computing the singular value decomposition of the centered data. As we are looking for 2D surfaces, we define a threshold γ to



(a) removing few outliers

(b) partitioning a segmentation into two surfaces

Fig. 2: Demonstrating the performance of our post-processing step on resolving two types of clustering imperfection shown in Figure 1. In (a), we remove those points lying outside the 2D surface in sub-figure 1(b). Also, in (b), we partition the obtained segmentation, shown in sub-figure 1(c), into two surfaces.

understand the distribution of points in each cluster. To be formal, when $\sigma_1/\sigma_3 > \gamma$, where σ_1 and σ_3 are the largest and smallest singular values in each cluster, then we accept the cluster as a 2D surface ($\gamma = 10$ in this work).

The next task is to devise a strategy to resolve two main forms of clustering imperfections (illustrated in Figure 1), so we do not have to repeat subspace clustering from scratch. In the first case, consider a cluster that consists of a 2D surface and a few data points lying outside this surface. We propose to use the Local Outlier Factor (LOF) algorithm [28] for filtering the segmentation by removing those points lying outside the surface. LOF is an unsupervised learning method that identifies anomalous data points by measuring the local deviation of a given data point with respect to its neighbors.

Another form of clustering imperfection relates to clusters that consist of two or more connected 2D surfaces as one of the main challenges in subspace clustering is to separate points near the intersections. We propose to use the normalized cut algorithm with a similarity function obtained using the Gaussian kernel function of the form $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\beta\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$, where $\beta > 0$ is the bandwidth parameter. This step enables partitioning each cluster for the second time if needed. Using the same strategy we discussed before, we can perform PCA to validate each sub-cluster, i.e., we measure the rank of centered points in each sub-cluster.

A potential drawback of the above approach is that we may segment a 2D surface into two parts. To avoid this issue, we use a measure of correlation between subspaces, known as subspace affinity. To be formal, this measure is defined as $\text{aff}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{\sqrt{2}}\|\mathbf{H}_1^T\mathbf{H}_2\|_F$, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are the two obtained surfaces in our application (we consider 2D surfaces in this work). Also, $\mathbf{H}_1, \mathbf{H}_2 \in \mathbb{R}^{3\times 2}$ are corresponding orthonormal bases. The subspace affinity is between 0 and 1; $\text{aff}(\mathcal{S}_1, \mathcal{S}_2) = 0$ indicates the subspaces are orthogonal and $\text{aff}(\mathcal{S}_1, \mathcal{S}_2) = 1$ means that the two subspaces are identical. Hence, we can use this measure to ensure that our method does not partition a 2D surface into two components.

Finally, we exemplify how our post-processing step can resolve some of the issues associated with subspace clustering. In Figure 2, we consider the same setup as in Figure 1. The overall approach is summarized in Figure 3.

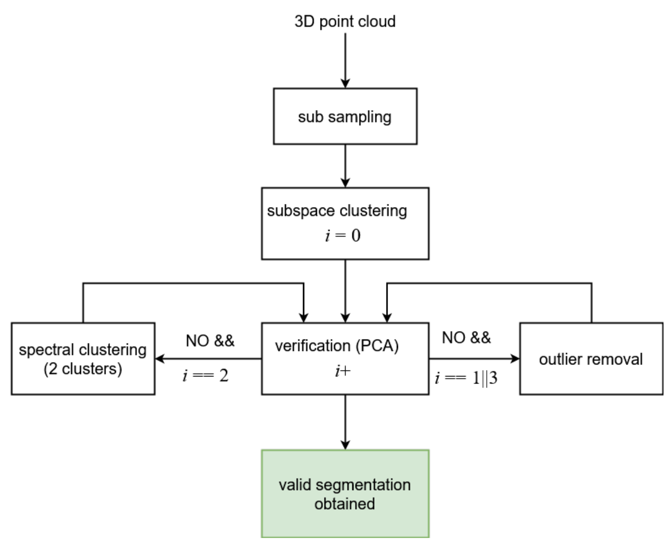


Fig. 3: Block diagram of the overall proposed scheme.

III. Results

To evaluate the performance of the proposed method, initially, the Gazebo [29] robot simulator is used. For this purpose, the quad-copter is equipped with the 3D lidar Velodyne VLP-16 and the method is evaluated with different environments, while the measurement of 3D lidar is used. Furthermore, the data-set collected from the MAV navigation equipped with 3D lidar in Sweden underground tunnel is used for the experimental evaluation of the proposed method. The proposed method has been developed in Python within the ROS[1] framework.

A. Plane segmentation

Figure 4 depicts the results of plane segmentation in different scenarios exhibiting complex environments. The quad-copter is located in the center of the point cloud.

[1] https://www.ros.org/

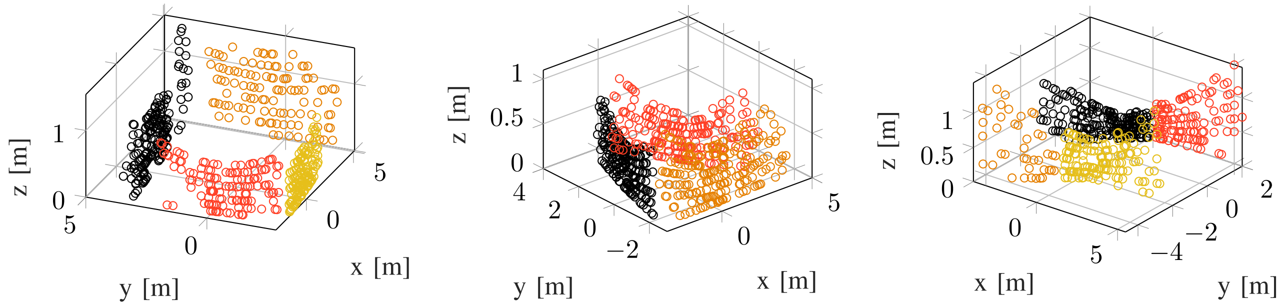


Fig. 4: The extracted surfaces from various 3D point clouds in different scenarios based on the proposed segmentation method, each cluster is indicated with a different color. Best viewed in color.

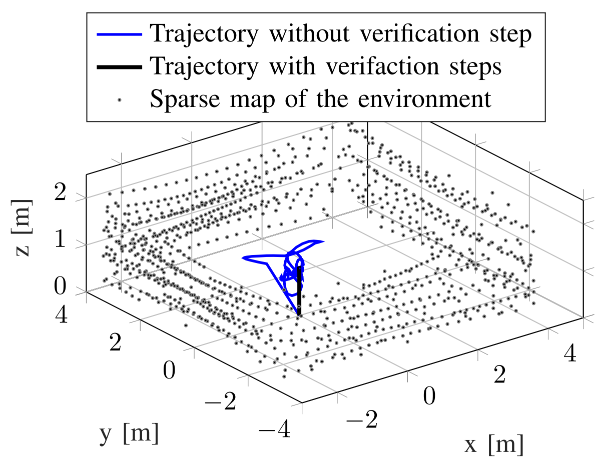


Fig. 5: The trajectory of the MAV during navigation in the confined environment, while the way-points sets to $[0, 0, 1]^\top$ for x, y, and z axes respectively.

The environments are inspired by human-made structures such as closed areas, corridors, and junctions. The mean and maximum of running times for our plane segmentation method are $0.01\,\text{sec}$ and $0.2\,\text{sec}$, respectively (standard subspace clustering takes at least a few seconds).

B. Navigation in confined environment

This scenario considers the integration of the proposed surface extraction method for the case of MAV navigation in confined environments. More specifically, a Nonlinear Model Predictive Control (NMPC) scheme is employed to generate the proper commands for the MAV system, while incorporating plane constraints for obstacle avoidance purposes. Thus, in the performed simulations, the proposed method is related to the avoidance task and provides the extracted plane coefficients to the control scheme in closed loop flights. In this scenario, a confined environment with no entry/exit is chosen to evaluate the performance of the method with and without the verification steps.The MAV takes off from the ground and follows the way-point reference located at $[0, 0, 1]^\top$ for x, y, and z axes correspondingly. The extracted planes are used in the controller of the MAV for collision avoidance, while we define safety distance of $1[m]$ from all extracted planes. The focus of these simulations is to highlight the necessity of the verification steps and the effect on the performance of the MAV navigation.

Two scenarios are considered one with and one without verification steps, while in both cases the collision avoidance constraints are active for the controller. Figure 5 depicts the trajectory of the MAV in each scenario. This Figure shows that the MAV for the majority of the simulation run hovers close to the desired location when the verification steps are enabled, compared to the other case where it oscillates more.

C. 3D lidar data-set from underground tunnels

In this case, the MAV navigates in underground tunnel and void environments located in Sweden [30], [31], while equipped with a Puck LITE Velodyne Lidar. The collected data-sets from the 3D lidar are post processed offline using the proposed method to extract surfaces, where different areas with different structures are used for the evaluation. Figure 6 depicts the resulting extracted surfaces from the 3D point clouds using the proposed method, while Figure 7 shows the images from the looking forward camera in the environment.

IV. Conclusion

In this paper, we present a robust subspace clustering for identifying planar segments. Notably, our approach can automatically examine the quality of recovered planes and take a few steps to improve the quality of obtained segmentation. In the future, we will work on improving our post-processing step to resolve other variants of clustering imperfections, which will increase the generality and reliability of our approach.

References

[1] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[2] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[3] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.

[4] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

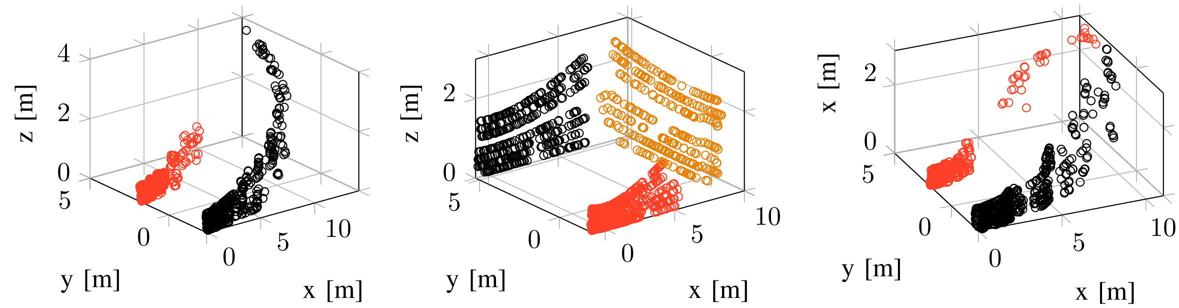


Fig. 6: The extracted surfaces from various 3D point clouds in 3D lidar data-sets extracted from MAV navigation in Sweden underground tunnels. Each cluster is indicated with a different color. Best viewed in color.

Fig. 7: The images from the looking forward camera of the MAV navigation in three different environments of an underground tunnels in Sweden.

[5] X. Qian and C. Ye, "3d object recognition by geometric context and gaussian-mixture-model-based plane classification," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3910–3915.

[6] R. Carelli and E. O. Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 235–247, 2003.

[7] S. S. Mansouri, C. Kanellakis, B. Lindqvist, F. Pourkamali-Anaraki, A.-A. Agha-Mohammadi, J. Burdick, and G. Nikolakopoulos, "A unified nmpc scheme for mavs navigation with 3d collision avoidance under position uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5740–5747, 2020.

[8] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," *IFAC-PapersOnLine*, 2020.

[9] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6218–6225.

[10] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, p. 433, 2017.

[11] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[12] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3918–3927.

[13] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[14] B. Sun and P. Mordohai, "Oriented point sampling for plane detection in unorganized point clouds," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2917–2923.

[15] T. T. Pham, M. Eich, I. Reid, and G. Wyeth, "Geometrically consistent plane extraction for dense indoor 3d maps segmentation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4199–4204.

[16] D. J. Lee and M. E. Campbell, "An efficient probabilistic surface normal estimator," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2248–2254.

[17] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra, "Rapter: rebuilding man-made scenes with regular arrangements of planes." *ACM Trans. Graph.*, vol. 34, no. 4, pp. 103–1, 2015.

[18] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5067–5073.

[19] X. Qian and C. Ye, "Ncc-ransac: a fast plane extraction method for 3-d range data segmentation," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2771–2783, 2014.

[20] C. You, C. Li, D. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *European Conference on Computer Vision*, 2018, pp. 67–83.

[21] F. Pourkamali-Anaraki and S. Becker, "Preconditioned data sparsification for big data with applications to PCA and K-means," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954–2974, 2017.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] F. Pourkamali-Anaraki, J. Folberth, and S. Becker, "Efficient solvers for sparse subspace clustering," *Signal Processing*, vol. 172, p. 107548, 2020.

[24] C. Li, C. You, and R. Vidal, "On geometric analysis of affine sparse subspace clustering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1520–1533, 2018.

[25] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[26] F. Pourkamali-Anaraki, "Large-scale sparse subspace clustering using landmarks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

[27] F. Pourkamali-Anaraki and S. Becker, "Improved fixed-rank Nyström approximation via QR decomposition: Practical and theoretical aspects," *Neurocomputing*, vol. 363, pp. 261–272, 2019.

[28] M. Breunig, H. Kriegel, R. Ng, and S., "LOF: identifying density-based local outliers," in *ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104.

[29] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[30] S. S. Mansouri, C. Kanellakis, G. Georgoulas, and G. Nikolakopoulos, "Towards MAV navigation in underground mine using deep learning," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 880–885.

[31] S. S. Mansouri, M. Castaño, C. Kanellakis, and G. Nikolakopoulos, "Autonomous MAV navigation in underground mines using darkness contours detection," in *International Conference on Computer Vision Systems*. Springer, 2019, pp. 164–174.