A Novel Topology Metric for Indoor Point Cloud SLAM Based on Plane Detection Optimization

Zhenchao Ouyang[1,2(✉)], Jiahe Cui[1,3], Yunxiang He[4], Dongyu Li[1,2], Qinglei Hu[1,2], and Changjie Zhang[5]

[1] Zhongfa Aviation Institute, Beihang University, 166 Shuanghongqiao Street, Pingyao Town, Yuhang District, Hangzhou 311115, China
ouyangkid@buaa.edu.cn
[2] Tianmushan Laboratory, Hangzhou 310023, Zhejiang, China
[3] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[4] Zhejiang Leapmotor Technology CO., LTD., Hangzhou 31000, Zhejiang, China
[5] Wisedawn Auto Co., LTD., South Henglong Road, Jingzhou 434000, Hubei, China

Abstract. Accurate self-localization and navigation in complex indoor environments are essential functions for the intelligent robots. However, the existing SLAM algorithms rely heavily on differential GPS or additional measuring devices (such as expensive laser tracker), which not only increase research costs but also limit the deployment of algorithms in specific scenarios. In recent years, reference-free pose estimation methods based on the topological structure of point cloud maps have gained popularity, especially in indoor artificial scenes where rich planar information is available. Some existing algorithms suffer from inaccuracies in spatial point cloud plane segmentation and normal estimation, leading to the introduction of evaluation errors. This paper introduces the optimization of plane segmentation results by incorporating deep learning-based point cloud semantic segmentation and proposes measurement indicators based on the Plane Normals Entropy (PNE) and Co-Plane Variance (CPV) to estimate the rotation and translation components of SLAM poses. Furthermore, we introduce a ternary correlation measure to analyze the relationship between noise, relative pose estimation, and the two proposed measures, building upon the conventional binary correlation measure. Our proposed PNE and CPV metrics were quantitatively evaluated on two different scenarios of LiDAR point cloud data in Gazebo simulator, and the results demonstrate that these metrics exhibit superior binary and triple correlation and computational efficiency, making them a promising solution for accurate self-localization and navigation in complex indoor environments.

Keywords: Point Cloud · SLAM · Topology Entropy · Plane detection · Segmentation

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 563, pp. 23–40, 2024.
https://doi.org/10.1007/978-3-031-54531-3_2

1 Introduction

With the development of artificial intelligence technology, sensors, and computing hardware, intelligent mobile robots have the potential to assist or replace humans in performing repetitive and simple daily tasks, freeing people from heavy labor and providing significant commercial and social benefits. Environment understanding and autonomous localization are fundamental capabilities for mobile robots, and mainstream solutions often employ GPS [1], Ultra-Wideband (UWB) [2], or Simultaneous Localization And Mapping (SLAM) [3–5] techniques. However, GPS signals can be obstructed indoors, and the UWB approach requires modifications to the environment and additional costs. When a robot moves in a complex environment, it needs to have a global map and its current pose like a human. Therefore, to improve robot flexibility and autonomy, existing solutions often utilize environmental sensors (such as stereo cameras or LiDAR) mounted on the robot with SLAM-based solutions [3,6–8].

For the past two decades, the evaluation of SLAM has heavily relied on simulation data or expensive equipment such as laser trackers, motion capture devices, or total stations [9], due to its dependence on relative pose and absolute pose errors (RPE and APE) [10]. This has greatly hindered the development of SLAM, as the relative pose estimation and absolute pose estimation directly calculate the difference between the estimated robot coordinate and the true displacement, and require temporal and spatial synchronization of the data. Even with expensive measurement equipment, constructing corresponding SLAM datasets [11,12] for large-scale, non-line-of-sight covered environments remains challenging.

In contrast to complex natural environments, artificial indoor environments generally have relatively stable and regular topological structures. These features have been utilized to develop SLAM algorithms, and some studies [13–15] have attempted to indirectly evaluate the accuracy of pose estimation for feature maps constructed by SLAM based on topological analysis. In addition, as the cost of LiDAR continues to decrease, mobile robots can efficiently scan the environment structure, ensuring rich local map features. If there is an error in SLAM pose estimation, the local feature map overlaid based on the pose estimation will also become distorted and offset. By analyzing the consistency of the map features through topological measurements, it is possible to infer the pose estimation error [16].

The existing topological analysis-based optimization can be divided into the following two categories. 1) the graph optimization based on loop closure detection. This kind of method requires the robot to have the ability to discover revisited places while performing continuous pose estimation, and then construct a loop closure based on the features of revisited places (such as two frames of scans or local feature maps), followed by graph optimization. Loop closure can be detected based on heuristic methods [17], or through human assistance [18,19], and some solutions utilize neural network models [20] for detection. However, if the robot's trajectory cannot form a loop closure, graph optimization correction cannot be performed. 2) Estimation errors based on topological information

from feature maps. The Mean Map Entropy (MME) and Mean Plane Variance (MPV) [13,14] estimate the consistency of the map by respectively estimating the entropy and planar variance of local point clusters in the map. However, both MME and MPV require traversing all the feature points of the map, resulting in low efficiency and an inability to provide specific error locations. Mutually Orthogonal Metric (MOM) [15] prunes the search space by selecting mutually orthogonal planar features, which greatly improves computational efficiency and makes the features more stable on mutually orthogonal planes. However, the deployment of MOM in real environments is affected by large errors in the plane detection algorithm as well as the introduction of normal estimation errors.

In order to provide a measure of pose estimation based on local topological features and overcome the problem of large errors in plane segmentation and numerous normal estimation errors in SLAM, this paper uses a neural network-based semantic segmentation method to improve the robustness and efficiency of plane segmentation. Furthermore, this paper analyzes the effects of rotation and translation components on different types of topological measurements in pose estimation and proposes a more comprehensive topological estimation method. Finally, we conduct a quantitative evaluation of our proposed method in an isometric simulation environment.

2 Basic Conception

2.1 Euclidean Transformation and RPE

The motion of the robot can be regarded as a coordinate transformation problem of a rigid body in Euclidean space, which consists of three mutually orthogonal axes. Rigid body motion can usually be split into two parts, rotation (R) and translation (t). The collection of three-dimensional rotation matrices based on a three-dimensional orthogonal basis is typically defined as Eq. 1, where $SO(3)$ is a special orthogonal group and I is the identity matrix, which is an orthogonal matrix with a determinant ($det(R)$) of 1.

$$SO(3) = \{R \in \mathbb{R}^{3*3} | RR^T = I, det(R) = 1\} \tag{1}$$

The Euclidean transformation of a robot from α to α' can be defined as Eq. 2, where the concepts of homogeneous coordinates and transformation matrices are introduced. The transformation matrix T forms a special Euclidean group, which is defined as Eq. 3.

$$\begin{bmatrix} \alpha' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ 1 \end{bmatrix} = T \begin{bmatrix} \alpha \\ 1 \end{bmatrix} \tag{2}$$

$$SE(3) = \left\{ T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4*4} | R \in SO(3), t \in \mathbb{R}^3 \right\} \tag{3}$$

Usually, in order to simplify expression and calculation, the R component is expressed by the Rodrigues' Formula, which rotates around the unit vector by

the rotation angle θ. The three-dimensional rotation R in real world is generally decomposed into the $[roll, pitch, yaw]$ rotation around the X-Y-Z axis with a fix order. The translation vector t is represented as $[x, y, z]^T$, and translation components T is called a pose of the robot.

$$\begin{aligned} RPE_{i,j} &= ||E_{i,j} - I_{4*4}|| \\ E_{i,j} &= \Delta T_{i,j}^{gt} (\Delta T_{i,j}^{est})^{-1} \end{aligned} \tag{4}$$

RPE compares the relative poses along the estimated $T^{est} = \{T_1^{est}, ..., T_N^{est}\}$ (from SLAM) and the reference $T^{gt} = \{T_1^{gt}, ..., T_N^{gt}\}$ (ground truth) trajectories as Eq. 4. This means that the calculation of RPE relies on GT ($\Delta T_{i,j}^{gt}$), but this value is difficult to obtain during actual robot deployment.

2.2 Topology-Based Metrics

During positioning, the robot continuously estimates its pose and integrates stable observation features into a local map. The topology-based metric algorithm indirectly evaluates the accuracy of pose estimation by assessing the consistency and stability of local map features.

MME. Mean map entropy (MME) calculates the mean value over all map points' entropy, and is defined as Eq. 5. Here, N represents the scale of the local map, which consists of a group of points. As point clouds are 3D data, in calculation, p_k represents the value of the determinant of the corresponding point cloud cluster covariance matrix of the k-th point.

$$\begin{aligned} MME &= \frac{1}{N} \sum_{k=1}^{N} h(p_k) \\ h(p_k) &= \frac{1}{2} ln |2\pi e \sum (p_k)| \end{aligned} \tag{5}$$

MPV. The Mean Plane Variance (MPV) assumes that the space is mainly composed of planes and calculates the variance of points within a range to the plane. MPV also traverses all global points and fits a plane based on the points (N) within the KNN search, as Eq. 6. The variance of current plane is equal to the minimum eigenvalue λ_{min} of the corresponding covariance matrix of current point set of p_k.

$$MPV = \frac{1}{N} \sum_{k=1}^{N} v(p_k) = \frac{1}{N} \sum_{k=1}^{N} \lambda_{min} \tag{6}$$

MOM. Both MME and MPV suffer from measurement errors due to the inability to estimate the drift of plane feature points. Furthermore, these methods have low traversal efficiency when dealing with the global point cloud, particularly in spaces with many non-plane points, which can result in further measurement deviation.

Mutually Orthogonal Metric (MOM) improves the accuracy and efficiency of plane segmentation by introducing orthogonal plane detection and traversing

the points belonging to the candidate orthogonal planes to compute the mean plane variance (according to Eq. 6) of the overall features. Orthogonal plane detection enables stable candidate point selection and filtering of non-planar features in complex scenes, resulting in a correlation metric with higher relevance compared to RPE. This, in turn, improves the accuracy and efficiency of plane segmentation.

2.3 Correlation Coefficient

Based on earlier studies, we introduce three binary correlation measures to analyze the correlation between different topology-based metrics and RPE. Furthermore, to consider the correlation between the rotation and translation components of the pose and the RPE, we introduce a new ternary correlation coefficient called the Multi-correlation coefficient.

Pearson Correlation Coefficient. Pearson product-moment correlation coefficient is defined as Eq. 7, where the numerator is the covariance of the two groups of variables X and Y, and the denominator is the power of the variance scores of the two groups of variables. Here, $\bar{(x)}$ represents the sample mean of the variable X.

$$Perason(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7}$$

Spearman Correlation Coefficient. The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the ranked variables X and Y. The ranking operation affects the covariance, which is the numerator part of the correlation coefficient calculation.

Kendall Correlation Coefficient. Kendall correlation coefficient first forms a set $(x_1, y_1), ..., (x_n, y_n)$ based on the joint random variable X and Y. Kendall correlation coefficient evaluates the rank correlation between X and Y and is insensitive to the specific distribution of the variables. It first forms a set $(x_1, y_1), ..., (x_n, y_n)$ based on the joint random variable X and Y, and determines whether each pair of points is concordant or discordant based on their relative ranks. The number of discordant pairs is denoted by DP.

$$Kendall(x, y) = 1 - \frac{2 * DP}{\binom{n}{2}} \tag{8}$$

Multi-relation Coefficient. Multi-relation [21] extends the linear correlation between two variables to more variables, and defined the new metric based on orthogonal hyperplane. For k variables Y_i, with n observations each, we can

form a kn matrix Y with kn total observations. Since each variable Y_i may be collected from a different background, we first normalize Y along each row to obtain a corresponding standardized matrix S. Using the standardized matrix S, we can calculate the sample correlation matrix R as $R = SS^T$. The Multi-relation coefficient is defined as Eq. 9, where $\lambda(R)$ represents the least eigenvalue of R, and $MR(Y_1, ..., Y_k) \in [0, 1]$ represents the strength of the multi-variable linear correlation.

$$MR(Y_1, ..., Y_k) = 1 - \lambda(R) = 1 - \lambda(SS^T) \tag{9}$$

3 The Proposed Topology Metric

3.1 Motivation

During our testing of the previous topology-based metrics in local simulation environments (Floor2 and Garage), we found that although MOM claims to improve the correlation of metrics with RPE through orthogonal plane detection, the algorithm's reliance on agglomerative clustering based on plane finding estimates suffers from the following drawbacks:

- As shown in the upper part of Fig. 1, agglomerative clustering based on normal can result in the loss of a significant number of critical plane features, which can have a significant impact on subsequent calculations.
- As shown in the bottom left of Fig. 1, clustering based on manual parameter tuning can significantly impact plane detection and generate a large number of incorrect plane results, which are represented by different colors.
- As shown in the bottom right of Fig. 1, plane detection errors can further impact normal estimation and result in multiple normal directions (represented by black lines) for the same plane.

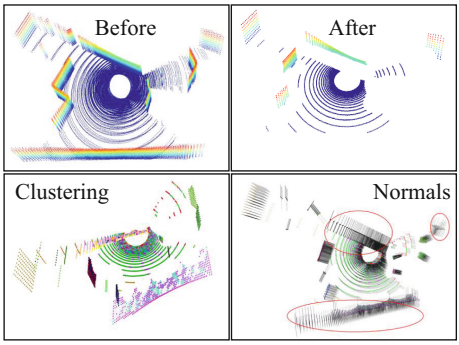


Fig. 1. Example of the drawbacks of orthogonal plane detection using a local point cloud map of five frames in Floor2.

We can see from the original point cloud (as shown in the top left of Fig. 1) that the topological structure of the point cloud map in the current area is relatively simple, and it does not contain any potential dynamic or semi-dynamic targets. However, heuristic algorithms that rely on manual parameter tuning have limited generalization abilities and are unable to understand the semantic information of the scene. As a result, when the topological information of the indoor scene is more complex and contains moving or potentially moving targets, MOM is likely to experience significant degradation.

3.2 The Workflow

We prefer to introduce the deep learning-based point cloud semantic segmentation to refine the plane detection, the whole workflow is as shown in Fig. 2.

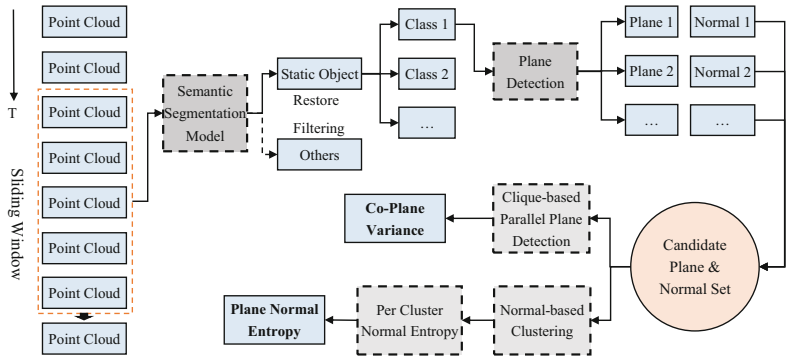


Fig. 2. The proposed workflow for calculating topology-based metrics involves a semantic segmentation model.

The algorithm first uses a sliding window to sample the continuous point cloud. In contrast to MOM, which only calculates orthogonal planes for the first frame of the point cloud, we segment each frame of the point cloud sequentially using a deep learning semantic segmentation model to obtain all semantic classes. This allows us to differentiate between static objects (such as walls, floors, and pillars) and other objects, and retain only the former after segmentation. Next, we apply robust statistical plane detection [22] to each class of static object points and estimate the corresponding plane normal. We store all candidate planes and normals for later topology metric calculation. Finally, we calculate the Co-Plane Variance (CPV) and Plane Normal Entropy (PNE) metrics based on all candidate planes and normals in the current window.

The selection of point cloud semantic segmentation models will be discussed in Sect. 4.2 of the experimental study, where we will comprehensively consider the segmentation accuracy, computational speed, and overall generalization ability of the models. Using semantic segmentation results of static targets in point

clouds allows us to enhance the accuracy and completeness of plane detection, as demonstrated in the qualitative analysis shown in Fig. 7. As a result, even though we process all point cloud frames in the sliding window, the number of planes obtained is comparable to that obtained by MOM. Additionally, we update the normals of plane points based on the detected plane normal, which further improves the accuracy of normal estimation for later processes.

3.3 Calculation of CPV and PNE

As presented in Eq. 3, each pose consists of a rotation component and a translation component, and these components have differing effects on the plane topology when there is noise in the related parts. To address these varying effects, we propose the use of Co-Plane Variance (CPV) and Plane Normal Entropy (PNE) metrics to detect the two types of noise during SLAM pose estimation with a local point feature map topology.

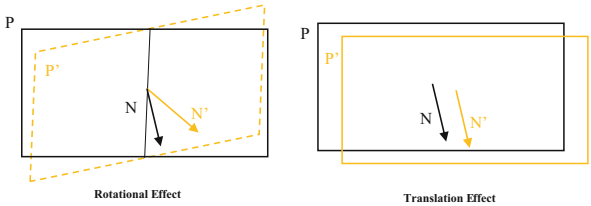


Fig. 3. Examples of how the rotation and translation of the pose affect plane changes.

Even slight translation noise can cause the same plane in two consecutive frames of point clouds to shift, as shown in Fig. 3 (right), leading to a larger variance of points on approximately co-planar planes. To mitigate this effect, we propose the use of a clique-based parallel plane detection algorithm. This algorithm identifies near co-planes based on the co-planar condition of Eq. 10 obtained from the parallel planes. This approach can be achieved through graph-based clique searching, which allows for efficient and accurate identification of near co-planes. Once the near co-planes have been identified, we calculate the point variance of all near co-planes as the final result of CPV. By using this approach, we can more accurately estimate the plane parameters and reduce the impact of translation noise on the topology of the point cloud.

$$\begin{aligned} Ax + By + Cz + D_1 &= 0 \\ Ax + By + Cz + D_2 &= 0 \\ d = \frac{|D_2 - D1|}{\sqrt{A^2+B^2+C^2}} &< \epsilon \end{aligned} \tag{10}$$

Slight rotation noise can cause the same plane in two consecutive frames of point clouds to rotate, as shown in Fig. 3 (left). Although it also leads to an increase in the variance of the plane, the change in the normal direction

is more significant. To detect the effect of rotation noise, we propose the use of Plane Normal Entropy (PNE), which is calculated based on clustering candidate normals and computing the entropy of all normals within each cluster. This approach is relatively simple and effective, as it allows us to accurately identify the change in the normal direction of the plane due to rotation noise.

4 Experimental Study

4.1 Data Introduction

To ensure the authenticity and reliability of the evaluation, we constructed a simulation environment in Gazebo [23] that was based on two local real scenes: Garage and Floor2. We deployed a Velodyne-32E (VLP-32) LiDAR with the same parameters as the actual robot configuration in the simulation environment, as shown in Fig. 4. By creating an equal-scale simulation environment, we were able to accurately evaluate the performance of our proposed method in a controlled and repeatable setting. Although both scenes contain a large number of orthogonal planes, the overall topological structures of the two local scenes are not highly regular. Moreover, Garage is a huge simulation scenario with a total area of up to 14,000 square meters. In addition, semi-dynamic objects such as tables and chairs are randomly added, which will also add certain interference to the robot's perception.

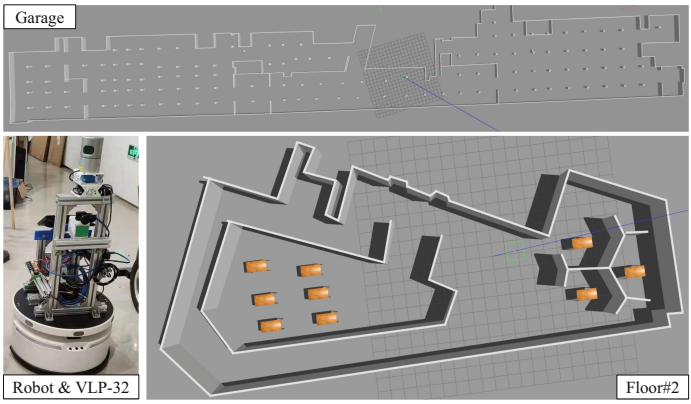


Fig. 4. Two public and real simulation scenes of local office and underground garage, and the robot equipped with VLP-32 LiDAR.

To collect the data for our evaluation, we controlled the robot to randomly move through each scene, collecting VLP-32 point cloud and the robot pose at a frequency of 5 Hz. To ensure comprehensive coverage of each scenario, we collected two sets of point cloud data with different trajectory sequences. We manually labeled the semantic information for each point cloud using the Point

Labeler tool [24], which allowed us to accurately evaluate the performance of our proposed method in detecting and classifying different objects in the point cloud data. We divided the labeled point cloud data into two non-overlapping sets, which were used for training and testing the semantic segmentation model, as well as for evaluating the topological metrics for SLAM. To illustrate the labeled point cloud data, we provide an example of the Garage scene in Fig. 5. The upper image shows the original point cloud, which is colored by height for the convenience of visualization. The bottom image shows the same point cloud data, but with colors assigned based on the manually labeled semantic information. This example illustrates the effectiveness of our labeling process in accurately identifying different objects and structures in the point cloud data.

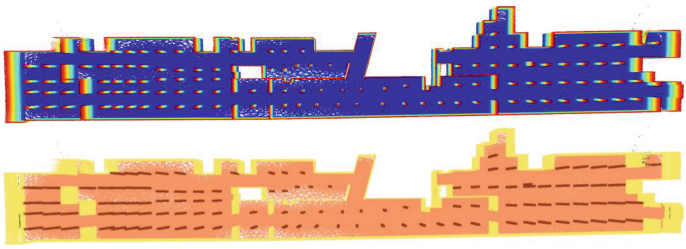


Fig. 5. Visualization of Garage point cloud overlapped scene map before and after semantic annotation (bird's eye view): coloring by height (top) and coloring by semantic label (bottom).

Table 1 provides a summary of the point cloud frames that make up the Train Set and Test Set, which consist of 3779 and 3561 frames, respectively. To ensure the generalization and augmentation of our data, we combined data collected from different scenes for training and evaluation of the deep learning-based point cloud semantic segmentation models. We also added two public scenes (Office1 & 2) collected from the Gazebo community to further enhance the diversity of our dataset. The total indoor scene semantic targets include five categories, which are ground, wall, pillar, table, and chair. To ensure the validity of our segmentation model evaluation, the two datasets are disjoint and do not overlap.

Table 1. Details of the Train set and Test set.

Scenes	Train Set (frames)	Test Set (frames)	Classes
Office1	410	427	4
Office2	375	333	4
Floor2	678	577	3
Garage	2316	2224	3
Total	3779	3561	5

4.2 Point Cloud Segmentation Performance

We trained and tested all models on a GPU server with the following specifications: CPU@Intel i9-12900K, 128 GB Memory, NVIDIA RTX 3090@24 GB. The frame per second (FPS) was calculated based on the mean value of continuous segmentation processing of 1000 frames of point cloud data. We tested three different models in this study: MinkowskiNet [25], CylinderNet [26], and SPVCNN [27]. We used the data in Table 1 in Sect. 4.1 to train and test the three deep learning models, and the results are shown in Table 2. The three models shared the same training configurations, which included 36 epochs, Stochastic gradient descent (SGD) optimizer, 0.02 learning rate with 0.0001 weight decay and 0.9 momentum. However, the batch size was different due to the constraints of video memory and model parameter scale.

Table 2. Performances of the per frame point cloud semantic segmentation results.

Model	mIoU	Per Class IoU					FPS
		Floor	Wall	Pillar	Desk	Chair	
MinkowskiNet [25]	95.84676↑	98.8446	97.7756	92.656	93.4457	96.5119	347.28 ↑
CylinderNet [26]	89.8108	96.956	95.4882	86.6077	81.4631	88.539	309.47
SPVCNN [27]	91.09102	98.6575	96.4578	87.0115	84.241	89.0873	151.48

We consider both the per class intersection-over-union (IoU), mean Jaccard (mIoU), and frames per second (FPS) in our evaluation. The mIoU is defined as shown in Eq. 11, where TP_c, FP_c, and FN_c correspond to the number of True Positive (TP), False Positive (FP), and False Negative (FN) predictions for the points of class c in the current frame, and C is the number of classes (which is 5 in our case). A higher mIoU indicates better semantic segmentation accuracy.

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c + FN_c} * 100\% \tag{11}$$

Table 2 shows that all models converge very well, but MinkowskiNet achieves the highest mIoU of 95.84, which is also the highest for per class IoU. Additionally, it is the fastest model with 347.28 FPS for point cloud of VLP-32 (∼57,600 points) on the GPU server. SPVCNN achieves the second highest mIoU, but its FPS is only about half of MinkowskiNet at around 151. CylinderNet has a slightly lower mIoU than SPVCNN at 89.8, but it achieves the second-fastest speed at 309 FPS. Figure 6 presents a detailed segmentation result (Confusion Matrix) of MinkowskiNet on the Test Set. In the confusion matrix, the values on the main diagonal represent the percentage of correctly segmented point clouds, while the remaining blocks indicate the specific category and percentage of wrongly segmented point clouds. The pillar and desk categories cause more incorrect segmentation, but the segmentation accuracy of the floor and wall categories, which contain planes, is both higher than 99%. This demonstrates that

introducing a segmentation model can provide a guarantee for subsequent correct plane detection. Therefore, we prefer to choose MinkowskiNet as the semantic segmentation model in our system to obtain semantic labels for each frame of point cloud.

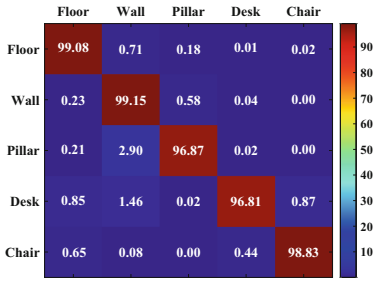


Fig. 6. Confusion matrix of the semantic segmentation results of MinkowskiNet on Test Set.

4.3 Plane Detection

Since the dataset does not include labels for point cloud planes, we can only evaluate the effectiveness of plane detection and normal estimation qualitatively. Figure 7 shows the detected planes and estimated normals of the points on each plane. By comparing these results to those from MOM in Fig. 1 (bottom), it is evident that almost all planes are correctly detected, and the directions of the point normals are also consistent.

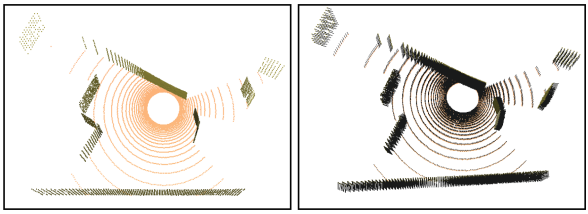
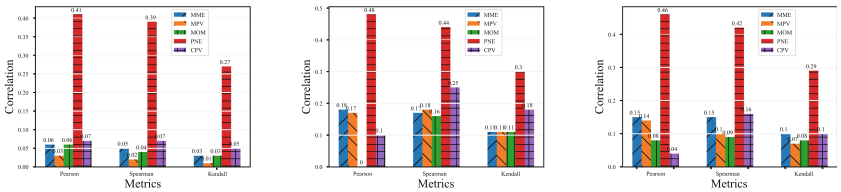


Fig. 7. The plane detection result based on semantic segmentation and robust statistic plane detection.

4.4 Binary Correlation Analysis

To analyze the correlation between RPE and different topology-based metrics, while considering the rotation and translation components of a pose, we added noise to the collected data's poses. Specifically, we added three types of noise

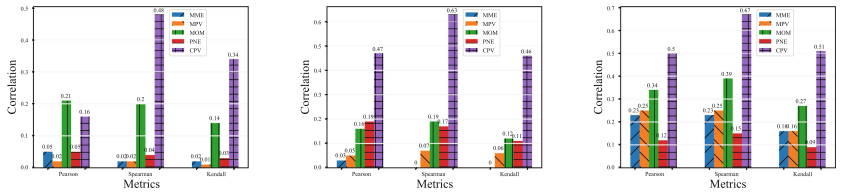
patterns, namely $[r, t, rt]$, and three levels of noise magnitudes, namely $[1, 1.5, 2]$. Figure 8, Fig. 9, and Fig. 10 illustrate the binary correlations between RPE and rotation (r), translation (t), and transformation (rt) noise at different scales.



(a) NoiseScale=1 (b) NoiseScale=1.5 (c) NoiseScale=2

Fig. 8. The binary correlation of Person, Spearman and Kendall between different topology-based metrics and the RPE under win = 5, noise mode = r, and different noise scales.

By analyzing the correlation between RPE and different topology-based metrics with varying scales of rotation noise as shown in Fig. 8, we found that PNE exhibits the highest correlation with RPE compared to other topology-based metrics. This result confirms our hypothesis in Sect. 3.3, which suggests that noise in the rotation component significantly affects the normals of parallel planar points.

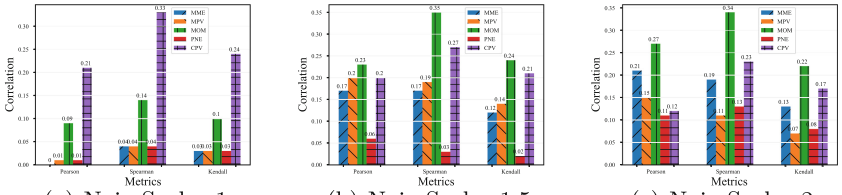


(a) NoiseScale=1 (b) NoiseScale=1.5 (c) NoiseScale=2

Fig. 9. The binary correlation of Person, Spearman and Kendall between different topology-based metrics and the RPE under win = 5, noise mode = t, and different noise scales.

By analyzing the correlation between RPE and different topology-based metrics with varying scales of translation noise as shown in Fig. 9, we found that CPV exhibits the highest correlation with RPE compared to other topology-based metrics, except for Pearson when NoiseScale = 1 (which is only slightly lower than MOM). This result confirms our hypothesis in Sect. 3.3 that noise in the translation component significantly affects the variance of points on nearby planes.

Simultaneously adding rotation and translation noise to the data poses resulted in various topology-based metrics showing no significantly strong correlation with RPE under different noise scales, as shown in Fig. 10. In some cases,



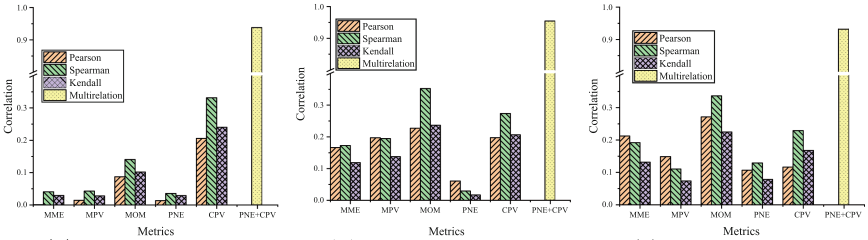
(a) NoiseScale=1 (b) NoiseScale=1.5 (c) NoiseScale=2

Fig. 10. The binary correlation of Person, Spearman and Kendall between different topology-based metrics and the RPE under win = 5, noise mode = rt, and different noise scales.

MOM showed the highest correlation, while in other cases, CPV showed the highest correlation. These results suggest that the combined effect of rotation and translation noise on pose estimation is complex, the current global topology metrics are difficult to evaluate and requires further investigation.

4.5 Triple Correlation Analysis

To evaluate the correlation between PNE, CPV, and RPE when both rotation and translation noise are added simultaneously, we used the Multi-relation of Eq. 9. As shown in Fig. 11, the Multi-relation (PNE, CPV, RPE) exhibited an extremely high correlation under different noise scales.



(a) NoiseScale=1 (b) NoiseScale=1.5 (c) NoiseScale=2

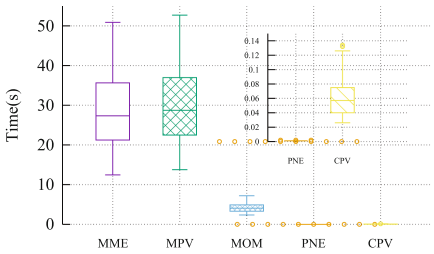
Fig. 11. The correlation of Person, Spearman, Kendall and Multi-relation between different topology-based metrics and the RPE under win = 5, noise mode = rt, and different noise scales.

Similar results were observed in the Garage scenario, as indicated by the correlation metrics. Considering redundancy, related results will not be presented here.

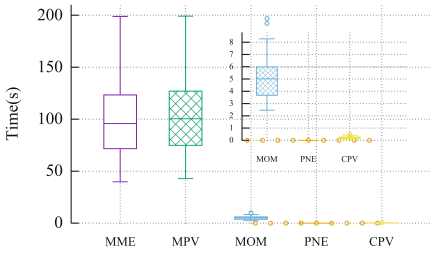
4.6 Time Complexity

Figure 12 presents a comparison of the time consumption of each topology-based metric under different sampling window sizes, namely 5, 10, and 15 frames. The

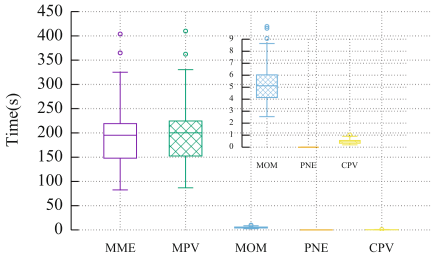
test was conducted on a mobile edge device (AMD R9-5900HX@3.3 GHz) in a single-threaded manner to assess the algorithm's feasibility and real-time performance in actual mobile robot deployment. We evaluated the entire sequence of Floor2 TestSet using a sliding window to sample the collected point clouds during calculation. The window step was set equal to the window length (e.g., $winsize = winstep = 5$) to avoid repetition of sampling. Gaussian random noise was added to each frame point cloud to simulate the point cloud distortion caused by actual robot motion. Each box-plot displays the maximum, minimum, median, 1st, and 3rd quartiles of all computation time spent for each metric, and each sub-figure shows an enlarged result.



(a) WinSize=5



(b) WinSize=10



(c) WinSize=15

Fig. 12. Time complexity statistics of each metric calculation under different window sizes, i.e., 5, 10 and 15 frames are considered in this study.

The calculation of MME and MPV requires traversing the global point cloud, and the corresponding calculation time increases proportionally when the window size of the superimposed local map increases. MOM reduces computation time by checking candidate orthogonal planes only in the point cloud of the first frame and performing KNN search on a local map based on a small set of candidate points belonging to the orthogonal planes. However, MOM still needs to perform operations such as KD-Tree construction, search, and clustering based on large-scale point clouds, and the single calculation time consumption takes several seconds. By introducing deep learning-based point cloud semantic segmentation, the calculation of PNE and CPV can be greatly accelerated. Although the neural network model itself also consumes time, it can execute independently on the GPU, and based on the analysis in Sect. 4.2, the segmentation time is about 0.0028 s (using MinkowskiNet), which is almost negligible compared to the 5/10 Hz of low-speed indoor robot. PNE and CPV further optimize the follow-up topology metric calculation, reducing the required calculation time and volatility. When processing 15 consecutive frames of point clouds, the overall calculation time does not exceed 1 s (the total time of 15 frames is 3 s under 5 Hz sampling).

5 Conclusions and Future Works

This paper proposes a plane detection algorithm based on neural network point cloud semantic segmentation optimization, starting from topology-based SLAM pose estimation, to promote the development of autonomous localization techniques for mobile robots in indoor environments. The algorithm combines robust statistical plane detection with optimized extraction of point cloud plane features to ensure comprehensive, complete, and accurate stable spatial topological features. We also analyzed the potential impact of rotation and translation noise in SLAM pose estimation on plane features and proposed two evaluation metrics, Co-Plane Variance (CPV) and Plane Normal Entropy (PNE), respectively. The proposed algorithm was qualitatively and quantitatively evaluated using point cloud and pose data from different simulation scenarios in Gazebo, which confirmed the validity of the proposed hypothesis and the rationality of the corresponding topology-based metrics.

Although we have validated the strong correlation of CPV+PNE with RPE in the presence of both rotation and translation noise through triple correlation (Multirelation), a feasible and unified quantitative calculation method for the two metrics is still lacking. We plan to improve this issue in the future and conduct experiments on datasets collected from real-world scenarios to further validate our proposed approach.

ACKNOWLEDGMENTS. This research was fully supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY23F020026, and partly supported by Tianmushan Laboratory Research Project TK-2023-B-010 and TK-2023-C-020.

References

1. Ohno, K., Tsubouchi, T., Shigematsu, B., Yuta, S.: Differential GPS and odometry-based outdoor navigation of a mobile robot. Adv. Robot. **18**(6), 611–635 (2004)
2. Xu, Y., Shmaliy, Y.S., Ahn, C.K., Tian, G., Chen, X.: Robust and accurate UWB-based indoor robot localisation using integrated EKF/EFIR filtering. IET Radar Sonar Navig. **12**(7), 750–756 (2018)
3. Zhang, J., Singh, S.: LOAM: lidar odometry and mapping in real-time. In: Robotics: Science and Systems, vol. 2, pp. 1–9. Berkeley, CA (2014)
4. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D.: LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5135–5142. IEEE (2020)
5. Shan, T., Englot, B.: LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4758–4765. IEEE (2018)
6. Livingstone, D., Miranda, E.: Orb3: adaptive interface design for real time sound synthesis & diffusion within socially mediated spaces. In: Proceedings of the 2005 Conference on New Interfaces for Musical Expression, pp. 65–69 (2005)
7. Wang, Y., Tan, R., Xing, G., Wang, J., Tan, X., Liu, X.: Samba: a smartphone-based robot system for energy-efficient aquatic environment monitoring. In: Proceedings of the 14th International Conference on Information Processing in Sensor Networks, pp. 262–273 (2015)
8. Sumikura, S., Shibuya, M., Sakurada, K.: OpenVSLAM: a versatile visual slam framework. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 2292–2295 (2019)
9. Helmberger, M., Morin, K., Berner, B., Kumar, N., Cioffi, G., Scaramuzza, D.: The hilti SLAM challenge dataset. IEEE Robot. Autom. Lett. **7**(3), 7518–7525 (2022)
10. Michael Grupp. evo: Python package for the evaluation of odometry and slam (2017). github.com/MichaelGrupp/evo
11. Burri, M., et al.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. **35**(10), 1157–1163 (2016)
12. Sturm, J., Burgard, W., Cremers, D.: Evaluating egomotion and structure-from-motion approaches using the tum RGB-D benchmark. In: Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS), vol. 13 (2012)
13. Droeschel, D., Stückler, J., Behnke, S.: Local multi-resolution representation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5221–5226. IEEE (2014)
14. Razlaw, J., Droeschel, D., Holz, D., Behnke, S.: Evaluation of registration methods for sparse 3d laser scans. In: 2015 European Conference on Mobile Robots (ECMR), pp. 1–7. IEEE (2015)
15. Kornilova, A., Ferrer, G.: Be your own benchmark: no-reference trajectory metric on registered point clouds. In: 2021 European Conference on Mobile Robots (ECMR), pp. 1–8. IEEE (2021)
16. Strasdat, H.: Local accuracy and global consistency for efficient visual SLAM. Ph.D. thesis, Department of Computing, Imperial College London (2012)
17. Guclu, O., Can, A.B.: Fast and effective loop closure detection to improve slam performance. J. Intell. Robot. Syst. **93**, 495–517 (2019)

18. Koide, K., Miura, J., Yokozuka, M., Oishi, S., Banno, A.: Interactive 3d graph slam for map correction. IEEE Robot. Autom. Lett. **6**(1), 40–47 (2020)
19. Ouyang, Z., Zhang, C., Cui, J.: Semantic SLAM for Mobile Robot with Human-in-the-Loop. In: Gao, H., Wang, X., Wei, W., Dagiuklas, T. (eds.) CollaborateCom 2022, Part II. LNICS, vol. 461, pp. 289–305. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-24386-8_16
20. Chen, X., et al.: OverlapNet: loop closing for lidar-based slam. arXiv preprint arXiv:2105.11344 (2021)
21. Drezner, Z.: Multirelation–a correlation among more than two variables. Comput. Stat. Data Anal. **19**(3), 283–292 (1995)
22. Araújo, A.M.C., Oliveira, M.M.: A robust statistics approach for plane detection in unorganized point clouds. Pattern Recogn. **100**, 107115 (2020)
23. Koenig, N., Howard, A.: 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, pp. 2149–2154. IEEE (2004)
24. Behley, J., et al.: SemanticKITTI: a dataset for semantic scene understanding of LiDAR sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
25. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3075–3084 (2019)
26. Zhu, X., et al.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9939–9948 (2021)
27. Tang, H., et al.: Searching efficient 3D architectures with sparse point-voxel convolution. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020, Part XXVIII. LNCS, vol. 12373, pp. 685–702. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_41