



# VoxelPlane-Reloc: an indoor scene voxel plane relocalization algorithm

Lulu Suo<sup>1</sup> · Bin Wang<sup>1</sup> · Longxiang Huang<sup>2</sup> · Xu Yang<sup>2</sup> · Qian Zhang<sup>1</sup> · Yan Ma<sup>1</sup>

Received: 18 September 2023 / Accepted: 1 January 2024 / Published online: 29 February 2024  
© The Author(s) 2024

## Abstract

We present VoxelPlane-Reloc, a novel indoor plane relocalization system based on voxel point clouds, designed for use with depth cameras. First, we propose an adaptive weighted plane extraction model that allows for dynamic adjustment of the correlation between points and plane accuracy. Second, we construct a plane merging model based on voxel growth, which employs a voxel neighborhood growth strategy to handle unmerged planes and allows for the merging of under-growing planes. Third, we present an incremental approach for plane input and propose a strategy for triplet selection and evaluation based on the structural constraints of the planes. This system relies solely on point clouds for relocalization and does not depend on other information, such as RGB data. We extensively evaluate the system on four datasets, and the experimental results demonstrate that the system can accurately and quickly perform relocalization with an average precision of 99.37%. The time for relocalization is improved by 92.43% compared to previous plane relocalization systems, and it exhibits strong robustness to indoor plane structures.

**Keywords** Relocalization · Depth camera · Adaptive weight function · Plane merging model · Structural plane constraint

## Introduction

Relocalization, also referred to as scene recognition, is a vital component in simultaneous localization and mapping (SLAM) [1, 2]. It serves to adjust the current position in cases where tracking is unsuccessful and to mitigate drift

resulting from prolonged system movement. Various sensors, such as laser radar [3–5] and camera [6, 7], can be utilized for relocalization. Laser radar has the capability to precisely capture environmental information [8, 9], but it is expensive and has a limited field of view, making it challenging for widespread implementation. On the other hand, cameras, despite having slightly lower accuracy, are gaining popularity in the field of computer vision due to their affordability and wide field of view. Additionally, with the increasing availability of depth cameras, it is now possible to directly measure the depth information of the scene, creating favorable conditions for relocalization tasks.

Methods for relocalization using depth cameras typically rely on feature extraction to describe the scene and feature matching to estimate pose. Most methods extract point and line features from the image and then search for candidate relocalization frames in a database [10–13]. While these methods are straightforward and computationally efficient in terms of feature description and matching, they encounter practical challenges in localization processes. For example, extracting sufficient features in indoor low-texture environments, such as walls and ceilings, can be problematic, leading to feature loss. Additionally, variations in lighting and viewing angles within the image can affect feature data association, making accurate estimation during subsequent

---

Lulu Suo and Bin Wang contributed equally to this work.

✉ Bin Wang  
binwang@shnu.edu.cn

Lulu Suo  
sll745353@163.com

Longxiang Huang  
longxiang.huang@deptrum.com

Xu Yang  
xu.yang@deptrum.com

Qian Zhang  
qianzhang@shnu.edu.cn

Yan Ma  
ma-yan@shnu.edu.cn

<sup>1</sup> The College of Information, Mechanical, and Electrical Engineering, Shanghai Normal University, Guilin Street, Shanghai 201400, China

<sup>2</sup> Shenzhen Guangjian Technology Co., Ltd, Shatou Street, Shenzhen 518048, China

feature matching more challenging. The accuracy of relocalization results largely depends on feature detection and association, so noise caused by factors like dynamic objects and occlusion during the measurement process can accumulate errors, particularly in indoor scenes. These inevitable issues are challenging to address solely using point and line features.

In recent years, the fields of SLAM and robotics has extensively studied the indoor environments. These indoor settings contain advanced and representative features, particularly planes. Plane features exhibit greater resilience to noise points and can effectively handle lighting changes and moving objects in indoor scenes. Moreover, they enable faster and more accurate feature association. Consequently, integrating plane features into indoor relocalization can significantly enhance its performance. In consideration of the aforementioned factors, efforts have been devoted to the relocalization of planar features [3, 6, 10, 14–16], yielding notable accomplishments. Nevertheless, several challenges endure. First, the accuracy of planar relocalization hinges on the precision of planar extraction. Therefore, addressing noise points in point clouds and fully recovering spatial plane structures for robust plane estimation are imperative for successful relocalization. Second, the planar relocalization process involves matching and associating multiple sets of planes, where the time consumption is directly proportional to the number of planes in the map and the relocalization frame. Moreover, storing plane information associations consumes significant computational resources, posing challenges in achieving real-time and lightweight performance.

Building upon these considerations, we propose a real-time and lightweight indoor plane relocalization system for robust pose estimation. Leveraging a depth camera, we detect and reconstruct 3D point cloud information, partitioning it into voxels for efficient plane extraction. Diverging from other plane relocalization algorithms that directly extract planes after segmentation, our approach decomposes the plane extraction process into two pivotal components. First, within the voxel, we introduce an adaptive weight function based on the depth camera's measurement model, accounting for the influence of points at different positions on plane estimation. Second, to reconstruct spatial plane structures, even when the same plane is partitioned into different voxels, we identify correlations and merge strongly related planes. Recognizing the complexities introduced by plane matching and aiming to meet the real-time and lightweight demands of SLAM, we incorporate plane structural constraints. These constraints expedite fast and robust plane feature matching, optimizing the best pose estimation based on these relationships. Additionally, we design an incremental model for plane input to minimize computational resource wastage and reduce memory usage. In summary, our main contributions are threefold:

1. We design an adaptive weighted plane extraction model that conceptualizes point clouds as a compilation of voxels. Employing a weight function, we dynamically modulate the impact of points within the voxels on the precision of plane estimation, with the objective of attaining more robust planes.
2. We exploit a plane merging model grounded in voxel growth, entailing the fusion of planes within the immediate voxel growth neighborhood. Additionally, we reintegrate planes with inadequate growth to reinstate a more comprehensive plane structure.
3. We propose an incremental plane input model and formulate two strategies for the selection and evaluation of plane triplets, guided by plane structural constraints. These strategies are designed to expedite the matching process, facilitating real-time and lightweight reorientation.

The rest of this paper is structured as follows. Section “Related work” reviews existing relocalization research. Section “Method” presents a comprehensive explanation of our methodology, while Section “Experiment” showcases our experimental findings. Lastly, Section “Conclusions” summarizes the key points of this paper and explores potential avenues for future research.

## Related work

We primarily focus on feature-based relocalization methods and categorize them into different types of features, including local features, global features, and planes. Our algorithm is closely related to those based on planar features.

## Local features-based relocalization

In the initial stages, local characteristics served as inputs for relocalization, primarily due to the emergence of the bag-of-words (BoW) model. Originally developed for text representation, the BoW model found applications in computer vision for effectively modeling image features [17]. This technique intelligently represents images as words to establish their relationships and has since been extensively referenced in subsequent studies. Angeli et al. [18] employed local shape and color features to construct an online visual dictionary, relying on Bayesian filtering to estimate the probability of relocalization. Garcia-Fidalgo et al. [19] introduced an indexing method for binary features, combined with an inverted index, to obtain relocalization candidates in real-time. To meet the demands of offline SLAM, Cummins et al. [20] adopted a method that employed concentration inequalities for fast approximate multiple hypothesis test-

ing and computed relocalization probability within a filtering framework. However, the speed improvement of this method heavily depends on the amount of feature data. Labbe et al. [21] proposed a memory management approach that relocated positions with high observation frequency, significantly enhancing real-time performance but potentially introducing feature quantization errors. To mitigate the impact of dynamic objects in the scene on relocalization accuracy, Gao et al. [22] used a dual Gaussian model to distinguish foreground and background, effectively identifying dynamic features. Yang et al. [23] employed a geometric constraint method to filter dynamic feature points in the scene, achieving accurate pose estimation in dynamic scenes, although this approach is less sensitive to feature extraction in low dynamic scenes. Hang et al. [24] improved the geometric constraint-based dynamic–static feature point classification scheme and developed a more refined feature partitioning strategy based on it.

### Global features-based relocalization

In contrast to local features, global features are directly computed as descriptors, and relocalization is achieved by assessing the correlation between image descriptors. Oliva et al. [25] introduced the concept of Gist global descriptors, which extract image information using Gabor filters with varying directions and frequencies, condensing it into a single vector representation. Ulrich et al. [26] employed the histogram of panoramic color images in combination with nearest-neighbor learning for image matching. Sunderhauf et al. [14] first downsample the image and then compute BRIEF descriptors around the center of the downsampled image. With the advent of deep learning, researchers have started utilizing deep learning techniques to design improved global descriptors. Zhao et al. [15] compress point clouds into normal distributions transform (NDT) units using 3D Gaussian distribution transformation and learn global descriptors from them. Hou et al. [27] introduced a hierarchical transformer that enhances the correlation between local neighboring points and the contextual dependency between global points.

### Plane-based relocalization

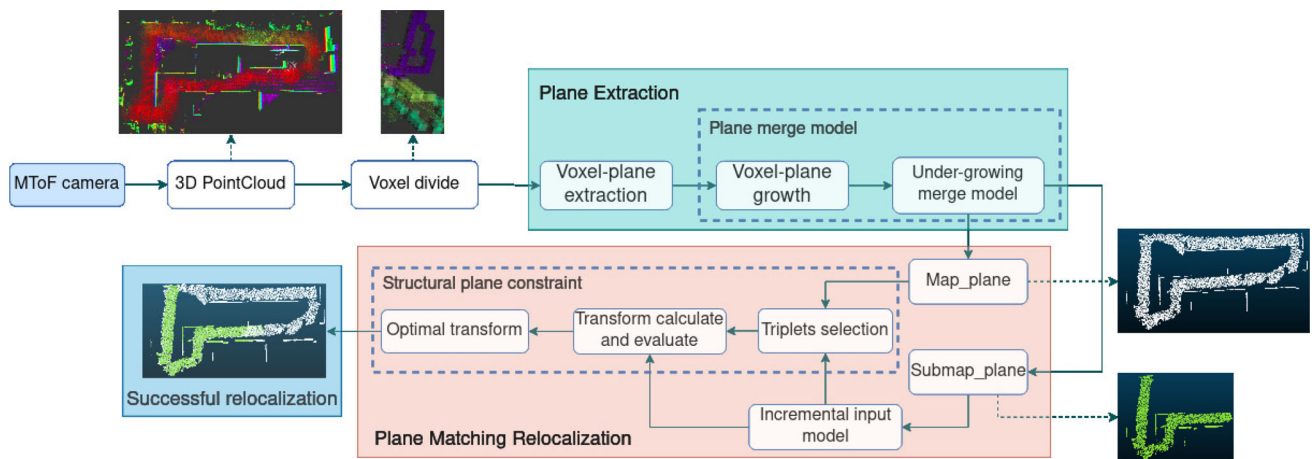
As an advanced capability, plane possess attributes such as fundamental data correlation and exceptional performance in low-texture environments. Consequently, plane features are widely acknowledged as valuable inputs for relocalization, particularly in indoor settings. Sun et al. [28] proposed using a probabilistic model to extract planes and mitigate the influence of depth noise. Lin et al. [29] divided the point cloud into NDT cells and used the RANSAC algorithm to extract plane features in each cell. Zhang et al.

[30] generated hypothetical planes based on the detection of plane features, which served as the foundation for pose estimation. Dominik Belter et al. [31] used deep learning to detect plane features and jointly optimize visual cues, normal vectors, and plane parameters. Zi et al. [32] proposed registering point features to corresponding plane features and then combining point features for pose estimation. Li et al. [33] integrated point, line, and plane features for pose estimation, imposing certain constraints on the plane features in indoor scenes. Although this significantly improved the accuracy of pose estimation, the computational complexity was high. Consequently, the relocalization thread only used the pose estimated from the front end and correlated it with point features. Shu et al. [34] followed a similar approach by incorporating point, line, and plane features, with the addition of line features in the relocalization process for more robust pose estimation. However, this method may not perform well in low-texture scenes where sufficient features cannot be extracted to support relocalization. J. Wietrzykowski et al. [35–37] proposed a method using RGB-D cameras to extract plane features by leveraging color information and the flood-fill algorithm. They transformed pose estimation into a calculation of Gaussian kernel probability distributions for relocalization. However, this method has limitations in plane segmentation, resulting in poor edge detection of the planes.

While the aforementioned work has yielded promising results in certain cases, it still encounters challenges due to inherent limitations. One such challenge is that extracting a large number of local features can hamper real-time performance. Moreover, employing local feature matching across the entire map can reduce the matching success rate. Conversely, global features heavily rely on training with outdoor large-scale datasets and are not suitable for indoor conditions, posing difficulties in achieving optimal results. Existing plane relocalization systems often suffer from inaccurate plane estimation due to coarse plane segmentation and encounter real-time processing difficulties during brute-force plane matching. Our approach offers a different perspective by leveraging a voxel structure to streamline the plane extraction process, enhancing robustness and accuracy. Furthermore, to address the computational resource demands introduced by incorporating plane features, we integrate structural constraints and pruning strategies based on the indoor scene itself, ensuring compatibility with the real-time requirements of relocalization.

## Method

We have developed a model for extracting planes and a relocalization strategy based on 3D point clouds from the depth camera, which is called VoxelPlane-Reloc. The overall methodology of our model is depicted in Fig. 1.



**Fig. 1** Overview of our architecture

Our main objective is to establish the relationship between the global map and the local map to facilitate matching. We have developed a relocalization system that integrates a depth camera with plane feature extraction. This system comprises three modules: voxelplane extraction module, plane merging module, and plane matching module. In the subsequent sections, we will provide a detailed explanation of our work.

### Voxelplane extraction

To ensure the precise and comprehensive extraction of planes in space, we have introduced the concept of voxels. Initially, the input point cloud information is mapped to an  $n \times n \times n$  ( $m$ ) voxel grid, followed by the extraction of planes from the points within each voxel. Traditional plane extraction strategies fall short of meeting the precision requirements of our system. For instance, the least squares (LS) methods are highly sensitive to outliers and may result in local convergence issues [38, 39]. RANSAC may lead to non-uniqueness of the plane [7, 40]. Algorithms based on feature space transformation, such as PCA [41, 42] and Hough Transform [43], require an analysis of point cloud quality. To address this, we incorporated information about point cloud quality and noise characteristics obtained during the actual collection process using the depth camera. Subsequently, we devised an adaptive weighting function. This function optimizes the initial plane parameter estimation by allocating varied contributions to the plane extraction accuracy at different positions, thereby enhancing the overall robustness of plane extraction.

First, the adjustment involves the extraction of the initial plane  $P_0$  through PCA decomposition applied to all points within the voxel. The theoretical equation for the plane is formulated as  $n \cdot p_i - d = 0$ . However, due to potential disturbances in measurements, certain points may exhibit deviations from this ideal equation. Consequently, the opti-

mization objective function is defined as follows:

$$S = \sum_{i=0}^{m-1} (n \cdot p_i - d)^2, \quad (1)$$

where  $n$  and  $d$  serve as the parameters characterizing the plane,  $n$  specifically designates the normal vector of the plane, while  $d$  signifies the distance from the plane to the origin. The variable  $m$  comprehensively encapsulates all points situated on the plane within the voxel, and the symbol  $p$  is employed to denote the coordinate values associated with these points.

The current problem is to update and obtain the optimal plane parameters  $n$  and  $d$  such that the objective function  $S$  approximates to 0. However, given the presence of noise points in actual measurements, we aim to derive more robust plane parameters. To achieve this, we consider assigning different weights to each point within the voxel, thereby adaptively distinguishing between inliers and outliers of the plane. Previously, several works have explored the use of different weight functions to obtain robust estimates. Tukey function [44], for instance, proposed reducing and eliminating outliers to determine the weights and selected the square model as the basis for weight reduction. Danish function [45], on the other hand, employed an exponential model to shrink values that deviated significantly from the residuals. Hampel function [46] and IGG-3 function [47] combined these approaches, using a polynomial model with three modes to allocate weights. Through a series of experiments, it was found that while there is not a substantial difference in the final performance of planar detection using different weight functions, the best results were achieved by combining the exponential and square models.

Although these weight functions are carefully designed, they all rely on iterative residual calculations to determine weight categories. However, this approach is not entirely con-

sistent with the actual plane extraction situation. Through experiments, we identified an issue related to inaccurate original depth images when employing a depth camera to measure planar point clouds, particularly at the image edges [48]. The edges exhibit lower signal intensity compared to the center and are more susceptible to noise interference. Consequently, for a plane, the contribution of the center points and the surrounding points, especially the edges, to the plane is not uniform. To address this phenomenon, we propose an adaptive weight function that aligns with the depth camera point cloud measurement model:

$$W_i = \begin{cases} -\alpha r_i^2 + 1, & \text{if } |r_i| \leq \sigma \\ \exp(-\beta r_i^2), & \text{if } |r_i| > \sigma, \end{cases} \quad (2)$$

where  $W_i$  denotes the weight value corresponding to point  $p_i$  and  $r_i$  represents the distance from a specific point  $p_i$  to the center of the plane, the parameter  $\alpha$  influences the steepness of the square model, leading to a faster decrease in the weight function. Simultaneously,  $\beta$  governs the convergence speed of the exponential model to 0. In our pursuit of extracting more precise plane parameters, we align the objective of the weight function by assigning greater weights to points in closer proximity to the center. This is achieved by selecting a smaller  $\alpha$  when the gradient of the plane model is diminutive. Conversely, lesser weights are assigned to points farther away from the center, indicating a larger  $\beta$  when the gradient of the exponential model is more substantial. The parameter  $\sigma$  delineates the boundary of the weighted calculation model for partitioning plane points. It is determined as half of the distance between the center point of the extracted plane and the farthest point from the center, contributing to the accurate delineation of the weighted calculation boundaries.

This weighting function segregates the points used for plane extraction into two groups. It leverages the center of the plane obtained from PCA as prior information and establishes a spherical model around this center point within the voxel. By evaluating the relationship between the points and this sphere, it reduces the weighting coefficient of the points at the edges, thereby diminishing their influence on plane extraction. Ultimately, the objective function optimized is as follows:

$$\min \sum_{n,d} |W_i (n \cdot p_i - d)|^2, \quad (3)$$

where  $p_i$  signifies the set of all points constituting the internal plane of a voxel, and  $W_i$  represents the weight coefficient assigned to point  $p_i$ , the parameters  $n$  and  $d$  characterize the plane parameters subject to optimization. The primary objective is to ascertain the weight contribution of each individual point to the plane. Given this objective, the optimization aims to derive optimal values for  $n$  and  $d$  that minimize the cumu-

lative sum of point-to-plane distances, effectively converging toward zero.

## Plane merging

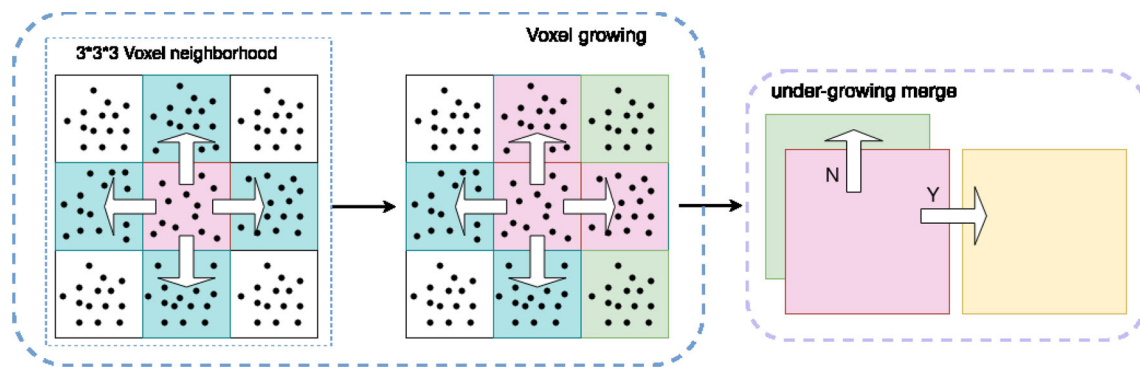
With the introduction of the voxel concept, we cannot guarantee the seamless division of the same plane within voxel space. To reconstruct a comprehensive plane structure, we have introduced a plane merging module. This module consolidates all point clouds representing the same plane in adjacent voxels, culminating in a more accurate and complete plane estimation. We propose a plane merging strategy: voxel growth and under-growing merge model. Further details are provided in Fig. 2.

The representative plane detection algorithm for merging on the same plane is the region growing algorithm [49]. It operates by applying region growing on a patch in a 2D depth map to identify a plane. To adapt this strategy to our 3D voxel point cloud model, we have made certain enhancements. To facilitate plane expansion using voxels, we examine the voxels within the  $3 \times 3 \times 3$  vicinity and their associated plane data. The process of voxel plane growth involves several steps. Initially, we create a seed list  $V_1$ , which includes all voxels containing planes. This list is denoted as  $V_1 = \{\forall v_i \in V_1, P_i \in v_i\}$ , where  $v_i$  represents the voxels, and  $P_i$  indicates the presence of a plane within that voxel. Next, we initiate the growth process by selecting the first voxel from the list as the seed. We then iterate through its 26 neighboring voxels. If a neighboring voxel meets specific criteria, it can be merged:

1. There is an ungrown plane within the voxel.
2. The plane's parameters ( $n$  and  $d$ ) in the neighboring voxel closely resemble those of the seed voxel, exhibiting a difference smaller than predefined thresholds. While the ideal scenario would involve identical plane parameters ( $n$  and  $d$ ) for merged planes in adjacent voxels, the presence of noise points during the actual measurement process introduces minor errors in plane estimation. Therefore, we establish a threshold of sufficient sensitivity to accommodate these small errors, ensuring the comprehensive merging of segmented planes.

And then, we proceed to update the merged planes and upgrade the voxel seed list to  $V_2$ , removing the seed voxels and the voxels that have already been merged in the previous stages. This updating process is carried out once the growth process of each voxel is completed. In summary, we repeat the aforementioned process continuously until the seed list becomes empty.

Following the voxel growth in the previous stage, certain small planes that were initially segmented may still be separated due to insufficient growth or viewing angles. To



**Fig. 2** Overview of the plane merging model based on voxel growth. The system comprises two essential components: the voxel growth and under-growing models. The cross section of the voxel growth cube is depicted as a large square composed of  $3 \times 3$  smaller squares. On the left square, the pink area designates the seed voxel, the blue area signifies adjacent voxels undergoing evaluation for growth. On the right square, the pink area represents voxels that have successfully grown, the blue

area designates voxels under evaluation for growth, and the green area indicates voxels that do not meet the growth conditions. In the under-growing model, the pink plane represents the initial plane slated for merging, yellow signifies the plane eligible for merging (denoted by Y, indicating that the merging condition is satisfied), and green designates the plane ineligible for merging (denoted by N, indicating that the merging condition is not met)

address this concern, we have proposed additional operations for merging these planes. If a plane is identified as a distinct and similar fragment due to inadequate growth in the actual area, it needs to be merged. This merging process must satisfy three specific conditions:

1. The angle between the normal vectors of the two planes is sufficiently small, indicating that the planes are nearly parallel.
2. The angle formed by the vectors connecting the centroids of the two planes is approximately 90 degrees, suggesting that they are at the same depth.
3. The merging process is constrained to the extent of the plane's radius multiplied by a scaling factor to prevent excessive merging.

After performing checks for parallelism, same depth, and proximity, we incorporate the small plane input into the larger plane. We then proceed to recalibrate the plane parameters to obtain more precise and consistent observations about the plane.

### Plane matching

Our plane matching module expands upon the traditional algorithm framework of *planoloc* [36]. It calculates the transformation by selecting sets of three planes and establishes a probability density function (PDF) to model the relationship between pose transformations. We depart from the previously mentioned method to establish a real-time and lightweight relocalization system by exclusively incorporating point cloud information, eliminating dependencies on RGB data. Concurrently, we propose an incremental plane

input model and integrate structural constraints based on the Manhattan assumption to eliminate redundant plane associations. For the sake of clarity and streamlined expression, we consistently employ the notation  $P^M$  to denote planes within the map and  $P^S$  to signify planes within the submap in the subsequent text.

### Incremental input model

To effectively prevent misalignment resulting from the similarity of indoor scene planes, our algorithm design stipulates the presence of at least 5 sets of matching planes for plane relocalization. The threshold setting should align with the unique scene identification strategy in subsequent transform evaluations, ensuring compatibility with the distinctiveness of planar scenes. Once the correct 5 sets of planes are found and the correct transformation relationship is calculated, the input of additional planes not only has no impact but also introduces redundancy.

We have devised a unique model for incrementally inputting submap planes, as illustrated in Fig. 3. To filter the input submaps comprehensively, we systematically select five representative planes with a defined angular separation in space and substantial size as the initial input. Subsequently, we identify three pairs of matching triplets from the newly introduced submap input planes and the existing map, computing the transformation relationships for each triplet. The remaining planes in the initial submap are then subjected to transformation based on the calculated relationships. We verify whether the majority of these planes align with their corresponding counterparts in the map. Concurrently, we restore the original order to scrutinize the plane correspondence. If the preponderance of planes successfully

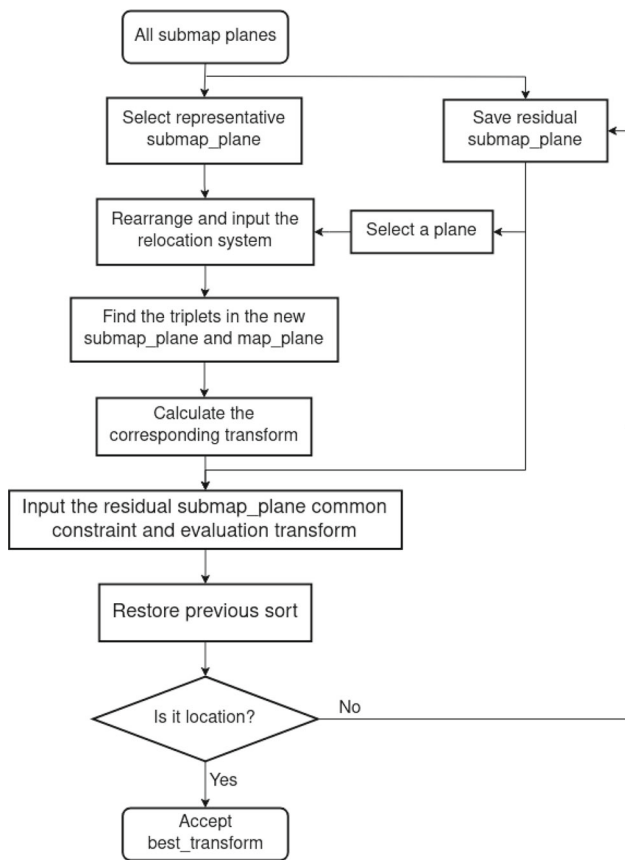


Fig. 3 Overview of the incremental input model

aligns with their correct matches, indicative of successful repositioning, we output the transformation relationship for the optimal triplet. In cases where this alignment is not achieved, we proceed to select the subsequent plane input from the remaining planes, update the submap accordingly, and iteratively repeat the aforementioned steps until successful repositioning is attained.

**Triplet selection**

For plane relocalization, we utilize the point cloud planes from both the global map and the local map. Typically, three sets of matched planes are sufficient to establish the pose transformation relationship. The selection of triplets is directly influenced by the number of planar maps and submaps. In prior methodologies, triplets were chosen using color histogram filtering and brute-force matching, resulting in a substantial number of triplets. This led to significant computational and evaluative demands in terms of time and space. Our approach incorporates the Manhattan hypothesis theory, which reduces the number of triplets based on the structural constraints of indoor scenes. Furthermore, we have implemented algorithm pruning and acceleration techniques to enhance the real-time performance of the relocalization. A

detailed description of the key strategies used in this module will be provided in the following section.

*Ground plane constraint.* The ground plays a pivotal role in indoor scenes as it provides a stable support surface for all viewpoints and scenarios. It can be regarded as the primary group of matching planes. By leveraging ground constraints, the initial set of matches can be determined, designating the ground plane. Following the preceding plane extraction and merging process, only one ground plane remains, considerably reducing the number of triplets. After determining the map plane  $P^M$  and the submap plane  $P^S$ , we calculate the angle relationship between each plane and the actual normal vector  $n_g$  of the ground. Subsequently, we select the ground planes  $P_g^M$  and  $P_g^S$  accordingly.

$$\langle n_g^M, n_g \rangle = \arccos(n_g^M \cdot n_g) < \tau_g, \tag{4}$$

$$\langle n_g^S, n_g \rangle = \arccos(n_g^S \cdot n_g) < \tau_g, \tag{5}$$

where  $n_g^M$  and  $n_g^S$  represent the normal vectors of the ground planes  $P_g^M$  and  $P_g^S$  respectively, with the true ground normal vector  $n_g$  is  $(0, -1, 0)^T$ . The symbol  $\langle n_1, n_2 \rangle$  denotes the angle measurement between two vectors. The threshold  $\tau_g$  is specifically designed to filter the ground plane. Ideally, the angle between  $n_g^M$  ( $n_g^S$ ) and  $n_g$  should be  $0^\circ$ . However, due to the presence of noise points in the actual measurement process, the estimation of the ground normal vector is not entirely accurate. Additionally, to accommodate a larger field of view, cameras are typically installed with a certain tilt angle. Consequently, a threshold aligned with the practical situation needs to be established.

*Orthogonal plane constraint.* The walls, floors, and ceilings in each room are either mutually orthogonal or parallel, adhering to the Manhattan assumption. To ensure accurate pose calculation, we have devised the following approach for filtering triplets:

1. The angle variation between the chosen  $P^M$  and  $P^S$  falls within a specific range, determined by the plane structure. Given that parallel or coplanar planes serve a similar role in transformation calculations, we prefer the three planes selected in the triplet to be approximately perpendicular to each other. However, the actual calculation angle difference may not be entirely accurate, and the existence of inclined planes in space cannot be ruled out. Therefore, the range is set to  $[40^\circ, 140^\circ]$ .
2. The distance between the selected three map planes in the triplet should not be too far apart, as it would make it challenging for them to appear in the same scene within the submap. Traditionally, the distance between planes is represented by their closest points. However, in certain cases, this method can lead to inaccuracies, inconsistencies, and

complexities, especially when dealing with non-convex shapes. To address this, we have developed a novel distance representation method where the maximum distance from all decentralized points to the origin is considered as the radius, signifying that all points in the plane are enclosed within a sphere. Thus, comparing the distance between two planes is converted into comparing the distance between two spheres in 3D space.

- During the extraction process,  $P^M$  and  $P^S$  exhibit a specific dimensional relationship due to perspective constraints. Typically, the area of the same plane in the map is equal to or larger than the area of the submap. To account for this, we utilized a grid-based method to estimate the planar area based on their relative relationship. Initially, the point cloud was divided into smaller patches with dimensions of  $l \times l$ . Then, we counted the number of point clouds in each patch to determine the relative relationship between the ground plane area and the submap plane area.

### Transform calculate and evaluate

After triplet filtering, determining the optimal triplet to attain the correct pose transformation relationship becomes imperative. This entails the evaluation of triplets. By leveraging the matching relationship of the three planes, we can calculate the corresponding pose transformation to verify the correctness of the matching relationship. By minimizing the disparity between the normal vectors  $n_i^M$  and  $n_j^S$  of  $P_i^M$  and the transformed current submap  $P_j^S$ , a quaternion  $r = [r_x \ r_y \ r_z \ r_w]^T$  is derived to signify the rotation. Similarly, the discrepancy between the offset components  $d_i^M$  and  $d_j^S$  of  $P_i^M$  and the transformed  $P_j^S$  is minimized to determine the translation component  $t = [t_x \ t_y \ t_z]^T$ .

$$E_r = \sum_{(i,j) \in \mathcal{T}} \left| W(r)^T Q(r) n_j^S - n_i^M \right|^2, \quad (6)$$

$$E_t = \sum_{(i,j) \in \mathcal{T}} [(d_i^M - d_j^S - (n_j^S)^T t)^2], \quad (7)$$

where  $E_r$  denotes the disparity between the transformed  $n_j^S$  and  $n_i^M$ , while  $E_t$  signifies the difference between the transformed  $d_j^S$  and  $d_i^M$ . Ideally, these discrepancies should be null, yet in practical scenarios, a certain degree of error persists. Our objective is to determine optimal values for  $r$  and  $t$  that minimize these discrepancies to the greatest extent possible. The pair  $(i, j)$  within the triplet  $\mathcal{T}$ , denoted as  $(i, j) \in \mathcal{T}$ , represents a set of corresponding planes. The matrices  $Q(r)$  and  $W(r)$  are two essential matrix functions related to quaternions and can be defined as follows:

$$Q(r) = \begin{bmatrix} rI + K(r) & r \\ -r^T & r \end{bmatrix}, \quad (8)$$

$$W(r) = \begin{bmatrix} rI - K(r) & r \\ -r^T & r \end{bmatrix}, \quad (9)$$

where  $K(r)$  is the anti-symmetric matrix and  $I$  is the identity matrix. To find the rotation quaternion  $r$ , we can set the partial derivatives to zero and perform an eigenvalue decomposition on the matrix, introducing Lagrange multipliers as necessary. The translation component  $t$  can be obtained using the method of singular value decomposition (SVD).

The transformation accomplished by matching triplets one-to-one solely emphasizes error reduction and does not account for situations where there is no matching relationship between the triplets. Hence, it is crucial to implement supplementary filtering and evaluation techniques. In the following section, we will provide comprehensive explanations of the three evaluation strategies that will be employed:

**Map plane projection strategy.** The accurate transformation relationship ensures that planar points in the transformed map exhibit an approximate distance to the planar points in the submap, approaching zero. We will convert  $P_i^M$  to the coordinate system of the submap plane, represented as  $T_{M \rightarrow S} P_i^M$ . By calculating the distance from the transformed map plane coordinates to the submap plane, we can identify the points within a specified range as internal points. Then, we can assess the transformation's effectiveness by determining the proportion of internal points.

$$p(i, j) = \frac{v(T_{M \rightarrow S} P_i^M, P_j^S) \times scale}{T_{M \rightarrow S} P_i^M}, \quad (10)$$

where  $p(i, j)$  represents the fraction of points within a submap categorized as inliers when projected onto the map coordinate system. The function  $v(a, b)$  calculates valid points in  $a$  that lie on the plane represented by  $b$ . The parameter  $scale$  constitutes a fixed threshold contingent on the number and density of the point cloud. In scenarios where the number of point clouds is substantial, the process of projecting points becomes time-consuming. To streamline this process, we employ the projection result of one point to represent all points in its surrounding area.

**Submap plane overlap strategy.** By utilizing the method of overlapping in the provided image, we successfully aligned and edited two images. This alignment requires the images to have overlapping areas [5]. To determine the extent of the overlap, we employ a novel distance representation approach based on the plane's center and the constraints established within the orthogonal plane.



$$o(i, j) = \max \left( \frac{p(T_{M \rightarrow S} P_i^M) - C^S < R^S}{p(T_{M \rightarrow S} P_i^M)}, \frac{p(P_j^S) - T_{M \rightarrow S} C^M < T_{M \rightarrow S} R^M}{p(P_j^S)} \right), \tag{11}$$

where  $o(i, j)$  denotes the overlapping area between a pair of planes  $(i, j)$  within the triplet. The notation  $p(A)$  refers to the ensemble of all points in set  $A$ . The transformation  $T_{M \rightarrow S}$  denotes the conversion from the map coordinate system  $M$  to the submap coordinate system  $S$ . Additionally,  $C$  and  $R$  signify the center of the plane and the radius of the corresponding spatial sphere, respectively.

*Scene identification unique strategy.* In indoor environments, considering only three sets of plane matches can lead to false-positive cases. To address this, we have introduced a scene uniqueness criterion to guide the transformation evaluation function in eliminating triplets that do not meet the requirements. By ensuring that the number of matches between the transformed submap planes and the map planes is not less than the threshold value ( $\tau_m = 5$ ), we can ensure that the obtained transformation is more consistent with the actual observational relationship. The establishment of this threshold is intricately linked to the planar complexity of the indoor scene. When assessing the matching of a triplet, challenges arise in avoiding triplets composed of wall corners in the indoor environment, where determining the position in the room with just one wall corner proves difficult. Consequently, more planes need to be incorporated for matching constraints. Through extensive experimentation, we have determined that achieving more efficient repositioning is feasible with five groups of plane matches. Naturally, this value can be adjusted based on the planar complexity of the indoor scene. The presence of more similar planes or spatial structures corresponds to a larger value for optimal performance.

We use the overlap rate generated by plane transformation as the fundamental measure for computing weights, as defined in Eq. 12. Additionally, we consider the frequency of each plane and plane pair in the triplet. The more representative a plane is, the less frequently it appears, resulting in a higher weight. The weight  $w_{a(i,j)}$  of the plane pair  $(i, j)$  in the triplet can be expressed as:

$$w_{a(i,j)} = \left[ \sum_{b \in \mathcal{T}} \sum_{(k,l) \in b} \prod_{i=k} \exp(-y(a, b)) \right]^{-1} = \left[ \sum_{b \in \mathcal{T}} \sum_{(k,l) \in b} \prod_{i=k} \exp\left(-\left| \log(v_a^{-1} v_b) \right| \right) \right]^{-1}, \tag{12}$$

where function  $\prod_{i=k}$  is a binary operator yielding a value of 1 exclusively when the parameters it operates

on are equal. Additionally,  $y(a, b)$  denotes the dissimilarity between transformations  $a$  and  $b$ . The pair  $(i, j)$  signifies the plane pair within triplet  $a$ , and  $(k, l)$  denotes the plane pair within triplet  $b$ . The variables  $v_a$  and  $v_b$ , respectively, represent the solutions for the transformations in triplets  $a$  and  $b$ , conceptualizing them as points in space. By systematically traversing through the set of triplets  $b = \{b \in \mathcal{T}, b \neq a\}$ , we can compute the comprehensive weight proportion:

$$w_a = \sum_{(i,j) \in a} o(i, j) w_{a(i,j)}^{-1}, \tag{13}$$

where  $o(i, j)$  is the overlap as defined in Eq. 11.

Using the method proposed by Wietrzykowski et al. [35], a probability distribution model is employed to describe the weights of the triplets:

$$P(x) = \frac{1}{Z} \sum_{a \in \mathcal{T}} w_a \exp \left\{ -\log(v_x^{-1} v_a)^T I_a \log(v_x^{-1} v_a) \right\}, \tag{14}$$

where  $Z$  is a constraint that ensures normalization. The distance between the kernel’s center, represented by  $v_x$ , and the transform for triplet  $a$ , represented by  $v_a$ , is computed using logarithm map.  $I_a$  is the information matrix that corresponds to it. Using the probability distribution model mentioned above, we can identify the point with the highest probability, which corresponds to the optimal triplet transformation.

Utilizing the three modules of plane extraction, plane merging, and plane matching, we ultimately obtained an accurate pose transformation relationship, achieving a real-time, lightweight, and robust relocalization system.

## Experiment

We evaluated the proposed VoxelPlane-Reloc system on four different datasets:

- ICL-NUIM dataset [50] is a simulated indoor environment dataset that includes living room and office scenes with a large amount of planar information, which makes it highly suitable for our indoor relocalization system.
- Simulation dataset for plane extraction using noisy data that follows a Gaussian distribution at different depths to obtain ground truth. This dataset is created to evaluate the accuracy of our plane extraction model.
- The mToF dataset is a collection of data captured using mToF cameras in real-world environments, including five common indoor scenes.
- The L515 dataset utilizes the L515 camera, with its acquisition mode being consistent with the mToF dataset.

## Evaluation details

Since our system operates within a global localization framework with pre-existing maps, we employed the adapted VoxelMap [51, 52] algorithm to accurately estimate visual odometry and facilitate the creation of global and local plane maps. The operating system used was Ubuntu Linux 20.04 LTS, along with ROS2 galactic.

We will compare our proposed system with other relocalization systems. Planeloc [36] is a state-of-the-art plane feature-based relocalization system, which utilizes both RGB and depth inputs. To demonstrate the performance of our algorithm, we compared it with the loop closure components of several planar SLAM systems. SP-SLAM [30] is a system that estimates pose using both real and inferred planes and detects relocalization frames using point features. PlanarSLAM [33] is a system that estimates pose using a combination of point, line, and plane features, while also recovering localization using point features. Structure-PLPSLAM [34] builds upon previous work by incorporating line feature constraints for relocalization. To showcase the performance of our repositioning system, we conducted comparative experiments with the aforementioned algorithm. Furthermore, to validate the accuracy of our plane extraction, we compared our proposed plane extraction model with other existing models. NDT-RANSAC [29] is a system that divides point clouds into NDT units for LS plane extraction. PCA-LS [53] is a plane extraction system that combines PCA and LS, but it is worth noting that it does not include weights. PCA [51] and LO-RANSAC [54] are two widely recognized algorithms in the field of plane extraction. Enhanced-VSLAM [32] utilizes region growing combined with RANSAC for plane extraction models. During the design phase of our system, numerous optimization strategies were proposed. To assess the effectiveness of these strategies, we conducted ablation experiments. Among them, VR-NE represents the relocalization system after closing the adaptive weight function in the planar extraction module, VR-NM represents the closure of the plane merging module, VR-NI represents the closure of the incremental input model, VR-NG represents the closure of the ground and orthogonal plane constraints in the triplet selection strategy, VR-NP represents the closure of the map plane projection strategy, VR-NO represents the closure of the submap plane overlap strategy, and VR-NS represents the closure of the scene identification unique strategy. In all experimental tests, the evaluation of relocalization performance included identifying true-positive positions (TP), false-negative positions (FN), true-negative positions (TN), and false-positive positions (FP). Furthermore, precision (P) and recall (R) were evaluated in all experiments. Precision refers to the probability of correctly predicting positive samples among all predicted samples, while recall represents the probability of correctly predicting positive samples

among all actual positive samples. Additionally, to verify the advantages of our system in terms of time consumption, we included a comparison of the average relocalization time (*avg\_time*) in the ablation experiment. The bold font in all tables indicates the optimal data for that experiment.

## Simulation dataset

For the two distinct situations in ICL-NUIM, we utilized the seq-kt1 to create the global map planes. The remaining three sequences were employed to create local submap planes, with each set comprising 50 frames. To assess the distinctiveness of the scenarios, we included 20 negative samples in each scenario test. These negative samples consisted of submap planes that were not part of the specific scenario. Table 1 presents the results of indoor sequence relocalization in publicly available datasets.

Based on the findings presented in Table 1, our method has demonstrated the highest precision and recall rates in indoor scenes. Additionally, Furthermore, Planeloc demonstrates strong performance in screening negative samples. In comparison to the two scenarios, some negative samples are extracted from self-built indoor scenes, and their plane features are relatively rich, presenting significant differences from the test sequences. Additionally, given that the Planeloc system takes RGB and point cloud inputs, localization benefits from the color information in RGB, imposing certain constraints. However, testing revealed that its recall rate in positive samples remains suboptimal, with only half of the localization frames being correctly identified on average. In contrast, our method has shown a 10.24% improvement in precision and a remarkable 58.76% improvement in recall rate, highlighting our method's strong generalization performance in indoor scenes. Although both SP-SLAM and PlanarSLAM incorporate planar features for pose estimation, they exhibit poor relocalization performance for scenes with a close view of the wall due to the use of point features for loop closure pose recovery. While Structure-PLPSLAM enhances performance by incorporating line feature constraints, there is still a noticeable gap when compared to our algorithm. This indicates that using planes for relocalization in indoor scenes can eliminate some undesirable landmarks and achieve more accurate relocalization.

Due to the unavailability of ground truth data for the given dataset, we conducted a simulation experiment to further evaluate the accuracy of the plane extraction model. Specifically, we randomly generated points within a range of 1 m–9 m with a certain depth and ensured that they lie on a given plane. We used this set of plane parameters as the ground truth and added noise along the direction of its normal vector, following a Gaussian distribution. Table 2 presents a comparison between the results of five different plane extraction methods and the ground truth. The value

**Table 1** Relocalization results of ICL-NUIM dataset

Methods	Sequence	Statistics				$P$ [%]	$R$ [%]
		TP	TN	FP	FN		
Ours	Living room	52	19	1	9	<b>98.11</b>	<b>85.24</b>
	Office room	47	20	0	13	<b>100.00</b>	<b>78.33</b>
Paneloc [36]	Living room	29	16	4	32	87.88	47.54
	Office room	34	17	3	26	91.89	56.67
SP-SLAM [30]	Living room	26	6	14	35	65.00	42.63
	Office room	29	8	12	31	71.43	48.33
PlanarSLAM [33]	Living room	29	7	13	32	69.05	47.54
	Office room	37	7	13	23	74.00	61.67
Structure-PLPSLAM [34]	Living room	34	11	9	27	79.07	55.74
	Office room	41	13	7	19	85.42	68.33

**Table 2** Precision for simulation plane extraction

Plane	NDT-RANSAC [29]		PCA-LS [53]		LO-RANSAC [54]		PCA [51]		Enhanced-VSLAM [32]		Ours	
	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]
1	14.259	0.072	15.425	0.104	15.411	0.103	15.604	0.107	9.548	0.307	<b>0.934</b>	<b>0.069</b>
2	7.062	0.176	15.420	0.223	13.802	0.494	15.790	0.221	12.878	0.283	<b>1.113</b>	<b>0.138</b>
3	11.043	0.225	22.006	0.153	11.105	0.160	19.879	0.130	13.128	0.106	<b>1.749</b>	<b>0.040</b>
4	4.074	0.090	3.332	0.203	5.152	0.235	1.608	<b>0.076</b>	13.374	0.191	<b>1.502</b>	0.112
5	3.781	0.145	<b>0.158</b>	<b>0.047</b>	6.677	0.500	4.128	0.083	3.922	0.262	1.059	0.359
6	6.351	0.626	10.761	0.414	11.784	0.349	4.002	0.332	10.684	0.488	<b>2.165</b>	<b>0.068</b>
7	9.889	0.183	14.330	<b>0.083</b>	13.900	0.133	20.372	1.247	13.519	0.181	<b>3.223</b>	0.092
8	4.810	0.146	9.587	0.633	8.325	0.794	12.766	0.736	7.676	0.503	<b>1.576</b>	<b>0.078</b>
9	9.171	0.210	6.505	0.130	13.676	0.259	21.081	0.139	13.469	0.383	<b>0.890</b>	<b>0.034</b>
Mean	7.827	0.208	10.836	0.221	11.092	0.336	12.803	0.341	10.911	0.301	<b>1.579</b>	<b>0.110</b>

$\alpha$  [°] denotes the angle between the normal vector of the extracted plane and the ground truth, while  $d$  [m] represents the distance between the extracted plane and the ground truth plane. In this table, *Mean* represents the average value. Our method outperforms other algorithms in terms of both angle and distance. However, for planes 5 and 7, the concentration of noisy points resulted in excessive iterations, and the plane parameter updates got trapped in local minima. Despite this, our algorithm performs comparably well with the optimal solution for these planes.

The results of the ablation experiments on the ICL-NUIM dataset are detailed in Table 3. Our optimization strategy has yielded substantial improvements in relocalization precision, recall rate, and time consumption. Specifically, VR-NE exhibits a marginal 4% difference in precision compared to our algorithm. This suggests that while our algorithm initially acquires less accurate planar features, the remaining optimization strategies enable the system to identify the correct pose, albeit at the expense of recall rate. VR-NM demonstrates no significant difference from the final algorithm in terms of both precision and recall rate. However, the surplus

of planar features inevitably leads to increased relocalization time. Testing VR-NI and VR-NG confirms that our optimization strategy provides time and computational resource advantages to the relocalization system. The three strategies VR-NP, VR-NO, and VR-NS focused on evaluating the optimal pose transformation, exhibit an average improvement of 17.87% in precision and 48.49% in recall rate. Although there is some time cost due to the addition of evaluation strategies, the overall difference is not significant.

### Real indoor dataset

The experimental setup consists of an mToF depth camera, an L515 camera, and a Turtlebot4 Lite. The mToF is a ToF camera developed by Deprum, which has the advantage of long-distance capability and strong resistance to multi-path interference. It also offers a wide field of view, up to  $100^\circ \times 75^\circ$ , although its precision is relatively lower. The L515 is a solid-state laser radar camera under Intel's brand, known for its high measurement point accuracy and depth information for each pixel. However, it has a relatively

**Table 3** Ablation experiments of ICL-NUIM dataset

Methods	Statistics				<i>P</i> [%]	<i>R</i> [%]	<i>avg_time</i> [s]
	TP	TN	FP	FN			
VR-NE	88	35	5	33	94.62	72.73	2.21
VR-NM	96	32	8	25	92.31	79.34	8.32
VR-NI	93	35	5	28	94.90	76.86	4.70
VR-NG	86	33	7	35	92.47	71.07	14.83
VR-NP	77	30	10	44	88.51	63.64	1.68
VR-NO	69	28	12	52	85.19	57.02	1.23
VR-NS	54	25	15	67	78.26	44.63	<b>1.18</b>
Ours	99	39	1	22	<b>99.00</b>	<b>81.82</b>	1.47

smaller field of view and a higher price. The Turtlebot4 Lite is built on the iRobot Create3 platform and is equipped with wheel encoders. The camera is positioned at the front of the robot, and the robot itself provides wheel odometry information. Sensor data, including data from the mToF depth camera, are transmitted as ROS topics via the ROS2 API over a network connection. We collected an indoor scene dataset in an office environment. The dataset comprises five sequences (seq1–seq5), featuring various office structures, including common office layouts, an activity room, a long corridor, and an open room. The dataset contains a total 124 planar samples. Employing an mToF camera enabled us to capture a more extensive visual range. Consequently, the global map we generated encompasses the unobstructed, condensed ceiling plane, which exists at a specific elevation relative to the ground. This provides valuable additional information for plane relocalization.

### mToF dataset

We use the same experimental evaluation method as the ICL-NUIM dataset. Since the mToF camera captures only point cloud data and does not obtain the original RGB and depth information, Table 4 only shows the comparison results with Planeloc. We partitioned the dataset into two modes: same-sequence testing and cross-sequence testing, to mimic the mapping and relocalization outcomes in real-world situations. We considered minor differences in both time and spatial positions between the relocalized plane map and the globally mapped map. Furthermore, including cross-sequence testing provides a more comprehensive assessment of the algorithm's precision in identifying similar scenes across different sections of the indoor environment.

The results of the tests conducted using both same-sequence and cross-sequence testing methods indicate that the proposed method is highly reliable. It can accurately identify a significant number of locations in real indoor trajectories. To showcase the superior plane relocalization effect

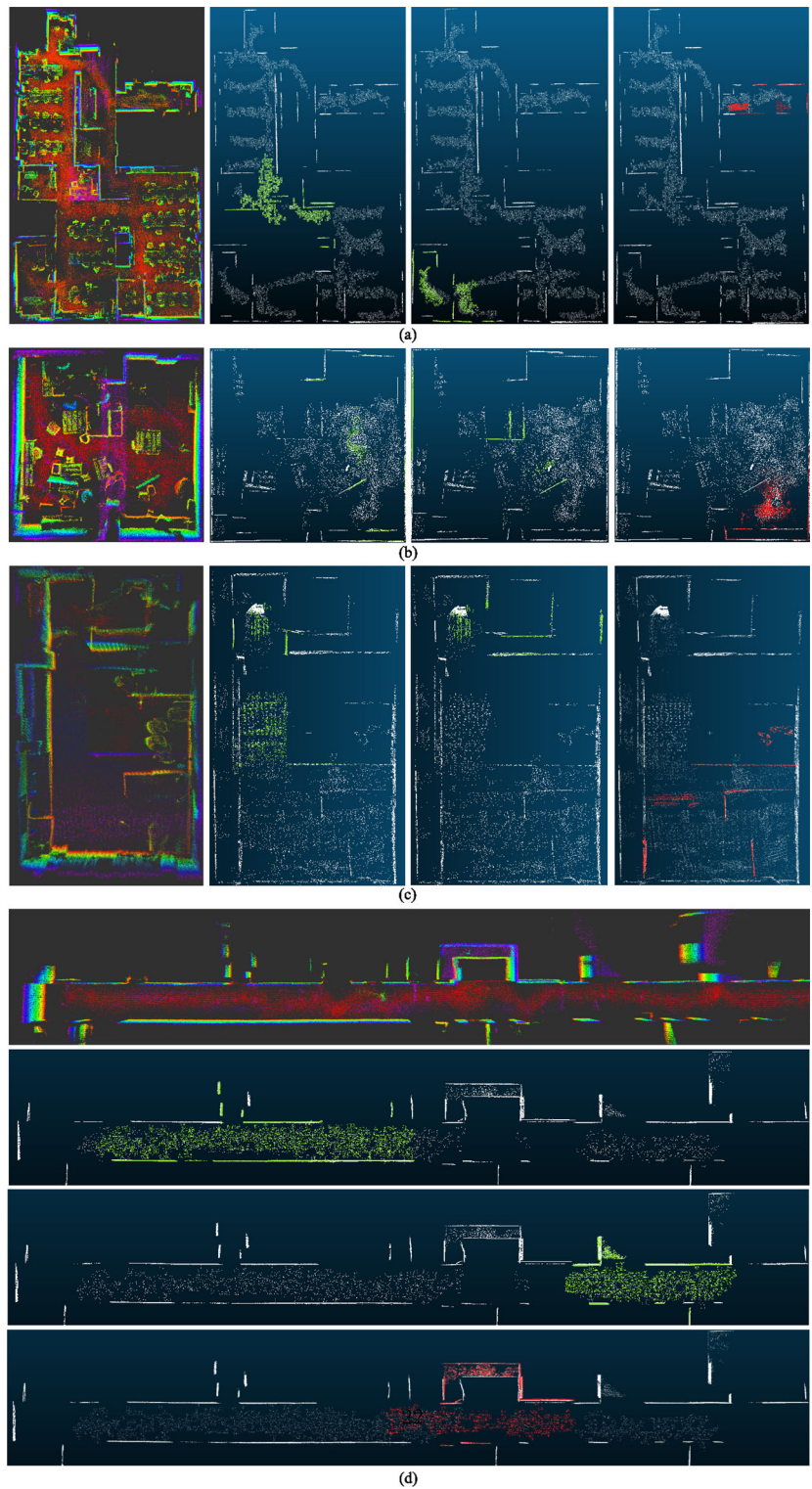
**Table 4** Relocalization results of Indoor dataset built with mToF

Methods	Sequence	Statistics				<i>P</i> [%]	<i>R</i> [%]
		TP	TN	FP	FN		
Ours	Same-seq	49	0	0	5	–	90.7-4
	Cross-seq	17	49	0	1	<b>100.00</b>	<b>94.44</b>
Planeloc [36]	Same-seq	33	0	0	21	–	61.11
	Cross-seq	7	26	23	11	23.33	38.89

of VoxelPlane-Reloc, we divided the indoor plane dataset into subsets and compared them to Planeloc. We differentiated between global maps and local submaps that were constructed from same-sequence and different-sequence data while utilizing the same plane information and metrics. Overall, the Recall of VoxelPlane-Reloc in the same-sequence test reached 90.7%, which represents a 32.1% improvement compared to Planeloc. In the cross-sequence test, we also evaluated Precision by introducing negative samples. It is evident that our algorithm effectively avoids misalignment when negative samples are present. However, there are still situations in which the algorithm proposed in this paper fails to recognize correct loop closures. This is because we intentionally relaxed Recall to achieve higher Precision and avoid incorrect matches. However, this issue can be addressed by increasing the number of local map planes or incorporating richer planar scenes.

The outcomes and comparisons of indoor relocalization, both intra-sequence and inter-sequence, are depicted in Fig. 4. This figure serves as a visual representation of the influence of our pioneering system on the relocalization process. The scenes include: (a) An office scene, spanning 832 square meters and featuring common office elements such as desks, chairs, office supplies, and glass doors. (b) An activity area scene, covering 70 square meters and showcasing sliding doors, display screens, and storage cabinets. (c) An apartment scene, with an area of 56 square meters, displaying common living elements such as sofas, wardrobes, and curtains. (d) A long corridor scene, stretching over 40ms, characterized by a lack of distinct objects and a low-texture environment, posing challenges for localization. In scene (a), the office setting contains multiple relatively similar conference room scenes. The cross-sequence on the left and the same sequence on the right involve the relocalization of the conference room. Despite the scenes' similarity and the absence of obvious tilted plane features, our system consistently provides relatively accurate relocalization results. Scene (b) involves a more complex activity area. Whether the submap scene includes the outline wall feature or not, it consistently achieves excellent relocalization. For example, the same-sequence scene on the right successfully achieved relocalization through the table and the tilted monitor. Scene

**Fig. 4** Some examples of our relocalization results. The first column of (a–c) and the first row of (d) show the mapping effect in the real indoor environment. The second and third columns of (a–c) and the second and third rows of (d) display the effect of relocalization in the same order, where the white point cloud represents the global map, and the green point cloud represents the transformed local submap. The fourth column of (a–c) and the fourth row of (d) show the effect of cross-sequence relocalization, where the white point cloud represents the global map extracted from one sequence, and the red point cloud represents the local submap extracted from another sequence



**Table 5** Precision for mToF Plane Extraction

Plane	NDT-RANSAC [29]		PCA-LS [53]		LO-RANSAC [54]		PCA [51]		Enhanced-VSLAM [32]		Ours	
	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]	$\alpha$ [°]	$d$ [m]
1	<b>0.329</b>	0.040	0.546	0.121	0.922	0.362	0.596	0.141	0.680	0.017	0.376	<b>0.005</b>
2	1.524	0.100	0.602	0.045	0.792	0.098	0.632	0.045	0.387	0.264	<b>0.345</b>	<b>0.004</b>
3	0.446	0.077	0.424	0.073	0.670	0.406	0.425	0.173	0.779	0.174	<b>0.289</b>	<b>0.007</b>
4	0.598	0.012	0.383	0.124	0.843	0.174	1.856	0.129	0.653	0.157	<b>0.160</b>	<b>0.004</b>
5	<b>0.681</b>	0.016	0.867	0.075	0.869	0.196	0.887	0.152	0.585	0.147	0.705	<b>0.007</b>
6	0.292	0.012	0.280	0.133	0.703	0.006	0.485	0.385	0.741	0.075	<b>0.279</b>	<b>0.002</b>
Mean	0.645	0.043	0.517	0.088	0.800	0.207	0.813	0.171	0.637	0.139	<b>0.359</b>	<b>0.005</b>

(c) depicts an apartment setting, showing undulations in the bottom curtain part on the left. However, our relocalization accurately and completely extracts this part of the plane. Additionally, in the cross-sequence on the right, not only is the point cloud forming a plane in the sofa area extracted, but this part also exhibits a remarkable overlap effect. Scene (d) poses a challenge with a long corridor lacking distinctive plane features, presenting a relatively simple and similar scene prone to mismatching. In this low-texture environment, the system proposed in this paper demonstrates a superior relocalization effect, illustrating the versatility of our algorithm for most indoor scenes. However, our algorithm exhibits certain limitations. In scenarios where the plane count within the submap is constrained (specifically limited to five planes), challenges arise when dealing with scenes characterized by significant height symmetry. An example is encountered when attempting to match two diagonals of a meeting room, where the planes are identical, posing difficulties in achieving accurate repositioning based solely on these five planes. In such instances, a pragmatic approach involves expanding the field of view by introducing additional planes capable of discerning nuances between the identical diagonals, thereby facilitating successful repositioning. While this often yields correct pose transformations, it introduces a caveat—the potential emergence of new planes that remain highly similar, such as in the case of a square room with diagonals of equal length. In scenarios where discernible plane features are entirely absent, the existing algorithm encounters challenges in effecting repositioning, unless supplementary sensors (e.g., wheel odometers) are incorporated to provide additional constraints.

Based on the evaluation of the simulation data, we have extracted a subset of planes from the dataset to assess the accuracy of the plane extraction model. Since obtaining the complete real parameters of indoor planes during the measurement process is impractical, we utilized the Fit Plane function in the CloudCompare software to fit the plane's normal vector and calculate the parameter  $d$  as the ground truth. Table 5 provides a comparison of the plane parameters

extracted by our method and other methods, in comparison to the ground truth.

By conducting comparative experiments, it becomes evident that all of these algorithms are capable of accurately extracting plane information. Our approach significantly outperforms other algorithms in terms of extraction accuracy, while also achieving much higher precision in distance calculation. In the cases of planes 1 and 5, NDT-RANSAC exhibits better performance in terms of angle, as it ignores the influence of some larger error outliers during extraction and does not consider all the data. However, it is worth noting that our algorithm still achieves remarkable results despite considering these outliers. There is a certain degree of error in the ground truth calculation using the CloudCompare software, which may affect the clarity of our algorithm's performance in plane extraction accuracy.

The outcomes of ablative experiments conducted on the mToF dataset are elucidated in Table 6. Specifically, VR-NE, VR-NP, VR-NO, and VR-NS substantiate the enhanced precision and recall achieved by the positioning system, with our system attaining 100% precision. On the other hand, VR-NM, VR-NI, and VR-NG primarily showcase a notable reduction in time and computational resource consumption, attributed to the strategies of merging, incremental input, and planar structure constraints. These experiments conclusively demonstrate that the proposed optimization strategies have significantly elevated the positioning system in terms of accuracy, robustness, and real-time performance.

### L515 dataset

To showcase the generalization capability of our relocalization system across diverse camera sensors, we performed ablative experiments on a dataset captured with the L515 camera. The results, depicted in Table 7, underscore the persistent efficacy of our proposed optimization strategy, yielding noteworthy improvements in precision, recall, and timing. This experiment serves as additional evidence of the system's robust generalization across varied sensors and its

**Table 6** Ablation experiments of mToF dataset

Methods	Statistics				<i>P</i> [%]	<i>R</i> [%]	<i>avg_time</i> [s]
	TP	TN	FP	FN			
VR-NE	57	41	8	15	87.69	79.17	1.68
VR-NM	63	35	14	9	81.82	87.50	9.37
VR-NI	62	43	6	10	91.18	86.11	5.02
VR-NG	55	38	11	17	83.33	76.39	14.68
VR-NP	47	34	15	25	75.81	65.28	1.49
VR-NO	43	32	17	29	71.67	59.72	1.38
VR-NS	38	27	22	34	63.33	52.78	<b>1.27</b>
Ours	66	49	0	6	<b>100.00</b>	<b>91.67</b>	1.45

**Table 7** Ablation experiments of L515 dataset

Methods	Statistics				<i>P</i> [%]	<i>R</i> [%]	<i>avg_time</i> [s]
	TP	TN	FP	FN			
VR-NE	52	40	9	17	85.25	75.36	1.74
VR-NM	58	36	13	11	81.69	84.06	9.76
VR-NI	59	42	7	10	89.39	85.51	4.36
VR-NG	54	39	10	15	84.38	78.26	15.10
VR-NP	42	32	17	27	71.19	60.87	1.97
VR-NO	37	28	21	32	63.79	53.62	<b>1.34</b>
VR-NS	34	22	27	35	55.74	49.28	1.86
Ours	63	45	4	6	<b>94.03</b>	<b>91.30</b>	1.52

effectiveness in dense mapping scenarios. However, given that this is not the primary focus of our experiment, we refrain from delving into further details.

## Runtime

In a laptop computer equipped with a Core i5-8250U 3.40GHz processor, the average duration for recognizing one relocalization is 1.48s. This method, described in the article, is more appropriate for real-time systems compared to Planeloc, which takes 19s. The computational time of the four main modules in VoxelPlane-Reloc has been analyzed in Table 8, and the majority of the time is spent on computing pose likelihood, accounting for 78.8% of the total time. The average time for plane relocalization is 157ms, ranging from 44.6ms to 172ms. The number of triplets is the main factor that affects the time, and without any other prior information, it is challenging to impose sufficient constraints on them based solely on structural information.

**Table 8** Average runtime (unit: ms) of different components

Main components	Ours	Planeloc [36]
Single frame	53.763	736
Local observation	101.513	518
Pose likelihood	1165.624	16,439
Localization	157.471	2255
All	<b>1478.371</b>	19,848

## Conclusions

The VoxelPlane-Reloc algorithm proposed in this paper utilizes the plane feature extraction module to improve the performance of relocalization. In the plane feature extraction phase, plane information and an adaptive threshold function are combined to accurately extract planes even in the presence of dynamic targets and noisy environments. The plane merging component enables local voxel plane growth, optimization of plane parameters, reduction in the number of planes, and enhancement of relocalization accuracy. Additionally, by analyzing indoor structures, pre-constraints are applied to global map and submap matching, and a probability density function is constructed using an incremental input model of the submap to determine correct pose relationships. Experiments conducted on the ICL-NUIM dataset and real indoor scenes demonstrate that VoxelPlane-Reloc achieves high precision and recall rates. In the plane feature extraction test, VoxelPlane-Reloc also improves the accuracy of plane information while maintaining high real-time performance for localization.

The VoxelPlane-Reloc algorithm is highly effective, precise, and reliable. It can be utilized for loop detection and scene recognition in indoor environments, specifically for SLAM of service robots. During our experiments, we observed that the algorithm faces difficulties in accurately identifying similar scenes with limited planar features. To address this issue, our next objective is to integrate supplementary features such as plane edges and intersections. This enhancement will broaden the algorithm's applicability to a wider range of scenes.

**Funding** This work was supported by National Natural Science Foundation of China (62301320) and Innovative Research Group Project of the National Natural Science Foundation of China (61373004).

**Data availability** Data is contained within the article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence,

unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P, Milford MJ (2015) Visual place recognition: a survey. *IEEE Trans Robot* 32(1):1–19
- Yin P, Zhao S, Cisneros I, Abuduweili A, Huang G, Milford M, Liu C, Choset H, Scherer S (2022) General place recognition survey: towards the real-world autonomy age. [arXiv:2209.04497](https://arxiv.org/abs/2209.04497)
- Dubé R, Dugas D, Stumm E, Nieto J, Siegart R, Cadena C (2017) Segmatch: segment based place recognition in 3d point clouds. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 5266–5272
- Wietrzykowski J, Skrzypczyński P (2021) On the descriptive power of lidar intensity images for segment-based loop closing in 3-d slam. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 79–85
- Chen X, Läbe T, Milioto A, Röhling T, Behley J, Stachniss C (2022) Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization. *Autonom Robot* 1–21
- Mur-Artal R, Tardós JD (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans Rob* 33(5):1255–1262
- Li Y, Li W, Darwish W, Tang S, Hu Y, Chen W (2020) Improving plane fitting accuracy with rigorous error models of structured light-based rgb-d sensors. *Remote Sens* 12(2):320
- Zaganidis A, Zerntev A, Duckett T, Cielniak G (2019) Semantically assisted loop closure in slam using ndt histograms. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 4562–4568
- Dellenbach P, Deschaut J-E, Jacquet B, Goulette F (2022) Ct-icp: Real-time elastic lidar odometry with loop closure. In: 2022 International Conference on Robotics and Automation (ICRA), IEEE, pp 5580–5586
- Mur-Artal R, Montiel JMM, Tardós JD (2015) Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans Rob* 31(5):1147–1163
- Wietrzykowski J, Skrzypczyński P (2020) A fast and practical method of indoor localization for resource-constrained devices with limited sensing. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 293–299
- Campos C, Elvira R, Rodríguez JGG, Montiel JM, Tardós JD (2021) Orb-slam3: an accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Trans Rob* 37(6):1874–1890
- Qin T, Li P, Shen S (2018) Vins-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans Rob* 34(4):1004–1020
- Sünderhauf N, Protzel P (2011) Brief-gist-closing the loop by simple means. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 1234–1241
- Zhou Z, Zhao C, Adolfsson D, Su S, Gao Y, Duckett T, Sun L (2021) Ndt-transformer: large-scale 3d point cloud localisation using the normal distribution transform representation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 5654–5660
- Wietrzykowski J (2018) Probabilistic reasoning for indoor positioning with sequences of wifi fingerprints. In: 2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), IEEE, pp 338–343
- Gálvez-López D, Tardós JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Trans Rob* 28(5):1188–1197
- Angeli A, Filliat D, Doncieux S, Meyer J-A (2008) Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Trans Rob* 24(5):1027–1037
- García-Fidalgo E, Ortiz A (2014) On the use of binary feature descriptors for loop closure detection. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), IEEE, pp 1–8
- Cummins M, Newman P (2010) Accelerating fab-map with concentration inequalities. *IEEE Trans Rob* 26(6):1042–1050
- Labbe M, Michaud F (2013) Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Trans Rob* 29(3):734–745
- Gao C, Zhang Y, Wang X, Deng Y, Jiang H (2019) Semi-direct rgb-d slam algorithm for dynamic indoor environments. *Robot* 41(3):372–383
- Yang S, Fan G, Bai L, Li R, Li D (2020) Mgc-vslam: a meshing-based and geometric constraint vslam for dynamic indoor environments. *IEEE Access* 8:81007–81021
- Hang C, Zhao B, Wang B (2021) A loop closure detection algorithm based on geometric constraint in dynamic scenes. In: CAAI International Conference on Artificial Intelligence, Springer, New York, pp 516–527
- Oliva A, Torralba A (2001) Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int J Comput Vision* 42:145–175
- Tjaden H, Schwanecke U, Schomer E (2017) Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms. In: Proceedings of the IEEE International Conference on Computer Vision, pp 124–132
- Hou Z, Yan Y, Xu C, Kong H (2022) Hitpr: Hierarchical transformer for place recognition in point cloud. In: 2022 International Conference on Robotics and Automation (ICRA), IEEE, pp 2612–2618
- Sun Q, Yuan J, Zhang X, Duan F (2020) Plane-edge-slam: Seamless fusion of planes and edges for slam in indoor environments. *IEEE Trans Autom Sci Eng* 18(4):2061–2075
- Li L, Yang F, Zhu H, Li D, Li Y, Tang L (2017) An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells. *Remote Sens* 9(5):433
- Zhang X, Wang W, Qi X, Liao Z, Wei R (2019) Point-plane slam using supposed planes for indoor environments. *Sensors* 19(17):3795
- Wietrzykowski J, Belter D (2022) Stereo plane r-cnn: accurate scene geometry reconstruction using planar segments and camera-agnostic representation. *IEEE Robot Autom Lett* 7(2):4345–4352
- Zi B, Wang H, Santos J, Zheng H (2022) An enhanced visual slam supported by the integration of plane features for the indoor environment. In: 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, pp 1–8
- Li Y, Yunus R, Brasch N, Navab N, Tombari F (2021) Rgb-d slam with structural regularities. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 11581–11587
- Shu F, Wang J, Pagani A, Stricker D (2023) Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 2105–2112
- Wietrzykowski J, Skrzypczyński P (2017) A probabilistic framework for global localization with segmented planes. In: 2017 European Conference on Mobile Robots (ECMR), IEEE, pp 1–6



36. Wietrzykowski J, Skrzypczyński P (2019) Planeloc: probabilistic global localization in 3-d using local planar features. *Robot Auton Syst* 113:160–173
37. Wietrzykowski J (2022) Planeloc 2: indoor global localization using planar segments and passive stereo camera. *IEEE Access* 10:67219–67229
38. Zhang J, Cao J-J, Zhu H-R, Yan D-M, Liu X-P (2022) Geometry guided deep surface normal estimation. *Comput Aided Des* 142:103119
39. Zhang L, Lin H, Li C, Song Y, Wang F (2020) Robust estimation approach for plane fitting in 3d laser scanning data. In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, pp 1869–1872
40. Yang L, Li Y, Li X, Meng Z, Luo H (2022) Efficient plane extraction using normal estimation and ransac from 3d point cloud. *Comput Standards Interfaces* 82:103608
41. Sanchez J, Denis F, Coeurjolly D, Dupont F, Trassoudaine L, Checchin P (2020) Robust normal vector estimation in 3d point clouds through iterative principal component analysis. *ISPRS J Photogramm Remote Sens* 163:18–35
42. Guo S, Rong Z, Wang S, Wu Y (2022) A lidar slam with pca-based feature extraction and two-stage matching. *IEEE Trans Instrum Meas* 71:1–11
43. Khanh HD, Do A, Dzung NT *et al.* (2019): An effective randomized hough transform method to extract ground plane from kinect point cloud. In: *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, pp 1053–1058
44. Chai D, Ning Y, Wang S, Sang W, Xing J, Bi J (2022) A robust algorithm for multi-gnss precise positioning and performance analysis in urban environments. *Remote Sens* 14(20):5155
45. Baselga S (2007) Global optimization solution of robust estimation. *J Surv Eng* 133(3):123–128
46. Yang L, Shen Y (2020) Robust m estimation for 3d correlated vector observations based on modified bifactor weight reduction model. *J Geodesy* 94:1–17
47. Yang Y, Song L, Xu T (2002) Robust estimator for correlated observations based on bifactor equivalent weights. *J Geodesy* 76:353–358
48. He Y, Chen S (2020) Error correction of depth images for multiview time-of-flight vision sensors. *Int J Adv Rob Syst* 17(4):1729881420942379
49. Jin Z, Tillo T, Zou W, Zhao Y, Li X (2017) Robust plane detection using depth information from a consumer depth camera. *IEEE Trans Circuits Syst Video Technol* 29(2):447–460
50. Handa A, Whelan T, McDonald JB, Davison AJ (2014) A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China
51. Yuan C, Xu W, Liu X, Hong X, Zhang F (2022) Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry. *IEEE Robot Autom Lett* 7(3):8518–8525
52. Yuan C, Liu X, Hong X, Zhang F (2021) Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robot Autom Lett* 6(4):7517–7524
53. Klasing K, Althoff D, Wollherr D, Buss M (2009) Comparison of surface normal estimation methods for range sensing applications. In: *2009 IEEE International Conference on Robotics and Automation*, IEEE, pp 3206–3211
54. Chum O, Matas J, Kittler J (2003) Locally optimized ransac. In: *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*, Springer, New York, pp 236–243

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.