sensors

MDPI

Article

RGB-D SLAM Using Point–Plane Constraints for Indoor Environments †

Ruibin Guo [1], Keju Peng [1,*], Weihong Fan [1], Yongping Zhai [2] and Yunhui Liu [3]

[1] College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; guoruibin08@nudt.edu.cn (R.G.); whfan@nudt.edu.cn (W.F.)
[2] College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China; ypzhai@foxmail.com
[3] Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong 999077, China; yunhui.liu@gmail.com
* Correspondence: keju009@nudt.edu.cn; Tel.: +86-731-845-03239
† This is an extended version of a conference paper: Plane Based Visual Odometry for Structural and Low-Texture Environments Using RGB-D Sensors. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019.

Received: 29 April 2019; Accepted: 15 June 2019; Published: 17 June 2019

Abstract: Pose estimation and map reconstruction are basic requirements for robotic autonomous behavior. In this paper, we propose a point–plane-based method to simultaneously estimate the robot's poses and reconstruct the current environment's map using RGB-D cameras. First, we detect and track the point and plane features from color and depth images, and reliable constraints are obtained, even for low-texture scenes. Then, we construct cost functions from these features, and we utilize the plane's minimal representation to minimize these functions for pose estimation and local map optimization. Furthermore, we extract the Manhattan World (MW) axes on the basis of the plane normals and vanishing directions of parallel lines for the MW scenes, and we add the MW constraint to the point–plane-based cost functions for more accurate pose estimation. The results of experiments on public RGB-D datasets demonstrate the robustness and accuracy of the proposed algorithm for pose estimation and map reconstruction, and we show its advantages compared with alternative methods.

Keywords: visual SLAM; pose estimation; map reconstruction; point–plane-based factor graph; Manhattan World; RGB-D camera

1. Introduction

This article is an extension of a recent conference paper [1] that presented the exploitation of plane features to estimate sensors' poses for low-texture indoor environments. Robust pose estimation and environment mapping are of great significance in the execution of robotic tasks, such as motion control and navigation. The robot's pose and the scene's map can be obtained by utilizing robotic sensors, such as wheel encoders, inertial measurement units [2–4], lasers [5,6], and cameras [7–9]. Among these solutions, the visual-based method is one of the more effective approaches because cameras can conveniently capture informative images to estimate the robot's poses and perceive its surroundings. Although there have been plenty of methods using monocular, stereo, or RGB-D cameras for pose estimation and 3D mapping, daunting challenges remain for structural and low-texture environments for several reasons. For instance, in existing point-based methods [10,11], key steps in pose estimation, such as image aligning and computing the transformation matrix, heavily rely on feature points or high-contrast pixels. However, feature points are generally absent in structural and low-texture

environments, and these methods can fail to estimate poses or result in low-accuracy estimations. To solve this problem, high-level features, such as lines and planes, are required.

Most indoor environments have many parallel and orthogonal lines and planes (called the Manhattan World [12]), and these high-level features can be exploited to improve the performance of pose estimation. Since these line and plane features can be easily calculated by using RGB-D cameras, which provide both depth information and a color image, the RGB-D camera has become a popular alternative to monocular and stereo cameras for the purpose of simultaneous localization and mapping (SLAM) tasks in indoor environments. These structural regularities have been exploited in studies [13–16] to estimate drift-free rotation, and by decoupling the rotation and translation components, pose estimation accuracy and map quality have been markedly improved by using the Manhattan World (MW) assumption with an RGB-D camera in scenes that satisfy the MW assumption.

In this paper, we propose a robust and accurate approach to pose estimation and 3D mapping using an RGB-D camera. We detected and matched point features from the color images by using the oriented fast and rotated brief (ORB) descriptor, and we detected and tracked multiple planes from the depth images on the basis of a motion model. Then, we exploited these features to construct cost functions to solve the pose of each captured frame and point–plane landmarks in local and global maps. Meanwhile, we added an orientation constraint to the loop detection process to reduce the drift error and avoid mismatches using an appearance constraint. Furthermore, we extracted the MW axes in the first captured frame of MW scenes, and we added the MW constraint to the previous point–plane-based cost functions to improve their performance.

Our algorithm exploits point and plane features and adds the MW constraint for pose estimation and scene reconstruction, which can perform well in harsh environments with low texture as well as general indoor environments. The contributions of this work are as follows:

- We exploited point and plane features, which provide reliable constraints for the estimation of poses and reconstruction of the scene's map for the majority of indoor environments.
- We added the MW constraint to point–plane-based cost functions, resulting in the provision of fixed-plane normals as global landmarks for more accurate pose estimation.
- We evaluated our proposed approach on two public available datasets, and we obtained robust and accurate performance.

2. Related Work

The existing RGB-D SLAM methods for structural and low-texture environments can be divided into three classes: plane-based methods, dense methods, and MW-based methods.

Plane-based methods use plane features to construct and solve the optimization function for pose estimation. Lee et al. [17] presented a fast plane extraction and matching method for indoor mapping, and Taguchi et al. [18] used both points and planes as primitives to realize the registration of different 3D data. Khoshelham et al. [19] proposed a no-iteration pose estimation method based on point–plane correspondences. Thomas et al. [20] presented a structured 3D representation with a point-to-plane relationship to correct the deformations, and local and global mapping were processed to reduce the accumulation error and obtain an accurate large-scale 3D model. Kaess [21] presented a minimal representation for planar features and introduced a relative plane formulation that improved the convergence properties for faster pose optimization. The plane-based methods mentioned above require plane extraction and matching for each frame to construct the optimization function. Since there are no plane descriptors to perform plane matching, it is achieved by utilizing additional odometry sensors, such as wheel encoders or an inertial measurement unit. However, these additional sensors increase the complexity of the SLAM system and may not be available in some circumstances, so plane matching methods that use only the RGB-D frame need to be developed.

In dense methods, almost all pixels are used to estimate the pose. Newcombe et al. [22] presented a frame-to-global method that maintained the single-scene model with a global volumetric so that each new frame would be integrated into the volumetric. Whelan et al. [23] used a rolling cyclical

buffer to operate in large environments and used place recognition for loop closing. Kerl et al. [24] proposed a dense visual SLAM method for RGB-D cameras that minimized both the photometric and depth error of all pixels, and Prisacariu et al. [25] presented a robust dense RGB-D SLAM method (InfiniTAM) that had low computational cost with RGB and depth constraints. These dense methods solve the pose with a dense vision front-end and are robust in low-texture environments. However, the number of points processed for each frame is large (typically hundreds of thousands), which makes the optimization computationally infeasible in real time without GPU implementation.

MW-based methods estimate the pose by decoupling the rotation and translation components. These methods utilize line and plane features to achieve a drift-free rotation matrix with the MW constraint, and the translational accuracy can be improved by using drift-free rotation. Zhou et al. [26] developed a mean shift paradigm to extract and track planar modes to achieve drift-free rotation, and they estimated the translation using three simple 1D density alignments in man-made environments. In the work of Kim et al. [27], lines and planes were exploited to estimate drift-free rotation, and the translation was recovered by minimizing the de-rotated reprojection error. Kim et al. [28] also proposed a linear SLAM method based on the Bayesian filtering framework for MW scenes. These methods have produced good SLAM performance results in MW scenes, but if the MW assumption is invalid, MW-based methods fail to estimate the pose or reconstruct the map.

3. Proposed Method

We propose a point–plane-based RGB-D SLAM system that exploits point and plane features to estimate the camera pose and generate the 3D global map for indoor environments. Our proposed system has two main parts: (1) we detect and track the point and plane features with respect to the local map for each new captured frame, and we estimate the current pose by solving the cost function that is constituted by the tracked features (tracking part); and (2) we update the local map that consists of point–plane landmarks and keyframes for each new inserted keyframe, and we process the full bundle adjustment to obtain the global map if a loop is detected (map management part). If the current environment satisfies the MW assumption, we add the MW direction to constrain the normal of plane landmarks for more accurate pose estimation. The overview of our proposed system is shown in Figure 1.

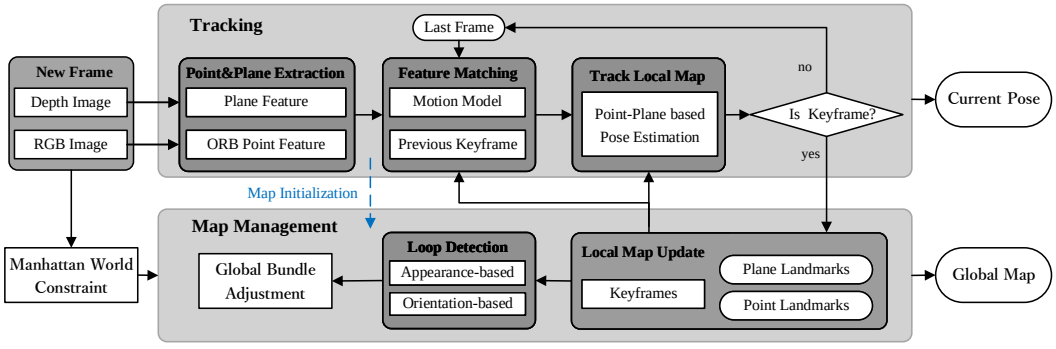


Figure 1. Overview of our point–plane-based RGB-D SLAM system. The inputs of the system are the RGB and depth images, and it can output the camera pose and scene's map. We estimate the camera poses by using the tracked point and plane features. We exploit the point and plane constraints to update the local and global maps. The MW constraint is added to the map management part if the global MW axes are extracted from the first captured RGB-D frame.

3.1. Preliminaries

In this section, we first introduce the representations for the point and plane features that are extracted from the color and depth images, respectively. Then, we detail the state transformation

and distance measurement, which are essential for constructing the cost functions and solving the nonlinear graph optimization problem for pose estimation.

3.1.1. Point and Plane Representation

We extract ORB features for point tracking, as these features are computed extremely quickly, and they present good invariance to the camera's auto-gain, auto-exposure, and changes in illumination. The point feature's 2D pixel coordinate in the color image domain is defined as $\mathbf{u^c} = (u^\mathbf{c}, v^\mathbf{c})^T$, where $\mathbf{c}$ represents the current processing frame, and u and v represent the coordinate values on the x-axis and y-axis, respectively. For the aligned color and depth images, the point feature's 2D coordinates are the same in the depth image and color image, so the corresponding value of $\mathbf{u^c}$ in the depth image is represented by $d(\mathbf{u^c})$. The point feature's 3D position $\mathbf{X^c}$ is reconstructed by using the inverse projection function $\mathbf{P}^{-1}(\cdot)$:

$$\mathbf{X^c} = \mathbf{P}^{-1}(\mathbf{u^c}, d(\mathbf{u^c})) = d(\mathbf{u^c})(\frac{u^\mathbf{c} - c_x}{f_x}, \frac{v^\mathbf{c} - c_y}{f_y}, 1)^T \in \mathbb{R}^3 \tag{1}$$

where f_x and f_y are the focal lengths on the x-axis and y-axis, and $(c_x, c_y)^T$ is the camera's central coordinate.

We detect the planes from the depth images using a fast plane extraction algorithm [29], which has three steps: generate initial blocks, merge the blocks on the basis of agglomerative clustering, and refine the border pixels. With this approach, we can obtain the plane: $\mathbf{n}^T \cdot \mathbf{X^p} = d$, where $\mathbf{n} = (n_x, n_y, n_z)^T$ represents the unit normal vector of the plane, $\mathbf{X^p}$ represents the 3D point lying on this plane, and d is the distance to the origin of the camera coordinate system. The plane can also be represented by a homogeneous vector, $\boldsymbol{\pi} = (\pi_x, \pi_y, \pi_z, \pi_w)^T$:

$$\boldsymbol{\pi} = \mathbf{Q}(\mathbf{n}, d) = \frac{1}{\sqrt{n_x^2 + n_y^2 + n_z^2 + d^2}} \begin{bmatrix} \mathbf{n} \\ -d \end{bmatrix} \in \mathcal{S}^3 \tag{2}$$

where $\mathbf{Q}(\cdot)$ is the normalized transfer function for a 4-dimensional vector, $\mathcal{S}^3$ represents the unit sphere, which can be identified with a set of unit quaternions, so the operations on the quaternions are suited to the plane's homogeneous representation [21].

3.1.2. State Transformation

The 3D point $\mathbf{X^c}$ in the current frame is transformed to $\mathbf{X^w} = \mathbf{R}_{w,c} \cdot \mathbf{X^c} + \mathbf{t}_{w,c}$ in the global coordinate, where $\mathbf{R}_{w,c} \in SO(3)$ is the rotation matrix and $\mathbf{t}_{w,c} \in \mathbb{R}^3$ is the translation vector. When we use the rigid-body transformation matrix that defined as $\mathbf{T}_{w,c} = \begin{bmatrix} \mathbf{R}_{w,c} & \mathbf{t}_{w,c} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$, the transformation relationship is expressed as $\begin{bmatrix} \mathbf{X^w} \\ 1 \end{bmatrix} = \mathbf{T}_{w,c} \cdot \begin{bmatrix} \mathbf{X^c} \\ 1 \end{bmatrix}$.

The plane $\boldsymbol{\pi^c}$ in the current coordinate is transformed to the global coordinate by $\boldsymbol{\pi^w} = \mathbf{Q}(\mathbf{T}_{w,c}^{-T} \cdot \boldsymbol{\pi^c})$, where $\mathbf{T}_{w,c}^{-T} = \begin{bmatrix} \mathbf{R}_{w,c} & \mathbf{0}_{3\times1} \\ -\mathbf{t}_{w,c}^T \cdot \mathbf{R}_{w,c} & 1 \end{bmatrix}$. In terms of the plane's normal-distance representation, its state transformation is represented by $\begin{bmatrix} \mathbf{n^w} \\ -d^\mathbf{w} \end{bmatrix} = \mathbf{T}_{w,c}^{-T} \cdot \begin{bmatrix} \mathbf{n^c} \\ -d^\mathbf{c} \end{bmatrix}$.

3.1.3. Distance Measurement

In the graph-based pose estimation problem, the error function is constructed from an edge that connects multiple nodes [30]. As the binary edge only connects two nodes, the error function measures the distance between these two nodes. For two 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$, we use the 2-norm function $\|\cdot\|_2$ to define their relative distance: $\mathbf{e^X} = \|\mathbf{X}_1 - \mathbf{X}_2\|_2$.

For two planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, their relative distance is defined as $\mathbf{e}^{\boldsymbol{\pi}} = \| \log(\boldsymbol{\pi}_1^{-1} \cdot \boldsymbol{\pi}_2) \|_2$, where $\boldsymbol{\pi}^{-1} = (-\pi_x, -\pi_y, -\pi_z, \pi_w)^T$, and we use the quaternion multiplication to operate the planes' multiplication:

$$\boldsymbol{\pi}_1^{-1} \cdot \boldsymbol{\pi}_2 = \begin{bmatrix} -\pi_{1x}\pi_{2w} - \pi_{1y}\pi_{2z} + \pi_{1z}\pi_{2y} + \pi_{1w}\pi_{2x} \\ \pi_{1x}\pi_{2z} - \pi_{1y}\pi_{2w} - \pi_{1z}\pi_{2x} + \pi_{1w}\pi_{2y} \\ -\pi_{1x}\pi_{2y} + \pi_{1y}\pi_{2x} - \pi_{1z}\pi_{2w} + \pi_{1w}\pi_{2z} \\ \pi_{1x}\pi_{2x} + \pi_{1y}\pi_{2y} + \pi_{1z}\pi_{2z} + \pi_{1w}\pi_{2w} \end{bmatrix} \tag{3}$$

The function $log(\cdot)$ maps an element of $\mathcal{S}^3$ to the 3D rotation vector:

$$log(\boldsymbol{\pi}) = \frac{2\cos^{-1}(\pi_w)}{\sqrt{\pi_x^2 + \pi_y^2 + \pi_z^2}} \begin{bmatrix} \pi_x \\ \pi_y \\ \pi_z \end{bmatrix} \tag{4}$$

3.2. Pose Estimation with Point and Plane Features

We match the points and planes detected in the current frame with the point and plane landmarks in the local map, and we utilize the tracked point and plane features to construct a cost function for estimating the current pose. By combining the point and plane constraints, we can accurately estimate the pose in scenes, even those with less texture.

3.2.1. Point and Plane Feature Tracking

As mentioned in Section 3.1.1, we extract the ORB point features and plane features from the current frame $F^{\mathbf{c}}$. For the point features, we get the initial point matches between the current frame and the last reference keyframe by using the ORB descriptors. Then, we project the corresponding map points onto the current frame and discard some outlier matches on the basis of the projection error. The set of optimized point matches is defined as $\mathcal{X} = \{(\mathbf{X}_i^{\mathbf{c}}, \mathbf{X}_i^{\mathbf{L}}), i = 1, 2, ...m\}$, where $\mathbf{X}^{\mathbf{c}}$ represents the point feature's 3D position in the current frame coordinate, and $\mathbf{X}^{\mathbf{L}}$ represents the 3D position of the point landmark in the local map.

Since there are no plane descriptors to perform plane matching, we search for plane matches by the motion-model-based distance constraint. If the previous two frames $F^{\mathbf{c}-2}$ and $F^{\mathbf{c}-1}$ were tracked successfully, we use the constant velocity motion model [10] to predict the current pose: $\mathbf{T}_{w,c}^{predict} = \mathbf{T}_{c-1,c-2} \cdot \mathbf{T}_{w,c-1}$, where $\mathbf{T}_{c-1,c-2}$ is the relative pose from $F^{\mathbf{c}-2}$ to $F^{\mathbf{c}-1}$, and $\mathbf{T}_{w,c-1}$ represents the estimated pose for $F^{\mathbf{c}-1}$. One detected plane $\begin{bmatrix} \mathbf{n}^{\mathbf{c}} \\ -d^{\mathbf{c}} \end{bmatrix}$ is transformed to $\begin{bmatrix} \mathbf{n}_{predict}^{\mathbf{c}} \\ -d_{predict}^{\mathbf{c}} \end{bmatrix} = \mathbf{T}_{w,c}^{predict} \cdot \begin{bmatrix} \mathbf{n}^{\mathbf{c}} \\ -d^{\mathbf{c}} \end{bmatrix}$, and we obtain the plane matches when they meet:

$$\begin{cases} \| (\mathbf{n}_{predict}^{\mathbf{c}})^T \cdot \mathbf{n}^{\mathbf{L}} \|_2 > 0.95 \\ |d_{predict}^{\mathbf{c}} - d^{\mathbf{L}}| < 0.05 \end{cases} \tag{5}$$

where $(\mathbf{n}^{\mathbf{L}}, d^{\mathbf{L}})$ is the normal-distance representation of the plane landmark in the local map. The set of plane matches is defined as $\mathcal{P} = \{(\boldsymbol{\pi}_i^{\mathbf{c}}, \boldsymbol{\pi}_i^{\mathbf{L}}), i = 1, 2, ...n\}$, where $\boldsymbol{\pi}^{\mathbf{c}}$ represents the plane's homogeneous representation extracted from the current frame, and $\boldsymbol{\pi}^{\mathbf{L}}$ represents the plane landmark in the local map.

To avoid incorrect plane matches in cluttered environments, we only select extracted planes that have enough points (more than 5000) lying on them to match with the plane landmarks in the local map. For parallel planes that satisfy the previous condition, we select the plane with the largest number of on-plane points. In this way, the number of incorrect matches can be effectively reduced.

3.2.2. Robust Pose Estimation

We jointly utilize the tracked points and planes to construct a cost function for the current pose estimation. Pose $\mathbf{T}_{w,c}$ can be computed by solving

$$\{\mathbf{R}_{w,c}, \mathbf{t}_{w,c}\} = \underset{\mathbf{R}_{w,c}, \mathbf{t}_{w,c}}{\arg\min} \Big(\sum_{i \in \mathcal{X}} \rho(\|\mathbf{R}_{w,c} \cdot \mathbf{X}_i^{\mathbf{c}} + \mathbf{t}_{w,c} - \mathbf{X}_i^{\mathbf{L}}\|_2^2) + \sum_{j \in \mathcal{P}} \lambda_j \|\log(\mathbf{Q}(\mathbf{T}_{w,c}^{-T} \cdot \boldsymbol{\pi}_j^{\mathbf{c}})^{-1} \cdot \boldsymbol{\pi}_j^{\mathbf{L}})\|_2^2 \Big) \tag{6}$$

where $\rho(\cdot)$ is the robust Huber cost function, λ_j represents the number of pixels in the jth tracked plane, and 3D point $\mathbf{X}_i^{\mathbf{c}} = \mathbf{P}^{-1}(\mathbf{u}_i^{\mathbf{c}}, d(\mathbf{u}_i^{\mathbf{c}}))$.

The cost function (6) contains two parts that correspond to point and plane constraints. The accurate pose can be solved by minimizing Equation (6), even in a texture-less environment in which few points tracked. The point constraints ensure that the pose estimation is reliable even in scenes in which there are not enough planes to be visible.

Keyframe Selection: By the previous process, we always know the number of tracked points and planes for each frame. If there is only one tracked plane but the number of detected planes is larger than 2 or the number of the tracked points is less than a threshold, this frame is selected as a keyframe. By inserting the keyframes and updating the local map, the drift error of the pose estimation can be markedly reduced.

3.3. Map Management and Loop Detection

In this section, we describe the operation for the local map when a new keyframe is inserted, and we detect the closing loop on the basis of both the appearance and orientation constraints. If loop detection is successful, the full bundle adjustment is performed to generate the final global map.

Similar to the co-visibility graph and essential graph in ORB-SLAM [10], we denote the set of co-visible keyframes by $\mathcal{K}^{\mathbf{L}}$; all points seen in $\mathcal{K}^{\mathbf{L}}$ are represented by $\mathbb{S}^{\mathbf{L1}}$, and all planes seen in $\mathcal{K}^{\mathbf{L}}$ are represented by $\mathbb{S}^{\mathbf{L2}}$. All other keyframes $\mathcal{K}^{\mathbf{F}_1}$ that are not in $\mathcal{K}^{\mathbf{L}}$, as well as the observation points in $\mathbb{S}^{\mathbf{L1}}$, contribute to the cost function but remain fixed in the optimization. All other keyframes $\mathcal{K}^{\mathbf{F}_2}$ that are not in $\mathcal{K}^{\mathbf{L}}$, as well as the observation planes in $\mathbb{S}^{\mathbf{L2}}$, contribute to the cost function and also remain fixed in the optimization. We define the set of point matches as $\mathcal{X}^{\mathbf{L_m}}$ between the points in $\mathbb{S}^{\mathbf{L1}}$ and keypoints in keyframe m, and we define the set of plane matches as $\mathcal{P}^{\mathbf{L_n}}$ between the planes in $\mathbb{S}^{\mathbf{L2}}$ and the keyplanes in keyframe n. The local map is updated by solving

$$\begin{aligned} \{\mathbf{X}_i^{\mathbf{L}}, \boldsymbol{\pi}_j^{\mathbf{L}}, \mathbf{R}_{w,l}^{\mathbf{k}}, \mathbf{t}_{w,l}^{\mathbf{k}} | i \in \mathbb{S}^{\mathbf{L1}}, j \in \mathbb{S}^{\mathbf{L2}}, l \in \mathcal{K}^{\mathbf{L}}\} = & \underset{\mathbf{X}_i^{\mathbf{L}}, \boldsymbol{\pi}_j^{\mathbf{L}}, \mathbf{R}_{w,l}^{\mathbf{k}}, \mathbf{t}_{w,l}^{\mathbf{k}}}{\arg\min} \Big(\sum_{m \in \mathcal{K}^{\mathbf{L} \cup \mathbf{F}_1}} \sum_{p \in \mathcal{X}^{\mathbf{L_m}}} \rho(E_{mp}^{\mathbf{X_L}}) + \sum_{n \in \mathcal{K}^{\mathbf{L} \cup \mathbf{F}_2}} \sum_{q \in \mathcal{P}^{\mathbf{L_n}}} \lambda_q \cdot E_{nq}^{\boldsymbol{\pi}_{\mathbf{L}}} \Big) \\ E_{mp}^{\mathbf{X_L}} = & \|\mathbf{R}_{w,m}^{\mathbf{k}} \cdot \mathbf{X}_p^{\mathbf{m}} + \mathbf{t}_{w,m}^{\mathbf{k}} - \mathbf{X}_p^{\mathbf{L}}\|_2^2 \\ E_{nq}^{\boldsymbol{\pi}_{\mathbf{L}}} = & \|\log(\mathbf{Q}(\mathbf{T}_{w,n}^{-T} \cdot \boldsymbol{\pi}_q^{\mathbf{n}})^{-1} \cdot \boldsymbol{\pi}_q^{\mathbf{L}})\|_2^2 \end{aligned} \tag{7}$$

where λ_q represents the number of pixels contained in the plane.

3.3.1. Local Map Update

The local map contains three elements: the keyframe, point landmark, and plane landmark. These elements are represented by nodes in a factor graph, which is shown in Figure 2. There are two kinds of binary edges in this point–plane-based factor graph: one connects the keyframe node and point landmark node, and the other one connects the keyframe node and plane landmark node. When a new keyframe is inserted, the poses of all elements in the local map are optimized by the local bundle adjustment (BA).

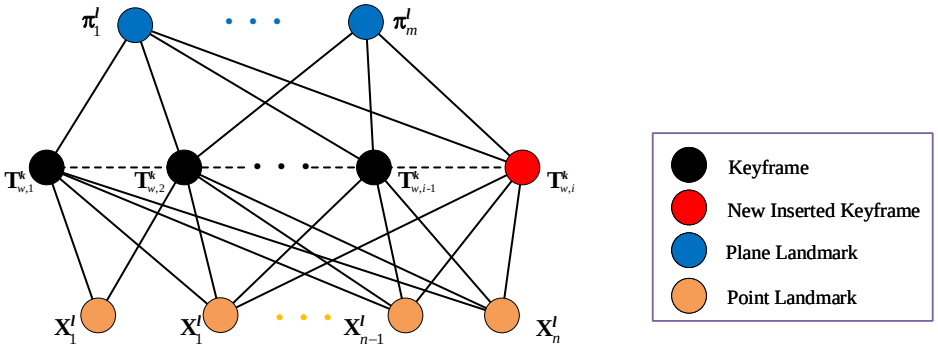


Figure 2. Point–plane-based factor graph that is used to represent the local map. Four colors denote four different elements. When a new keyframe (red circle) is inserted, the pose of the keyframes (black circles), point landmarks (blue circles), and plane landmarks (yellow circles) are optimized by minimizing the error cost function constructed by the factor graph.

3.3.2. Loop Detection based on Appearance and Orientation Constraints

In the point-based SLAM system, loop detection is performed by using a bag-of-words (BoW) place recognition module with DBoW [31]. This visual vocabulary is an appearance constraint for loop detection, and it is created by pretraining with a large set of pictures. Since places that appear similar in indoor environments are familiar, as shown in Figure 3, we add an orientation constraint to complement the appearance constraint in determining the loop keyframes.

(**a**) (**b**)

Figure 3. Two mismatched loop images based on the appearance constraint: (**a**) Color image of Frame 182 in the 'OfficeRoom1' sequence of the ICL-NUIM dataset. (**b**) Color image of Frame 319 in the 'OfficeRoom1' sequence. These two images are viewed from two completely different perspectives, but their similar appearance score is high because the ceiling is a common feature in the environment.

In our proposed system, if the similarity score (using the visual vocabulary) between a new inserted keyframe K_i and an existing keyframe $K_p^{\mathbf{L}}$ is higher than a threshold, we select this keyframe as the potential loop keyframe, and we compute their orientation distance in degrees by Equation (8). We confirm loop detection if the potential loop keyframe meets $d_{p,i}^{\mathbf{O}} < 90deg$. By adding this orientation constraint, the mismatch (based on the appearance constraint) of these two loop images is revised.

$$d_{p,i}^{\mathbf{O}} = \arccos\left(\frac{tr(\mathbf{R}_{w,p}^T \cdot \mathbf{R}_{w,i}) - 1}{2}\right) \times 57.3 \tag{8}$$

where $tr(\cdot)$ denotes the trace of a matrix, and $\mathbf{R}_{w,i}$ and $\mathbf{R}_{w,p}$ represent the rotation matrix component for keyframe K_i and $K_p^{\mathbf{L}}$, respectively.

3.3.3. Global Map Generation

If a loop is detected in the previous step, all poses of the elements are optimized by the full bundle adjustment, and the first keyframe is fixed in the process of full BA. We represent the set of all keyframes by $\mathcal{K}^{\mathbf{G}}$: all point landmarks are represented by $\mathbb{S}^{\mathbf{G1}}$, and all plane landmarks are represented by $\mathbb{S}^{\mathbf{G2}}$. We define the set of point matches as $\mathcal{X}^{\mathbf{G_m}}$ between points in $\mathbb{S}^{\mathbf{G1}}$ and keypoints in keyframe m, and we define the set of plane matches as $\mathcal{P}^{\mathbf{G_n}}$ between planes in $\mathbb{S}^{\mathbf{G2}}$ and planes in keyframe n. The global map is optimized by solving

$$\begin{aligned} \{\mathbf{X}_i^{\mathbf{G}}, \boldsymbol{\pi}_j^{\mathbf{G}}, \mathbf{R}_{w,l}^{\mathbf{k}}, \mathbf{t}_{w,l}^{\mathbf{k}} | i \in \mathbb{S}^{\mathbf{G1}}, j \in \mathbb{S}^{\mathbf{G2}}, l \in \mathcal{K}^{\mathbf{G}}\} = & \underset{\mathbf{X}_i^{\mathbf{G}}, \boldsymbol{\pi}_j^{\mathbf{G}}, \mathbf{R}_{w,l}^{\mathbf{k}}, \mathbf{t}_{w,l}^{\mathbf{k}}}{\arg\min} \Big(\sum_{m \in \mathcal{K}^{\mathbf{G}}} \sum_{p \in \mathcal{X}^{\mathbf{G_m}}} \rho(E_{mp}^{\mathbf{X_G}}) + \sum_{n \in \mathcal{K}^{\mathbf{G}}} \sum_{q \in \mathcal{P}^{\mathbf{G_n}}} \lambda_q \cdot E_{nq}^{\boldsymbol{\pi}_{\mathbf{G}}} \Big) \\ E_{mp}^{\mathbf{X_G}} = & \| \mathbf{R}_{w,m}^{\mathbf{k}} \cdot \mathbf{X}_p^{\mathbf{m}} + \mathbf{t}_{w,m}^{\mathbf{k}} - \mathbf{X}_p^{\mathbf{G}} \|_2^2 \\ E_{nq}^{\boldsymbol{\pi}_{\mathbf{G}}} = & \| \log(\mathbf{Q}(\mathbf{T}_{w,n}^{-T} \cdot \boldsymbol{\pi}_q^{\mathbf{n}})^{-1} \cdot \boldsymbol{\pi}_q^{\mathbf{G}}) \|_2^2 \end{aligned} \tag{9}$$

where λ_q represents the number of pixels in the plane.

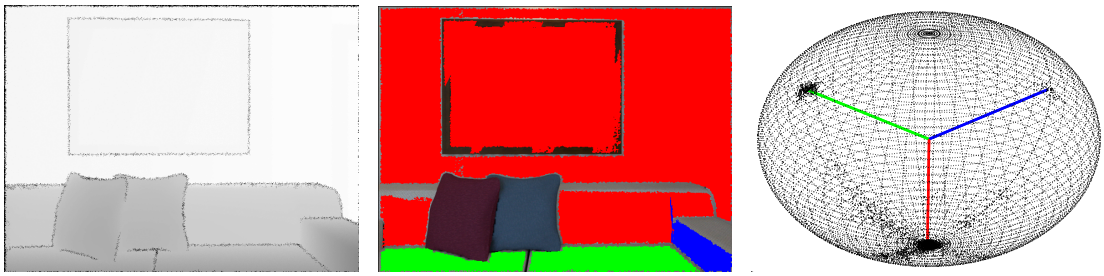
After the full BA, the global map with point and plane landmarks is generated, which can be applied to robotic localization, navigation, and path planning.

3.4. Pose and Plane Optimization with the MW Constraint

For the environment that satisfies the MW assumption, we exploit the parallel and orthogonal lines and planes to extract the MW axes, and we add the MW constraint to construct the cost function to optimize the poses of keyframes and landmarks.

3.4.1. MW Axes Extraction

We extract the MW axes from the first frame by utilizing the plane normal vectors and the parallel lines' vanishing directions (VDs), the details of which are given in our previous work [32]. To extract the accurate plane normals, we use the normal vectors obtained by the previous fast plane extraction method as the initial value and then perform the mean shift algorithm in the tangent plane of the unit sphere to get the final plane normal vectors, as shown in Figure 4. As the normals of parallel planes are regularly distributed and more likely to be around the ground MW axes on the unit sphere, the final extracted results are obtained by utilizing all normals of parallel planes, which are more accurate than the initial plane normals.



(**a**) Depth image (**b**) Plane primitives (**c**) Extracted Plane Normals

Figure 4. Result of the plane normal extraction: (**a**) depth image of Frame 0 in the 'Living Room 1' sequence of the ICL-NUIM dataset; (**b**) detected planes using fast plane extraction method; (**c**) extracted plane normals on the unit sphere. The plane primitives in the image domain and their corresponding extracted normal vectors are in the same color.

The geometric relationship between the VDs and parallel lines is shown in Figure 5. To extract accurate VDs ($\mathbf{d}_k^v, k = 1, 2, 3$), we use the simplified Expectation–Maximization (EM) clustering method to group image lines and compute their corresponding 3D direction vectors. We use the linear-time Line Segment Detector (LSD) [33] to extract 2D line segments from the color image and roughly cluster

the lines using the K-means method. Then, we compute the VDs by solving the weighted objective function for each $\mathbf{d}_k^v$:

$$\mathbf{d}_k^v = argmin \sum_i \frac{length(\mathbf{l}_i^{(k)})}{max(length(\mathbf{l}^{(k)}))} \cdot (\mathbf{l}_i^{(k)\mathrm{T}} \mathbf{K} \mathbf{d}_k^v)^2 \tag{10}$$

where $\mathbf{l}^{(k)}$ represents the ith line cluster obtained by the K-means method, $length(\mathbf{l}_i^{(k)})$ represents the length of the ith line, $max(length(\mathbf{l}^{(k)}))$ represents the maximum line length in cluster k, and $\mathbf{K}$ represents the internal camera parameters.

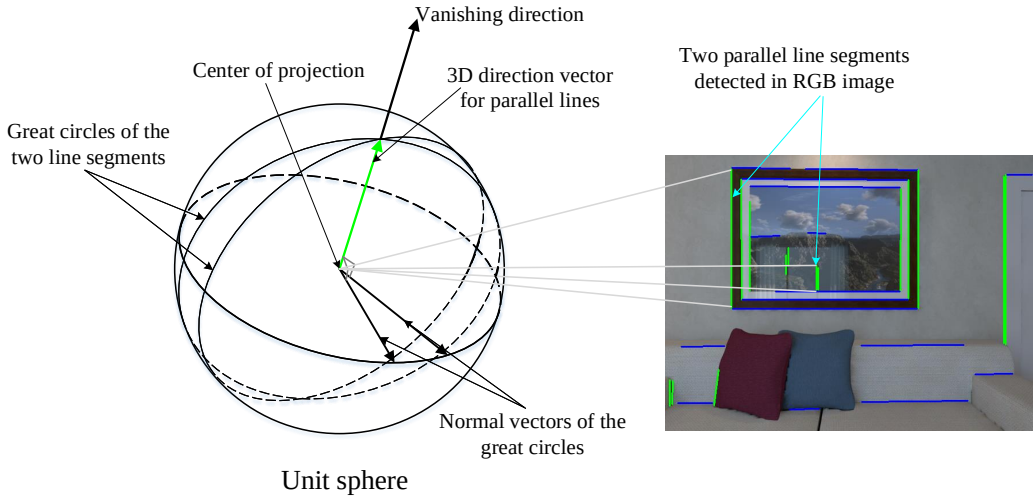


Figure 5. Three-dimensional geometric relationship between parallel lines and their vanishing direction. The unit sphere is in the center of a camera projection. Two parallel lines are projected onto the unit sphere as two great circles, and the vanishing direction is obtained by the cross-projection of these two great circles' normal vectors. Two parallel lines and their corresponding vanishing direction are drawn in red.

MW Axes Seeking: In the scenes that satisfy the MW assumption, there are three fixed axes $(\mathbf{r}_1^g \ \mathbf{r}_2^g \ \mathbf{r}_3^g)$. It should be noted that we treat $\mathbf{r}^g$ and $-\mathbf{r}^g$ as the same direction. To determine the MW axes, we first get a redundant set by using the plane normals and VDs obtained using the previous method. Then, we seek the plane that contains the most pixels and set its plane normal as the first MW axis $\mathbf{r}_1$. The other two MW axes $\mathbf{r}_2$ and $\mathbf{r}_3$ are sought on the basis of two principles: the number of pixels belonging to the plane or line and the orthogonal constraint. The larger the number of pixels, the higher the priority of the plane normal or VD. The final global MW axes are obtained by using singular value decomposition (SVD):

$$\left[\mathbf{r}_1^g \ \mathbf{r}_2^g \ \mathbf{r}_3^g\right] = \mathbf{U}\mathbf{V}^T \tag{11}$$

where $[\mathbf{U}, \mathbf{D}, \mathbf{V}] = SVD([\lambda_1 \mathbf{r}_1 \ \lambda_2 \mathbf{r}_2 \ \lambda_3 \mathbf{r}_3])$, and factor λ_i represents the number of pixels belonging to a plane or line.

3.4.2. Optimization with Fixed Plane Normal

In the previous section, we present the extraction of the global MW axes $(\mathbf{r}_1^g \ \mathbf{r}_2^g \ \mathbf{r}_3^g)$, which are used to fix the normals of plane landmarks during optimization. We add the MW constraint to construct the cost functions (6), (7) and (9), in which the plane landmarks are represented by

$$\boldsymbol{\pi}_i^{\mathbf{L}_{fixed}} = \mathbf{Q}(\mathbf{r}_i^g, d_i) = \frac{1}{\sqrt{r_{ix}^2 + r_{iy}^2 + r_{iz}^2 + d_i^2}} \begin{bmatrix} \mathbf{r}_i^g \\ -d_i \end{bmatrix} \in \mathcal{S}^3 \tag{12}$$

For MW environments, this MW constraint can effectively improve the accuracy of the SLAM system because adding the MW axes to the cost function is equivalent to setting three global directions in the optimization for poses and landmarks, so the drift can be reduced. If the MW axes are not detected in the current environment, we simply use the point–plane-based SLAM without the MW constraint to perform the localization and mapping tasks.

4. Results

We evaluated our proposed approach on a synthetic dataset (ICL-NUIM [34]) and a real-world dataset (TUM RGB-D [35]). All experiments were run on a desktop computer with an Intel Core i7, 16 GB memory, and Ubuntu 16.04 platform. Our proposed system was built on ORB-SLAM2 [10], and our system is executed in the same manner as ORB-SLAM2.

- The ICL-NUIMdataset comprises images from a hand-held RGB-D camera in synthetically generated environments. These sequences were captured in a living room and an office with perfect ground-truth poses to fully quantify the accuracy of a given visual odometry or SLAM system. Depth and RGB noise models were used to alter the ground images to simulate realistic sensor noise. Some image sequences are in low-texture environments, which makes it difficult to estimate the poses of the whole images in these sequences.
- The TUM RGB-D dataset is a famous benchmark that is used to evaluate the accuracy of a given visual odometry or visual SLAM system. It contains various indoor sequences captured by a Kinect RGB-D sensor. The sequences were recorded in real environments at a frame rate of 30 Hz with a 640 × 480 resolution, and their ground-truth trajectories were obtained from a high-accuracy motion-capture system. The TUM dataset is more challenging than the ICL dataset because includes some blurred images and inaccurate alignment image pairs that make it difficult to estimate the camera poses.

We compared our proposed approach with five methods: ORB-SLAM2 [10], DVO [24], InfiniTAM [25], LPVO [27], and L-SLAM [28]. ORB-SLAM2 is a state-of-the-art point-based SLAM system; DVO estimates the robust poses with photometric and depth error by using the color and depth images together; InfiniTAM estimates the camera poses from the RGB and depth images with a GPU in real time; LPVO exploits the line and plane to estimate the zero-drift rotation and then estimates the 3D poses with tracked points in the MW scenes; L-SLAM estimates the camera position and plane landmarks with a linear SLAM formulation in the MW environments. We use the root-mean-square error (RMSE) of the absolute translational error (ATE) as the performance metric for the entire sequences:

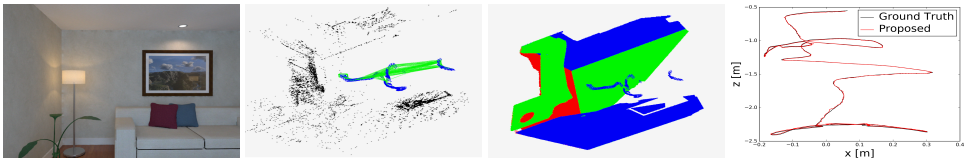
$$\mathbf{ATE}.RMSE = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{R}_{g,p} \cdot \mathbf{X}_i^p + \mathbf{t}_{g,p} - \mathbf{X}_i^g\|_2 \tag{13}$$

where $\mathbf{R}_{g,p}$ and $\mathbf{t}_{g,p}$ represent the rotation matrix and translational matrix that transform the trajectory coordinate obtained by our proposed method to the ground-truth coordinate; three-dimensional points $\mathbf{X}_i^p$ and $\mathbf{X}_i^g$ denote the traces of the proposed method and ground truth, respectively; and N represents the number of frames in the tested sequence.

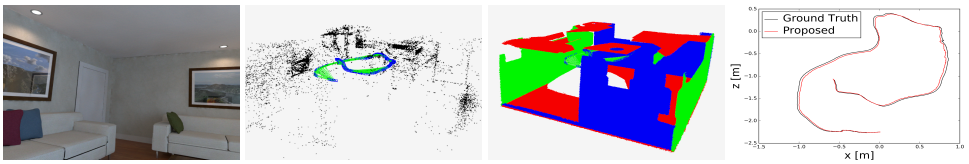
4.1. Evaluation on Synthetic Dataset

We first evaluated our proposed method on the ICL-NUIM dataset. The estimated trajectories and point–plane landmarks are shown in Figure 6, and the measured RMSE values of the ATE for each sequence are shown in Table 1. The smallest values are bolded and indicate the most accurate result for the pose estimation. For example, in 'Living Room 0', the ATE.RMSE value of our proposed method is 0.006 m, while those of ORB-SLAM2, DVO, LPVO, and L-SLAM are 0.010, 0.108, 0.015, and 0.012 m, respectively. The 'Living Room 1' sequence includes images that are mostly composed of a texture-less wall, so the accuracy of the point-based ORB-SLAM2 method is poor. As the DVO method does not have an efficient loop-closing process, the drift error cannot be avoided, and its ATE.RMSE is large.

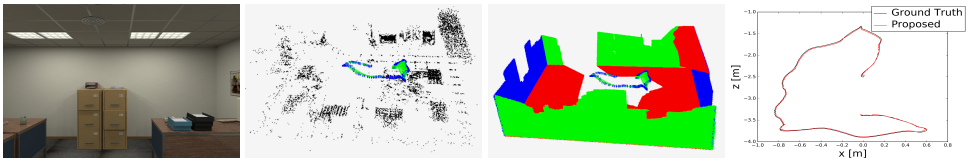
InfiniTAM fails to estimate the whole frames' poses in three sequences ("Living Room 0", "Living Room 3", and "Office Room 2") because there are some frames with only one visible plane in the depth image and low texture in the color image. We marked the result as '×'. Although the LPVO method can provide drift-free rotation, it estimates the 3D pose using only the tracked points; thus, if there are not enough points, the accuracy decreases. L-SLAM is a linear SLAM method that uses the MW constraint and does not need to estimate the 3 degrees of freedom rotation. L-SLAM performs well with the MW scenes, and our method is comparable to it. The last column in Table 1 shows the number of frames in the current sequence.



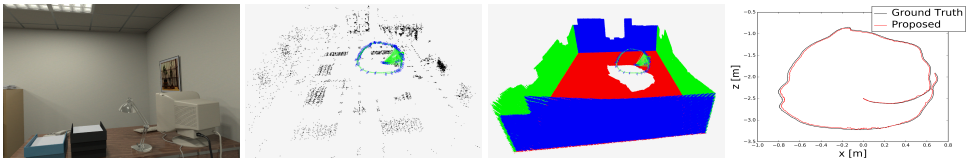
(**a**) LivingRoom0



(**b**) LivingRoom2



(**c**) OfficeRoom1



(**d**) OfficeRoom3

Figure 6. Results of camera poses and landmarks estimated by our proposed method on the ICL-NUIM dataset: (**a**) 'Living Room 0'; (**b**) 'Living Room 2'; (**c**) 'Office Room 1'; (**d**) 'Office Room 3'. For each sequence, the four images from left to right represent, respectively, one color image in the current sequence, the point landmarks (black dots) obtained by our proposed method, the plane landmarks obtained by our proposed method, and the trajectory comparison between the ground truth and our proposed method. The estimated keyframe trace (blue boxes) and connection graph between them (green lines) were added to the middle images that show the point and plane landmarks.

The MW assumption is sufficiently suitable for the ICL-NUIM benchmark. To clearly show the effect of the MW constraint, we measured the ATE.RMSE for all sequences obtained by our method without the MW constraint; this corresponds to the 'No MW' column in Table 1. We recorded the values of the absolute translational error (ARE) for each frame in the 'Living Room 0' sequence, and the ATE values with and without the MW constraint are shown in Figure 7: in this figure, the smaller the ATE value, the more accurate the pose estimation. This demonstrates that the MW constraint can improve the accuracy of pose estimation for MW environments.

Table 1. Comparison of ATE.RMSE (unit: m) on the ICL-NUIM dataset. The smallest values are bolded, which indicates the most accurate method for the pose estimation.

Sequence	Proposed	No MW	ORB-SLAM2	DVO	InfiniTAM	LPVO	L-SLAM	Frames
Living Room 0	**0.006**	0.007	0.010	0.108	×	0.015	0.012	1508
Living Room 1	0.010	0.011	0.185	0.059	**0.006**	0.039	0.027	965
Living Room 2	0.026	0.027	0.028	0.375	**0.013**	0.034	0.053	880
Living Room 3	**0.013**	0.016	0.014	0.433	×	0.102	0.143	1240
Office Room 0	**0.019**	0.025	0.049	0.244	0.042	0.061	0.020	1507
Office Room 1	0.016	0.017	0.079	0.178	0.025	0.052	**0.015**	965
Office Room 2	**0.017**	0.019	0.025	0.099	×	0.039	0.026	880
Office Room 3	0.016	0.018	0.065	0.079	**0.010**	0.030	0.011	1240

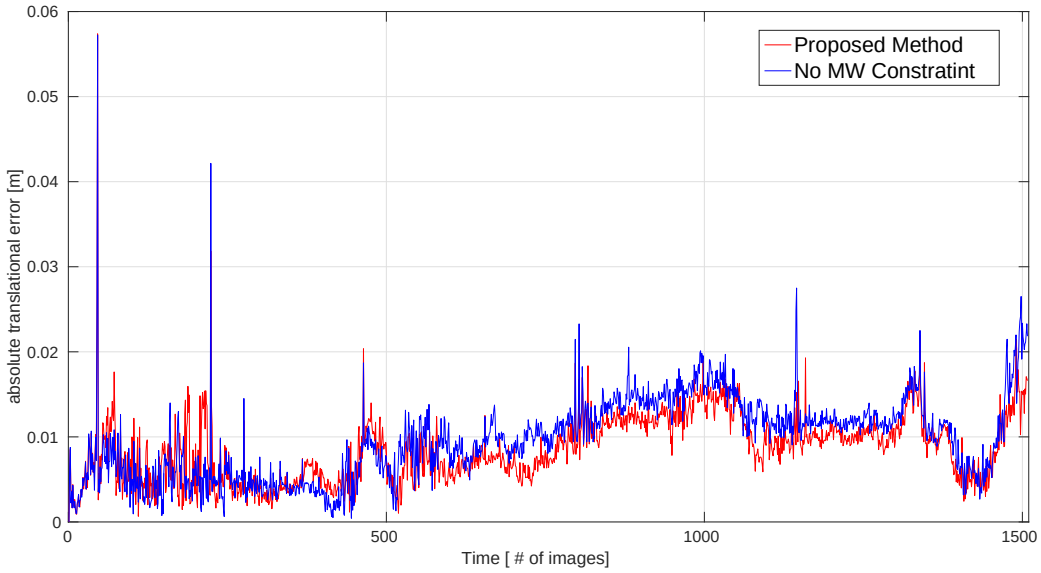
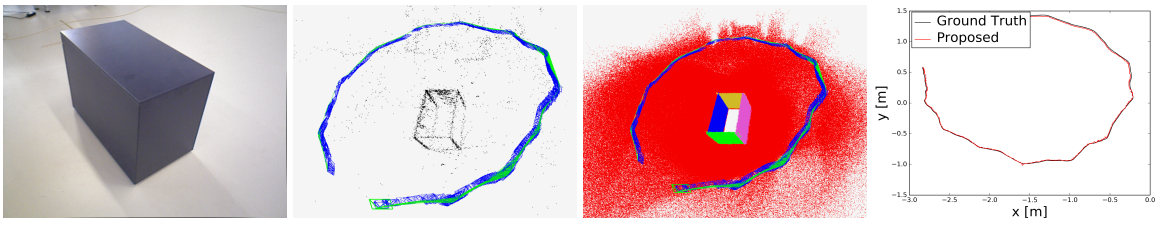


Figure 7. Performance evaluation for the MW constraint on the 'Living Room 0' sequence. Absolute translational errors for our proposed method with and without the MW constraint are compared.

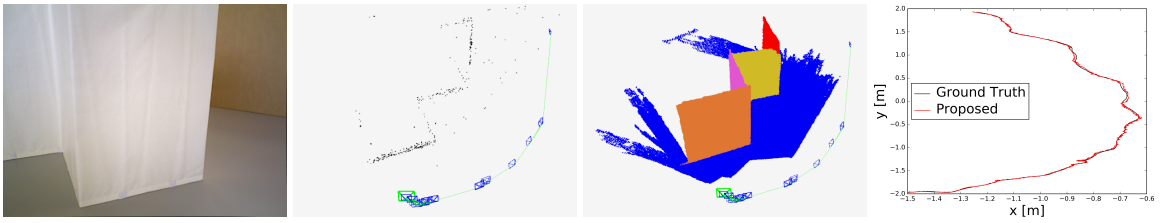
4.2. Evaluation on Real-World Data

We then evaluated our proposed method on the TUM RGB-D dataset. The trajectories generated by the poses of the whole captured frames and the point–plane landmarks in the map are shown in Figure 8. We compared the performance of our proposed algorithm with that of the other five methods on six real-world TUM RGB-D sequences that contain structural regularities. The comparison results are shown in Table 2. We provide the ATE.RMSE for 3D pose estimation, and the smallest values are indicated in bold. Our proposed method performs better in low-texture environments because it uses the point and plane features to estimate poses. In 'fr3_cabinet', the ORB-SLAM2 method failed to estimate the poses for the entire sequence because there are not enough reliable tracked points for some frames; we marked the result as '×' in Table 2. The last column in Table 2 also represents the number of frames in the current TUM RGB-D sequence.

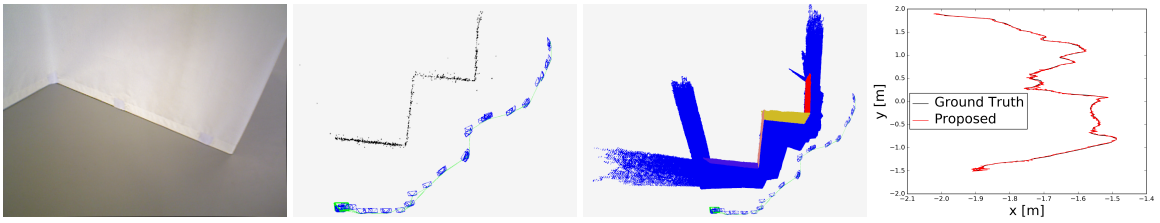
The performance results of the MW constraint on the 'fr3_struc_notex_far' sequence is shown in Figure 9. The final translational drift obtained by our proposed method with and without the MW constraint is 0.031 and 0.072, respectively. It is clear that the MW constraint can effectively reduce the drift error for the MW scenes.



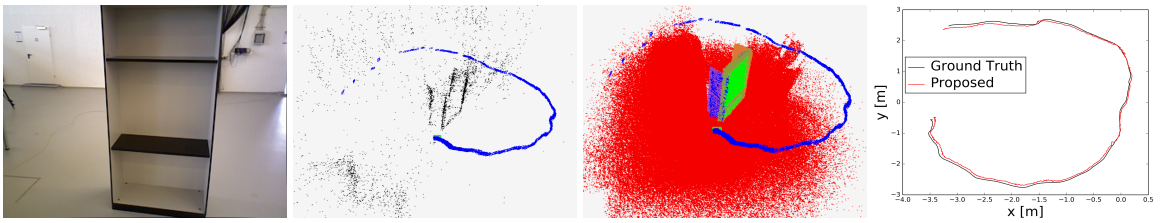
(**a**) fr3_cabinet



(**b**) fr3_struc_notex_far



(**c**) fr3_struc_notex_near



(**d**) fr3_large_cabinet

Figure 8. Results of camera poses and landmarks estimated by our proposed method on the TUM RGB-D dataset: (**a**) 'fr3_cabinet'; (**b**) 'fr3_struc_notex_far'; (**c**) 'fr3_struc_notex_near'; (**d**) 'fr3_large_cabinet'. For each sequence, four images from left to right represent, respectively, one color image in the current sequence, the point landmarks (black dots) obtained by our proposed method, the plane landmarks obtained by our proposed method, and the trajectory comparison between the ground truth and our proposed method. The estimated keyframe trace (blue boxes) and connection graph between them (green lines) were added to the middle images that show point and plane landmarks.

Table 2. Comparison of ATE.RMSE (unit: m) on the TUM RGB-D Dataset. The smallest values are bolded, which indicates the most accurate method for the pose estimation.

Sequence	Proposed	No MW	ORB-SLAM2	DVO	InfiniTAM	LPVO	L-SLAM	Frames
fr3_struc_notex_far	**0.017**	0.029	0.276	0.213	0.037	0.075	0.141	790
fr3_struc_tex_far	**0.011**	0.012	0.024	0.048	0.030	0.174	0.212	904
fr3_struc_notex_near	**0.008**	0.009	0.652	0.076	0.022	0.080	0.066	1031
fr3_struc_tex_near	**0.011**	0.013	0.019	0.031	0.034	0.115	0.156	1054
fr3_cabinet	**0.012**	0.013	×	0.690	0.035	0.520	0.291	926
fr3_large_cabinet	**0.074**	0.094	0.179	0.979	0.512	0.279	0.140	979

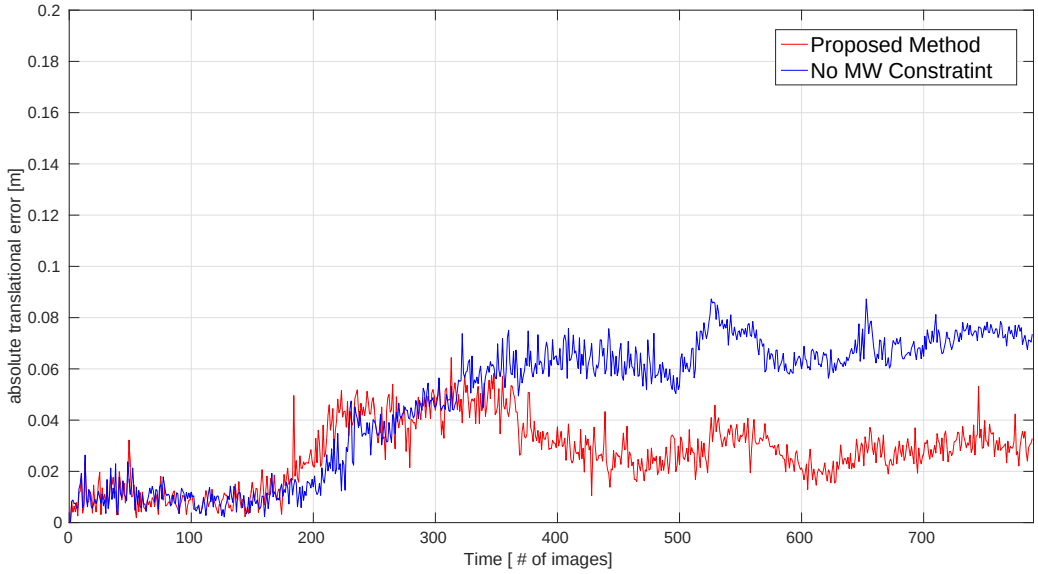


Figure 9. Performance evaluation for the MW constraint on the 'fr3_struc_notex_far' sequence. Absolute translational errors for our proposed method with and without the MW constraint are compared.

5. Conclusions

We proposed a point–plane-based method to estimate robot poses and reconstruct the maps of scenes of indoor environments using an RGB-D camera. We exploited point and plane features to generate reliable constraints, which we applied to the constructed cost function for solving the transformation matrix, and we used minimal representation for planes in the nonlinear optimization process. We developed a vanishing direction extraction method based on parallel lines and combined it with the detected plane normals to seek the MW axes in the current environment. Then, we added the MW constraint to further improve accuracy for MW environments. The proposed algorithm was tested on both synthetic and real-world publicly available RGB-D datasets, and we compared the pose estimation performance of our method with that of five existing methods. The results demonstrate the accuracy and robustness of the proposed method. Our approach can be used for a robot's tasks in indoor environments. In future work, we will extend our approach to the point–line–plane feature fusion SLAM system, which may provide robust pose estimation in more general environments and generate structural maps.

Author Contributions: Methodology, R.G.; Resources, R.G.; Software, R.G.; Supervision, K.P. and W.F.; Validation, K.P. and Y.L.; Writing—original draft, R.G.; Writing—review & editing, Y.Z. and Y.L.

Funding: This work was supported in part by the Natural Science Foundation of China under Grant U1613218.

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

References

1. Guo, R.; Zhou, D.; Peng, K.; Liu, Y. Plane Based Visual Odometry for Structural and Low-Texture Environments Using RGB-D Sensors. In Proceedings of the BigComp 2019, 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019.
2. Wang, Z.; Chen, Y.; Mei, Y.; Yang, K.; Cai, B. IMU-Assisted 2D SLAM Method for Low-Texture and Dynamic Environments. *Appl. Sci.* **2018**, *8*, 2534. [CrossRef]
3. Fu, Q.; Li, S.; Liu, Y.; Zhou, Q.; Wu, F. Automatic Estimation of Dynamic Lever Arms for a Position and Orientation System. *Sensors* **2018**, *18*, 4230. [CrossRef] [PubMed]
4. Guo, R.; Zhou, D.; Peng, K.; Fan, W.; Liu, Y. Improved real-time odometry estimation method for incremental RGB-D mapping by fusing IMU data. In Proceedings of the 2016 IEEE 12th World Congress on Intelligent Control and Automation (WCICA), Gui-lin, China, 12–15 June 2016; pp. 2991–2995.

5. Jiang, L.; Zhao, P.; Dong, W.; Li, J.; Ai, M.; Wu, X.; Hu, Q. An Eight-Direction Scanning Detection Algorithm for the Mapping Robot Pathfinding in Unknown Indoor Environment. *Sensors* **2018**, *18*, 4254. [CrossRef] [PubMed]
6. Zhang, X.; Rad, A.B.; Wong, Y.K. Sensor fusion of monocular cameras and laser rangefinders for line-based simultaneous localization and mapping (SLAM) tasks in autonomous mobile robots. *Sensors* **2012**, *12*, 429–452. [CrossRef]
7. Payá, L.; Reinoso, O.; Jiménez, L.M.; Juliá, M. Estimating the position and orientation of a mobile robot with respect to a trajectory using omnidirectional imaging and global appearance. *PLoS ONE* **2017**, *12*, e0175938. [CrossRef] [PubMed]
8. Yoon, S.J.; Kim, T. Development of Stereo Visual Odometry Based on Photogrammetric Feature Optimization. *Remote Sens.* **2019**, *11*, 67. [CrossRef]
9. Li, J.; Gao, W.; Li, H.; Tang, F.; Wu, Y. Robust and Efficient CPU-Based RGB-D Scene Reconstruction. *Sensors* **2018**, *18*, 3652. [CrossRef] [PubMed]
10. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
11. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
12. Coughlan, J.M.; Yuille, A.L. Manhattan world: Compass direction from a single image by bayesian inference. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 941–947.
13. Joo, K.; Oh, T.H.; Kim, J.; So Kweon, I. Globally optimal Manhattan frame estimation in real-time. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1763–1771.
14. Straub, J.; Bhandari, N.; Leonard, J.J.; Fisher, J.W. Real-time manhattan world rotation estimation in 3d. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1913–1920.
15. Zhou, Y.; Kneip, L.; Li, H. Real-time rotation estimation for dense depth sensors in piece-wise planar environments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2271–2278.
16. Straub, J.; Freifeld, O.; Rosman, G.; Leonard, J.J.; Fisher, J.W. The manhattan frame model—Manhattan world inference in the space of surface normals. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 235–249. [CrossRef] [PubMed]
17. Lee, T.K.; Lim, S.; Lee, S.; An, S.; Oh, S.Y. Indoor mapping using planes extracted from noisy RGB-D sensors. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 1727–1733.
18. Taguchi, Y.; Jian, Y.D.; Ramalingam, S.; Feng, C. Point-plane SLAM for hand-held 3D sensors. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5182–5189.
19. Khoshelham, K. Direct 6-DoF pose estimation from point-plane correspondences. In Proceedings of the 2015 IEEE International Conference on Digital Image Computing, Techniques and Applications (DICTA), Adelaide, SA, Australia, 23–25 November 2015; pp. 1–6.
20. Thomas, D.; Sugimoto, A. Modeling large-scale indoor scenes with rigid fragments using RGB-D cameras. *Comput. Vis. Image Underst.* **2017**, *157*, 103–116. [CrossRef]
21. Kaess, M. Simultaneous localization and mapping with infinite planes. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 25–30 May 2015; Volume 1, p. 2.
22. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Basel, Switzerland, 26–29 October 2011; pp. 127–136.
23. Whelan, T.; Kaess, M.; Johannsson, H.; Fallon, M.; Leonard, J.J.; McDonald, J. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *Int. J. Robot. Res.* **2015**, *34*, 598–626. [CrossRef]

24. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 2100–2106.
25. Prisacariu, V.A.; Kähler, O.; Golodetz, S.; Sapienza, M.; Cavallari, T.; Torr, P.H.; Murray, D.W. InfiniTAM v3: A framework for large-scale 3D reconstruction with loop closure. *arXiv* **2017**, arXiv:1708.00783.
26. Zhou, Y.; Kneip, L.; Rodriguez, C.; Li, H. Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds. In *Asian Conference on Computer Vision*; Springer: Berlin, Germany, 2016; pp. 3–19.
27. Kim, P.; Coltin, B.; Kim, H.J. Low-drift visual odometry in structured environments by decoupling rotational and translational motion. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7247–7253.
28. Kim, P.; Coltin, B.; Jin Kim, H. Linear RGB-D SLAM for planar environments. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 333–348.
29. Feng, C.; Taguchi, Y.; Kamat, V.R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–5 June 2014; pp. 6218–6225.
30. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g^2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
31. Lau, J.H.; Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv* **2016**, arXiv:1607.05368.
32. Guo, R.; Peng, K.; Zhou, D.; Liu, Y. Robust visual compass using hybrid features for indoor environments. *Electronics* **2019**, *8*, 220. [CrossRef]
33. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A line segment detector. *Image Process. Line* **2012**, *2*, 35–55. [CrossRef]
34. Handa, A.; Whelan, T.; Mcdonald, J.; Davison, A.J. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–5 June 2014; pp. 1524–1531.
35. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.