IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Tracking by Detection: Robust Indoor RGB-D Odometry Leveraging Key Local Manhattan World

Zhiyu Zhou[1,2], Zhi Gao[1,2] and Jingzhong Xu[1]

***Abstract*—In indoor scenes, pose estimation drift in SLAM systems is challenging to mitigate due to factors like low texture and texture repetition. Several studies utilize Manhattan structural information to achieve low-drift rotation estimation consequently reducing the cumulative error. However, the application scenarios of these methods are limited by the Manhattan assumption. To address this problem, we propose a robust RGBD odometry for tracking in more general structured scenes, which can represent a wider range of scenes by leveraging Atlanta World assumption, consisting of a vertical direction and multiple horizontal directions. To this end, we design tracking by detection algorithms for extracting Key Local Manhattan World in each frame, which is defined as the most stable structural information in current frame. Specifically, we detect local Manhattan Worlds in current frame, if a more stable structural feature emerges, it becomes the new tracking target. The angle between these Manhattan Worlds are then computed, constructing the Atlanta World. For accurate pose estimation, we use Key Local Manhattan World to first estimate low-drift rotation followed by estimating translation using point-line structural features. Extensive experiments on public benchmark and self-recorded datasets show that our method outperforms existing state-of-the-art methods with a significant margin of 22% while extending their applicability to the Atlanta World.**

***Index Terms*—SLAM, Localization, RGB-D Perception.**

## I. Introduction

In recent years, RGB-D sensors, such as Microsoft Kinect, Intel RealSense, and LiDAR systems, have become more accessible, leading to the widespread adoption of RGB-D odometry. These sensors capture depth information alongside RGB color images, facilitating the estimation of robot pose with higher precision [1], [2]. Accordingly, RGB-D-based odometry plays a key role in pose estimation for robots in various scenarios. By leveraging both RGB and depth information, RGB-D odometry enhances robot tracking robustness in complex environments, and enables various applications such as SLAM, autonomous driving, and augmented reality. Thus, RGB-D odometry has become a research hotspot.

Manuscript received: December, 22, 2023; Revised March, 2, 2024; Accepted March, 30, 2024.

This paper was recommended for publication by Editor Asfour Tamim upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Hubei Province Natural Science Foundation (Grant No. 2021CFA088), the Science and Technology Major Project (Grant No. 2021AAA010, 2021AAA010-3) and Zhongguan Co., Ltd., Wuhan, China, Cooperation Project (*Corresponding author: Zhi Gao*).

[1]Zhiyu Zhou, Zhi Gao and Jingzhong Xu are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: zhouzhiyuwhu@whu.edu.cn; gaozhinus@gmail.com; jz_xu@whu.edu.cn).

[2]Zhiyu Zhou and Zhi Gao are also with the Hubei Luojia Laboratory, Wuhan 430079, China.

Digital Object Identifier (DOI): see top of this page.



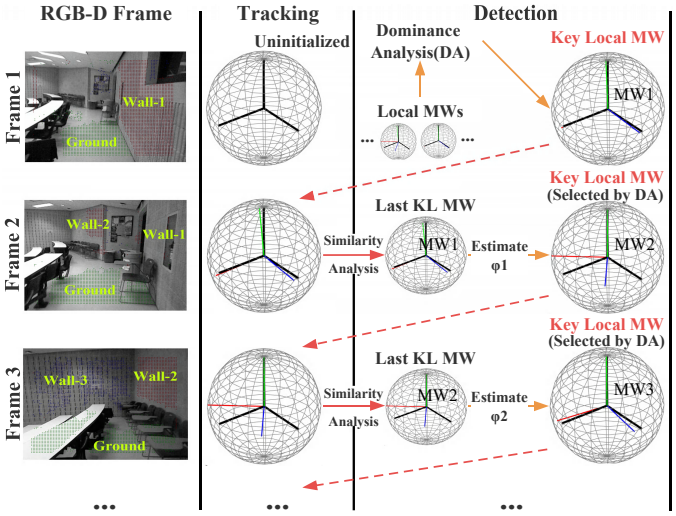


Fig. 1. Detection and tracking of Key Local MW on TAMU dataset [8]. Plane detection results are labeled on RGB-D frame (Wall-1, Wall-2, Wall-3 and Ground form MW1, MW2 and MW3, respectively). Sphere shows local MW (black line represents camera coordinate system, colored line represents three direction of MW). In detection section, we detect multiple local MWs and select Key Local MW according to dominance analysis (called DA), red dotted line represents replacement of tracking target; after initialization, we continue to select Key Local MW according to DA, and connect MWs by calculating horizontal angle. (Zoom in for details)

However, due to the presence of a large number of low-texture and repeatable-texture regions in man-made indoor scenes, traditional visual odometry methods based on point feature [3], [4] struggle to achieve satisfactory performance in such scenes. To tackle this challenge, a common alternative is to leverage additional high-level structural features (lines and planes) in indoor structured scenarios [5]–[7]. The utilization of RGB-D sensors adds stability to the extraction of structural features [8], [9]. Nonetheless, these techniques are constrained by their non-linear and error-prone arithmetic processes, making the cumulative error difficult to avoid in the pose estimation. To alleviate these limitations, the Manhattan World (MW) assumption was used for low-drift rotation estimation [10]–[12], consisting of one vertical and one horizontal direction in the real world (e.g., combination of wall and ground), but can only represent regular structured scenes. In order to broaden the application scenarios, the Atlanta world (AW) assumption has been attempted to be used in odometry [13]–[15], which includes one vertical and multiple horizontal directions (e.g., combination of multiple non-perpendicular walls and ground). However, the AW-based odometry is still limited by its unstable tracking strategy, as they struggle to

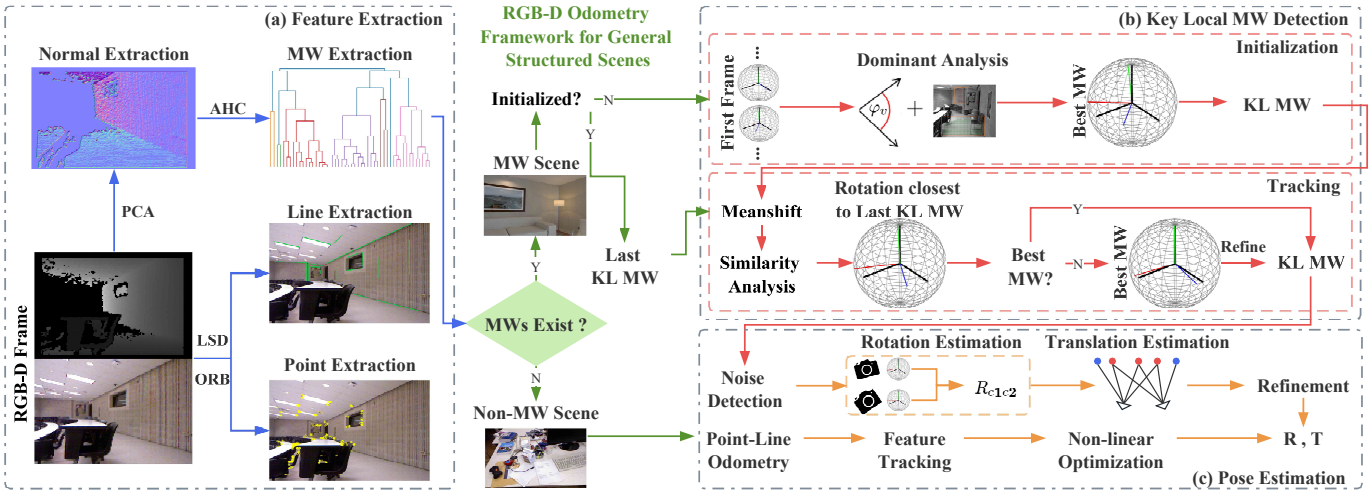


Fig. 2. Overview of our RGB-D odometry framework. (a) Firstly, features are extracted from the input RGB-D frame, normal features are extracted from the depth map, and point and line features are obtained from the RGB image. The normal and line features are utilized to detect the presence of MW in the scene. If not, we go to (c) for feature tracking; If yes, we go to (b) to select the Key Local MW based on dominance level of local MWs, and then we go to (c) to estimate the rotation and translation, respectively. Finally the camera pose is obtained.

handle the situation where local MW in current frame suddenly disappears in next frame.

To overcome these limitations, we propose an RGB-D odometry method for the general AW based on tracking by detection strategy. While tracking MW in current frame, we extract multiple local MWs based on plane detection, choose Key Local MW (KL MW) according to dominance level of these local MWs, and refine the deviation of the world coordinate system by calculating horizontal angle. We utilize Key Local MW to first estimate low-drift rotation followed by estimating accurate translation using structural features (lines and planes). The proposed method achieves low-drift pose estimation in general low-texture scenarios, and the main idea is illustrated in Fig. 1. Overall, the contributions of our approach can be summarised as follows:

- We design the tracking by detection framework to maximize the extraction of valuable information from RGB-D data, to improve the robustness of odometry while constructing AW for complex structured scenarios.
- We propose a robust RGB-D odometry for indoor structured scenes, which decouples rotation and translation based on Key Local MW to achieve low-drift pose estimation in general and complex low-texture scenes.
- Our experiments on real-world RGB-D benchmark and self-recorded datasets demonstrate that the proposed method significantly reduces cumulative error compared to state-of-the-art methods with a margin of 22%.

## II. RELATED WORK

In this section, we briefly review traditional RGB-D odometry methods, focusing on those utilizing structural features (lines and planes) to address the low-texture issue. We also explore research related to low-drift pose estimation using the Manhattan/Atlanta World assumption.

### A. Feature-based RGB-D Odometry

***Point-based:*** ORB-SLAM [4] is a representative work in point-feature based visual odometry. This method obtains the pose of key frame by tracking ORB point features, and uses loop closure detection and global bundle adjustment to form optimization module. Another notable work based on ORB-SLAM is ORB-SLAM2 proposed by Mur-Artal et al. [16], which ingeniously uses ORB point features of RGB-D frames to estimate the camera pose. ORB-SLAM2 obtain 3D points from the depth map and fuses them, utilizing the camera poses of keyframes to build an accurate 3D model. Endres et al. [17] proposed a more comprehensive and excellent open-source system, RGB-D SLAMv2, which adds loop closure detection and bundle adjustment to KinectFusion to avoid pose drift as much as possible. However, due to the difficulty of guaranteeing point feature matching accuracy in low-texture scenes, the above point-based methods cannot avoid tracking failure and cumulative error.

***Line and Plane-based:*** Several research works have incorporated high-level structural features, such as line and plane features, in addition to point features, to address challenges posed by low-texture scenes. PL-SLAM, proposed by Pumarola et al. [5], integrates line features into a point-based SLAM system, obtaining camera poses by minimizing the reprojection error of detected lines. This enhancement effectively improves the robustness of the SLAM system in low-texture scenes. Pop-up-SLAM [6] extracts planar features from a stereoscopic 3D planar model and synthesizes these features with a point-based SLAM framework, enhancing algorithm stability. Benefiting from the excellent performance of RGB-D cameras in acquiring 3D information of the scene, numerous research works have been conducted on odometry based on Line and Plane features using RGB-D data. Lu et al. [8] extract accurate 3D points and lines from RGB-D data, analyze their measurement uncertainty, and utilize maximum likelihood estimation to compute the camera motions. Plane-

Edge-SLAM [7] fully leverages constraints provided by planes and edges to achieve stable performance in any scene. Zhang et al. [9] use orthogonal and parallel relationship between planes as additional constraints to improve the stability of pose estimation in the above methods. Although line and plane-based methods can effectively cope with low-texture scenes, compared to our linear optimization system, their non-linear and non-convex optimization problems still require difficult pose graph optimization process [13]. And computational complexity of these methods increases dramatically with the growth of line and plane features, while requiring longer iterative computation times.

### B. RGB-D Odometry based on Manhattan/Atlanta World

In general, a man-made scene can be modeled by three orthogonal directions (e.g., combination of wall and ground), known as the Manhattan World assumption [14]. Although accurate vertical direction can be obtained from IMU, but horizontal direction can't. So researchers generally use RGB-D sensors to design MW/AW-based odometry. Straub et al. [18] utilized MW to achieve a stable performance in camera rotation estimation. Kim et al. [11] introduced the RANSAC method to estimate the camera rotation from a single line and a single plane. Zhou et al. [10] implemented low-drift camera rotation estimation based on the mean-shift algorithm by extracting the normal features of each pixel in the depth map. Planar-SLAM [19] incorporates the algorithm proposed in [10] into the SLAM system to enhance its robustness in indoor low-texture structural scenes. L-SLAM [20] designs a linear optimization method based on the MW assumption to reduce the computational complexity. However, when multiple non-orthogonal horizontal directions appear in the scene (e.g., walls are not perpendicular to each other), none of the aforementioned methods is applicable.

Many researchers have extended applicability of MW-based odometry to more general scenarios based on the Atlanta World assumption, which can be applied to represent more complex structured scenarios with multiple non-orthogonal horizontal directions. Both Joo et al. [21] and Liu et al. [22] used the BnB (Branch and Bound) algorithm for detecting AW structures in depth maps. Joo et al. [13] designed a module for identifying new or missing directions, updating local AW and global AW during the tracking process to construct the Atlanta World. Li et al. [14] extended the application scenario of Planar-SLAM [19] to the Atlanta World by analysing the visibility relation of MW.

It is worth mentioning that the above RGB-D odometry methods based on AW assumption are all limited by unstable tracking strategies. These algorithms (AF-SLAM [13] and Manhattan-SLAM [14]) will not be able to connect the local MW as well as avoid incorrect pose estimation when the MW tracked in the current frame suddenly disappears in next frame. In contrast, our proposed method employs an adaptive strategy that switches the tracking target based on the dominance level of local MW. This approach achieves stable and robust pose estimation in complex structured scenes by consistently tracking the Key Local MW.

## III. METHOD

Our odometry framework utilizes Key Local MW to construct Atlanta World containing multiple MWs, achieving low-drift pose estimation in more general structural scenarios (see Fig. 2 for details). In this section, We first introduce detection and tracking of Key Local MW and provide a detailed explanation of key technology of our odometry, including structural feature extraction, Key Local MW detection, low-drift pose estimation, noise detection and refinement.

### A. Representation of Key Local MW

For indoor scenes, most of them have both a ground and multiple non-perpendicular walls, which can be modeled by a vertical direction and multiple non-orthogonal horizontal directions. Obviously, simultaneously detecting and tracking local MWs is complex and difficult, thus for $i$-th frame, we only focus on the local MW that is most stable and most likely to appear in ($i$+1)-th frame (called the most dominant). We define the most dominant wall-ground combination of $i$-th frame as $MW_i^{KL}$ (see Eq. 1) by analyzing orthogonality of ground and multiple walls as well as their area occupied in this frame. We represent Key Local MW as a vector:

$$MW_i^{KL} = \left(R, \varphi_{refine}, L_d^{\max}\right), \tag{1}$$

where $MW_i^{KL}$ represents Key Local MW detected in the $i$-th frame, $R \in SO(3)$ is rotation matrix from the camera to $MW_i^{KL}$, consisting of three orthogonal directions of the most dominant MW observed by the camera. $\varphi_{refine}$ serves to indicate horizontal refine angle of $MW_i^{KL}$ relative to $MW_{i-1}^{KL}$, $L_d^{\max}$ indicates the dominance level (highest among local MWs) of $MW_i^{KL}$, which is determined by the number of nodes detected in planar features (as described in [23]) and the angle between the most dominant wall-ground combination (for analysis of orthogonality).

### B. Feature Extraction

Our approach leverages structural information of the environment to adapt to low-texture scenes, including point and line features for translation estimation and normal features for rotation estimation.

***Points:*** We use ORB descriptor to extract point features in RGB images, known for its fast performance and robustness [24]. We denote the $j$-th 2D point feature on the $i$-th frame as $p_{i,j} = \left(x_{p_{i,j}}, y_{p_{i,j}}\right)$, and its corresponding 3D map point as $P_{i,j} = \left(x_{P_{i,j}}, y_{P_{i,j}}, z_{P_{i,j}}\right)$. Here, $x_{p_{i,j}}$ and $y_{p_{i,j}}$ represent the pixel coordinates of 2D feature points, while $x_{P_{i,j}}$, $y_{P_{i,j}}$ and $z_{P_{i,j}}$ represent the coordinates of 3D map points in the world coordinate system.

***Lines:*** We employ Fast Line Segment Detector (LSD) to extract line features from RGB images [25], We denote the $i$-th 2D line feature on the $i$-th frame as $l_{i,j} = \left(p_{i,j}^{start}, p_{i,j}^{end}\right)$, and its corresponding 3D line as $L_{i,j} = \left(P_{i,j}^{start}, P_{i,j}^{end}\right)$. $p_{i,j}^{start}$ and $p_{i,j}^{end}$ correspond to the start and end points of 2D lines, and use $P_{i,j}^{start}$ and $P_{i,j}^{end}$ to denote the start and end points of 3D lines in the world coordinate system.

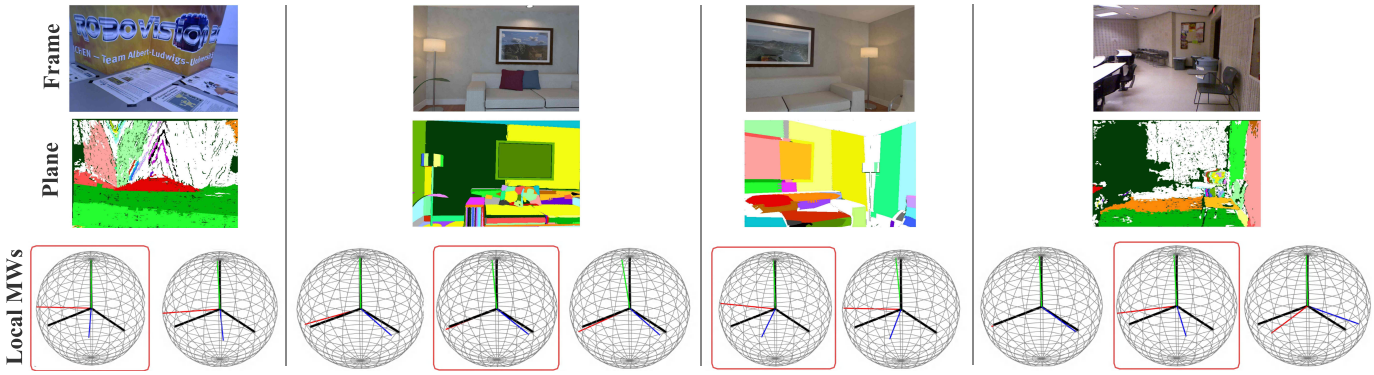


Fig. 3. Examples of Key Local MW detection module. The first line shows examples of Datasets, including TUM RGB-D [29], ICL-NUIM [30], TAMU [8] dataset. The second row shows plane detection results. The third row indicates the multiple local MWs detected in current frame, and the red box serves to indicate the Key Local MW selected by our algorithm.

***Normals:*** We first utilise depth information to obtain the 3D points in the camera coordinate system corresponding to each pixel in the RGB image. Subsequently, we fit a surface through the 3D points corresponding to a patch of pixels. For the normal vector estimation part, Principal Component Analysis (PCA) is employed to extract the normal information of the pixel at the centre of the patch. The set of normal vectors of the $i$-th frame is represented as $N_i = \{n_{i,j}\}_{j=1}^{(l-5)(w-5)}$, where $n_{i,j} = \left(n_{i,j}^x, n_{i,j}^y . n_{i,j}^z\right)^T$, $l$ and $w$ indicates the length and width of depth map.

### C. Tracking-by-Detection Algorithm for Key Local MW

For a given RGB-D sequence, we adapt the tracking by detection concept from multi-target tracking [26], [27] to design an algorithm for tracking multiple Manhattan Worlds (MWs) in complex scenes, extending application scenarios to the broader Atlanta World. While tracking $MW_{i-1}^{KL}$ from previous frame, we detect and determine multiple MWs and $MW_i^{KL}$ in current frame, enhancing the stability and reliability of low-drift odometry in intricate indoor environments.

***Detection:*** The detection module encompasses two essential steps: detecting all potential local MWs in current frame and determining their dominance level. In the initial stage, we employ the Agglomerative Hierarchical Clustering (AHC) algorithm [23] to extract planar features for the surface normal (see Sec. III-B) in the current frame. We represent all planar features in current frame in the following form:

$$S_i = \{n_{i,j}, N_{i,j}\}_{j=1}^{M}, \tag{2}$$

where $S_i$ denotes the set of planar features detected in the $i$-th frame, $n_{i,j}$ represents the normal vector corresponding to the $j$-th planar feature, and $N_{i,j}$ means the number of support points in the point cloud for the $j$-th planar feature, $M$ represents the total number of planar features detected in the $i$-th frame. From the detected set, two planar features are randomly selected to assess their orthogonality. In order to minimise the computational complexity, we design a simple filtering mechanism (as in Eq. 3) to determine whether planar features can form a local MW or not.

$$\begin{cases} \angle(a,b) \in (90-\theta, 90+\theta) \\ N_{i,a} + N_{i,b} > \sigma \end{cases}, \tag{3}$$

where a and b are two randomly selected planar features. $\theta = 2.5°$, which is used to ensure orthogonality between planes. Since we set the minimum number of support points for each planar feature to 150 in plane detection algorithm [23], $\sigma$ is set to 500, which is used to ensure that planar features will appear in next frame. Let $r_1 = \left(n_{i,a}^x, n_{i,a}^y, n_{i,a}^z\right)^T$, $r_2 = \left(n_{i,b}^x, n_{i,b}^y, n_{i,b}^z\right)^T$, and $r_3 = r_1 \times r_2$. The rotation matrix from camera to the local MW is denoted as:

$$R = (r_1, r_2, r_3)^T, \tag{4}$$

Owing to noise in depth map and errors in plane detection, we can't guarantee that $R$ is an orthogonal matrix. To address this limitation, we re-represent the camera-to-local MW rotation transformation in terms of the closest $R$ to $R_{cm} \in SO(3)$ following [10]:

$$\begin{aligned} R_{cm} &= UV^T, \ where \\ \left(U, \varepsilon, V^T\right) &= SVD\left(r_1, r_2, r_3\right)^T. \end{aligned} \tag{5}$$

In the second stage, in order to identify Key Local MW we require for tracking among multiple local MWs, we calculate dominance level of each local MW according to Eq. 6.

$$L_d = N_{i,a} + N_{i,b} - \tau \left| \frac{\pi}{2} - \angle(a,b) \right|, \tag{6}$$

After extensive experimentation, we set $\tau$ to 5000 for combining orthogonality and the number of support points to express degree of dominance of MW. With above two stages, the detection module leads us to obtain multiple local MWs in current frame as well as their dominance level $L_d$.

***Initialization:*** For the first frame of RGB-D sequence, We first detect all local MWs and calculate their $L_d$, directly using the one that has the highest $L_d$ as $MW_1^{KL}$. During the initialization phase, $\varphi_{refine}$ is set to zero.

***Tracking:*** Similar to the detection module, this part is also executed in two phases: tracking $MW_{i-1}^{KL}$ of the last frame, and determining $MW_i^{KL}$ for the current frame. In the first stage, we apply a similar approach as in [10], where we project

the normal vector $N_{i-1}$ of the ($i$-1)-th frame onto the unit sphere, and project each of the three principal directions of $MW_{i-1}^{KL}$ onto the tangent plane, utilizing mean-shift algorithm to update each direction to the center position where the density of the normal vectors is maximal.

As $MW_{i-1}^{KL}$ may disappear from the camera's field of view at any time, this could result in an incorrect $MW_i^{track}$, leading to pose estimation inaccuracies due to coordinate system inconsistencies. To address this issue, in the second stage, multiple local MWs are extracted in current frame using the detection module along with their dominance level $L_d$. We consider the MW with the highest $L_d$ among local MWs to be the most stable one, least likely to disappear in next frame. It is then defined as $MW_i^{KL}$, but if $L_d$ of the most stable local MW is not as high as that of $MW_i^{track}$, we consider $MW_i^{track}$ as $MW_i^{KL}$ for $i$-th frame.

When replacing Key Local MW, we need to update $\varphi_{refine}$ to obtain the angle between the two non-perpendicular walls, thus refining the deviation of the coordinate system (as described in Sec. III-A). Let the dominance level of $MW_i^{track}$ (obtained from the closest MW detected in the current frame) be $L_d^{track}$, and the dominance level of local MW with the highest $L_d$ be $L_d^{\max}$. We determine the updated $\varphi_{refine}^{'}$ according to Eq. 7.

$$\begin{aligned} &if\ L_d^{\max} > L_d^{track}, \quad \varphi_{refine}^{'} = \varphi_{refine} + \Delta\varphi, \\ &if\ L_d^{\max} = L_d^{track}, \quad \varphi_{refine}^{'} = \varphi_{refine}. \end{aligned} \tag{7}$$

where $\Delta\varphi$ is the angle between the two non-orthogonal horizontal directions of $MW_i^{track}$ and $MW_i^{max}$.

### D. Pose Estimation

Rotation and translation of camera to the world coordinate system are decoupled and estimated separately. We determine rotation by tracking Key Local MW in each frame, and estimate the translation based on point and line features.

***Rotation:*** For every two frames in the RGB-D sequence, we detect the rotation transformation from each frame to the Key Local MW using the method described in Sec. III-C. The frame-to-frame rotation is calculated as follows:

$$R_{cc'} = R_{cm} R_{refine} R_{c'm'}^T, \tag{8}$$

where $R_{refine} \in SO(3)$ is a rotation matrix with a horizontal rotation angle of $\varphi_{refine}$.

***Translation:*** We use point-line features to estimate translation. Unlike the way [28] estimates both rotation and translation, we define the 3Dof translation parameter $\varphi_i$ of frame i as the optimization variable and construct the least squares cost function as in Eq. 9.

$$\varphi_i^* = arg\min_{\varphi_i} \sum_{j \in n_p} {e_{i,j}^p}^T P_{i,j}^p e_{i,j}^p + \sum_{j \in n_l} {e_{i,j}^l}^T P_{i,j}^l e_{i,j}^l, \tag{9}$$

where $R_{wc}$ is obtained from Eq. 8. $e_{i,j}^p$ and $e_{i,j}^l$ are reprojection errors of $j$-th point feature and line feature in $i$-th frame, respectively. $n_p$ and $n_l$ are total number of point features and line features in ith frame. Finally, the optimal translation parameters are solved using Levenberg-Marquardt algorithm.

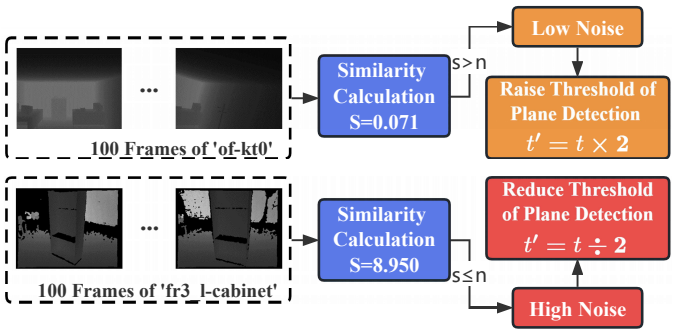


Fig. 4. Overview of the noise detection module. Left is a flowchart of the algorithm, $s$ is similarity of multiple local MWs, $n$ is a pre-set threshold. Right shows the performance of the noise detection module on two sequences, 'of-kt0' and 'fr3_l-cabinet', which were determined to be low and high noise sequences, respectively. $t$ is the plane detection threshold for current hundred frames, and $t'$ is the threshold for next 100 frames.

### E. Noise Detection

The MW detection module of our odometry is highly sensitive to noise in the depth map due to the use of a plane detection algorithm [23] for MW extraction in the scene. Interestingly, in the presence of significant noise, we effectively suppress the noise by choosing $MW_i^{KL}$ with the highest dominance level. However, in scenes with minimal noise, the presence of incorrect planar features may introduce discontinuities in $MW_i^{KL}$. To address this limitation, we design a noise detection module for detecting noise in the scene and thus adaptively adjusting the plane detection threshold. Specifically, we first accessing all local MWs in current 100 frames and compute their similarity by Eq. 10.

$$S = \sum_{i \in 100} \sum_{b \in n_i} \sum_{a \in n_i} trace\left(R_{cm}^{i,a} \cdot R_{cm}^{i,b}\right) - 1, \tag{10}$$

where $S$ represents the similarity of local MWs in current 100 frames, where a larger value indicates higher similarity. $R_{cm}^{i,a}$ represents the rotation matrix corresponding to $a$-th local MW in $i$-th frame, and $n_i$ is the total number of local MWs in $i$-th frame . As shown in Fig. 4, if $S$ is larger than the preset threshold, the scene is considered to be low-noise, and plane detection threshold is raised to reduce planar features and avoid detection errors. Conversely, we consider the scene to be high-noise, lower the threshold, increase planar features, and further suppress noise by selecting $MW_i^{KL}$.

### F. Refinement

All the mentioned methods rely on structured scenes that strictly adhere to Manhattan World assumption. However, in cases where Manhattan World cannot be detected or current scene deviates from Manhattan assumption, we revert to point-line SLAM framework to obtain the camera 6Dof pose. We project point and line features detected in current frame to the last 20 frames, calculate the reprojection error, and reuse method similar to [28] to estimate both rotation and translation if the error is too large.

TABLE I
COMPARISON OF RMSE OF ABSOLUTE TRAJECTORY ERROR (M) ON TUM RGB-D [29] AND ICL-NUIM [30] DATASETS.

| | | Dominance Level | | Methods | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | Sequence | KL MW | Texture | ORB-SLAM2 [16] | DVO [31] | L-SLAM [20] | Planar-SLAM [19] | AF-SLAM [13] | MW–SLAM [14] | Ours | Ours+ND |
| TUM RGB-D | fr1_xyz | low | high | **0.010** | 0.026 | - | × | - | 0.010 | 0.013 | 0.013 |
| | fr1_desk | low | high | **0.022** | 0.036 | - | × | - | 0.027 | 0.033 | 0.031 |
| | fr3_snt-far | high | low | × | 0.039 | 0.141 | 0.041 | - | 0.040 | 0.040 | **0.036** |
| | fr3_snt-near | high | low | × | 0.021 | 0.066 | 0.027 | - | 0.023 | 0.023 | **0.018** |
| | fr3_st-far | medium | high | **0.011** | 0.390 | 0.212 | - | - | 0.022 | 0.022 | 0.012 |
| | fr3_st-near | medium | high | **0.011** | 0.041 | 0.156 | - | - | 0.012 | 0.012 | **0.011** |
| | fr3_cabinet | high | low | × | 0.690 | 0.291 | 0.035 | - | 0.023 | 0.019 | **0.017** |
| | fr3_l-cabinet | high | low | × | 0.979 | 0.140 | 0.071 | - | 0.083 | **0.064** | **0.064** |
| ICL NUIM | lr-kt0 | medium | medium | 0.010 | 0.108 | 0.012 | 0.006 | 0.014 | 0.007 | **0.005** | **0.005** |
| | lr-kt1 | low | medium | 0.185 | 0.059 | 0.027 | 0.015 | 0.035 | **0.011** | 0.017 | **0.011** |
| | lr-kt2 | medium | medium | 0.028 | 0.375 | 0.053 | 0.020 | 0.027 | **0.015** | 0.019 | **0.015** |
| | lr-kt3 | low | medium | 0.014 | 0.433 | 0.143 | 0.012 | 0.117 | **0.011** | 0.012 | 0.012 |
| | of-kt0 | high | low | 0.049 | 0.244 | **0.020** | 0.024 | 0.041 | 0.036 | 0.025 | **0.020** |
| | of-kt1 | high | low | 0.079 | 0.178 | 0.015 | 0.020 | 0.022 | 0.013 | **0.012** | **0.012** |
| | of-kt2 | high | low | 0.025 | 0.099 | 0.026 | 0.011 | 0.027 | 0.015 | 0.015 | **0.014** |
| | of-kt3 | high | low | 0.065 | 0.079 | **0.011** | 0.014 | 0.025 | 0.013 | 0.015 | 0.013 |

× represents tracking or initialization failure. - represents algorithms or results are not avaliable. **Ours+ND** represents our method with noise detection.

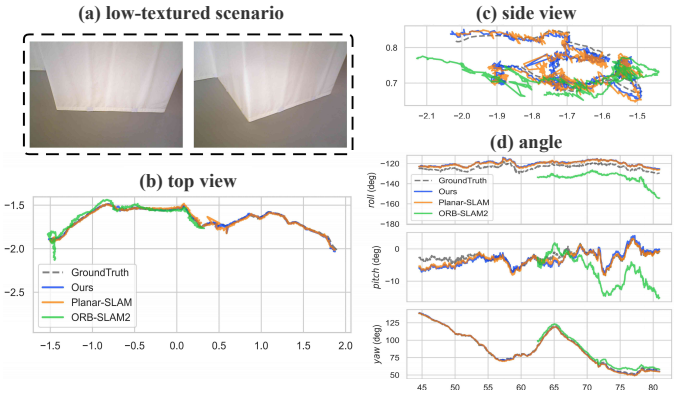


Fig. 5. Performance of Planar-SLAM, ORB-SLAM2 and our method on the typical low texture sequence 'fr3_snt-near' of the TUM RGB-D dataset [29]. (a) is an example of the sequence, (b,c) represent two different views of the tracking trajectory, and (d) shows the rotation estimation results.

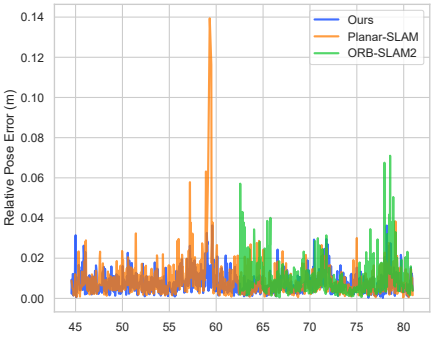


Fig. 6. Comparison of Planar-SLAM, ORB-SLAM2 and proposed odometry method for per-frame Relative Pose Error (M) on 'fr3_snt-near' sequence. The horizontal axis represents the timestamp. ORB-SLAM2 lost the first half of the data due to initialization failure.

## IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed odometry algorithm frame by frame on public benchmark and self-recorded datasets in several complex structured scenarios, including TUM RGBD [29], ICL-NUIM [30], and TAMU RGB-D [8] datasets. We compare our odometry with feature-based methods ORB-SLAM2 [16] and DVO [31], MW-based methods L-SLAM [20] and Planar-SLAM [19] and AW-based methods AF-SLAM [13] and MW-SLAM [14]. All experiments run on a PC with Intel Core i9-12900KF 3.2GHz CPU and 64GB RAM. It should be noted that, for a fair comparison with other SLAM systems, we turned off backend optimization in all methods, including loop closure and global bundle adjustment.

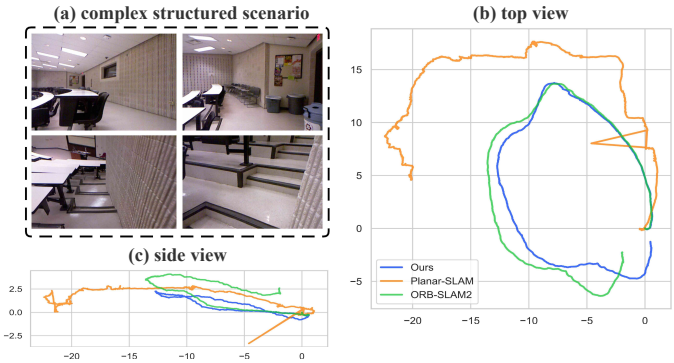


Fig. 7. Performance of Planar-SLAM, ORB-SLAM2 and our method in Auditorium-const for a typical complex structured scenario (containing multiple MWs) on TAMU dataset [8]. (a) is an example of the sequence and (b,c) represent two different views of the tracking trajectory.

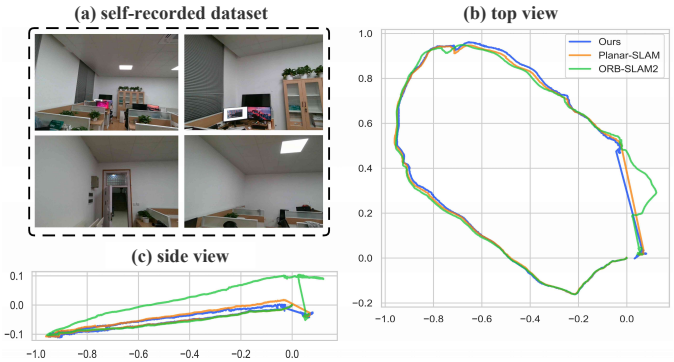


Fig. 8. Performance of Planar-SLAM, ORB-SLAM2 and our method in Auditorium-const for another complex structured scenario (containing multiple planes) on Self-recorded dataset. (a) is an example of the sequence and (b,c) represent two different views of the tracking trajectory.

TABLE II
RMSE OF RELATIVE POSE ERROR (M) ON TUM RGB-D DATASET [29].

| | Dominance Level | | Methods | | |
|---|---|---|---|---|---|
| Sequence | KL MW | Texture | Ours | Planar-SLAM | ORB-SLAM2 |
| fr3_snt-near | medium | low | **0.009** | 0.027 | × |

× represents tracking or initialization failure.

TABLE III
RMSE OF TRAJECTORY ENDPOINT DRIFT (M) ON TAMU DATASET [8].

| | Dominance Level | | Methods | | |
|---|---|---|---|---|---|
| Sequence | KL MW | Texture | Ours | Planar-SLAM | ORB-SLAM2 |
| Auditorium | medium | low | **1.42** | 20.64 | 3.53 |

TABLE IV
RMSE OF TRAJECTORY ENDPOINT DRIFT (M) ON OUR DATASET.

| | Dominance Level | | Methods | | |
|---|---|---|---|---|---|
| Sequence | KL MW | Texture | Ours | Planar-SLAM | ORB-SLAM2 |
| office | medium | low | **0.063** | 0.075 | 0.097 |

TABLE V
AVERAGE RUNTIME (MS) ON TUM [29] AND ICL [30] DATASETS

| Dataset | ORB-SLAM2 | Planar-SLAM | Ours | Ours+ND |
|---|---|---|---|---|
| TUM RGB-D | 28.91 | 42.11 | 41.27 | 41.84 |
| ICL-NUIM | 30.51 | 41.74 | 41.23 | 41.95 |

**Ours+ND** represents our method with noise detection.

### A. Estimation Accuracy

***TUM RGB-D dataset:*** The TUM RGB-D dataset [29] is a well-known public benchmark for evaluating performance of SLAM systems, providing real-world indoor data with diverse textures and structures. Table I shows ATE RMSE achieved by our method compared to other methods. We also evaluated dominance level of each RGB-D sequence as a reference, including KL MW and texture. Higher dominance level indicates more stable MW structure and richer texture. The 'xyz' and 'desk' sequences of fr1 are rich in texture, but since MW structure of these sequences is so poor that they hardly meet MW assumption, methods relying exclusively on MW: L-SLAM, Planar-SLAM and AF-SLAM fail to work. In such cases, our method and MW-SLAM degrade into point-line odometry, still achieving reliable performance. However, due to poor structural regularity of these two sequences, their actual performance is not inferior to ORB-SLAM2.

Benefiting from strict MW structure in the fr3 sequence, both MW- and AW-based methods outperform traditional approaches. While the 'st' sequence, with rich texture, enables ORB-SLAM2 to achieve high accuracy, it struggles in the texture-lacking 'snt' and 'cabinet' sequences. Both MW-SLAM and our method employ point-line odometry for enhanced accuracy. However, RGB-D noise may lead to erroneous MW detection in MW-SLAM. Our method addresses this with a noise detection module (as described in Sec. III-E), adapting plane detection thresholds and selecting Key Local MW with highest dominance level. This effectively suppresses noise, yielding superior results to MW-SLAM and our method without ND in both 'snt' and 'cabinet' sequences.

***ICL-NUIM dataset:*** The ICL-NUIM dataset [30] consists of synthetic scenes without noise, featuring living room and office environments. These scenes lack texture but exhibit strong structure, aligning with MW assumption. The lower section of Table I presents ATE RMSE for our method compared to others in all sequences. As all ICL-NUIM sequences conform to perfect MW structures, MW- and AW-based methods perform credibly. Leveraging noise detection module and employing point-line structural features, our full odometry achieves optimal accuracy in most sequences compared to the case without ND. Notably, in 'lr-kt1' and 'lr-kt3' sequences, where scenes have only a single wall without ground, our method matches MW-SLAM's accuracy due to the use of point-line features. Despite the strict MW structure in 'of' sequences favoring MW-based methods like L-SLAM, our method still achieves stable results, outperforming Planar-SLAM and MW-SLAM by adaptively adjusting plane detection threshold to reduce incorrect features and achieve superior performance (as described in Sec. III-E).

### B. Drift Evaluation under Challenging Scenarios

***Low Texture:*** In order to evaluate the cumulative error of the proposed odometry algorithm, we first tested the performance of the feature-based method: ORB-SLAM2, MW-based method: Planar-SLAM, and our method in the low-texture sequences of the TUM RGB-D dataset. Fig. 5 shows tracking trajectory of each method on the 'snt' sequence. As shown in Fig. 5-(a), due to the lack of texture, ORB-SLAM2 struggles to successfully initialize, loses the first half of the trajectory, and exhibits a significant drift during tracking compared to ground truth. The 'snt' sequence is a structured scenario that strictly conforms to the MW assumptions, and it is worth mentioning that such a scenario is highly favorable for Planar-SLAM, which relies on a single MW only. Nevertheless, as shown in Fig. 5-(c), from the side view, our method fits the ground truth more closely. As shown in Fig. 5-(d), our method and Planar-SLAM achieve low-drift rotation estimation. As shown in Fig. 6, the proposed method outperforms other methods in terms of RPE achieved in almost every frame. Table II shows the RTE RMSE tested on low texture sequences, providing a measure of odometry drift. Finally, our method demonstrates superior performance compared to Planar-SLAM.

***Complex Structure (Multiple MWs):*** We define indoor scenes with non-perpendicular walls as complex structured scenes. We evaluated the performance of ORB-SLAM2, Planar-SLAM and our method on a sequence of complex structured scenes from the TAMU RGB-D dataset [8]. Fig. 7 shows the tracking trajectories, and Planar-SLAM based on a single MW exhibits a significant drift due to the presence of multiple MWs in the scene. However, ORB-SLAM2 successfully tracks the sequence due to its rich texture. In contrast, our method achieves low-drift pose estimation in complex structured scenes by tracking Key Local MWs, resulting in superior tracking results compared to ORB-SLAM2. Table III shows TED for each method, indicating the distance between the start and end points of the trajectory and used to evaluate the degree of drift of the odometry. The performance of Planar-

SLAM is very poor, and although ORB-SLAM2 can achieve stable results, the drift of our method is lower than it.

***Complex Structure (Multiple Planes):*** Since we depend on the plane detection algorithm to extract multiple local MWs in the scene, it is crucial to evaluate whether our odometry can operate stably in scenes with multiple planes. For this purpose, we recorded an RGB-D dataset for an office scene with multiple planes (consisting of desks) using a hand-held device: Intel RealSense D455. Fig. 8 illustrates the tracking trajectory of each method. ORB-SLAM2 exhibits severe drift in the height dimension. From Table IV, we can conclude that our method still achieves the lowest drift in multiplanar scene.

### C. Runtime Analysis

We tested per-frame runtime of ours and other methods on publicly available datasets. As shown in Table V, although our method and Planar-SLAM, which is also based on MW, underperform ORB-SLAM2 (around 35fps), they still achieve an average of 25fps while maintaining high accuracy, which can satisfy requirement of real-time operation. In addition, since noise detection only needs to do one operation every 100 frames, it does not take up the runtime.

## V. CONCLUSIONS

In this paper, we present an RGB-D odometry tailored for complex structured scenes under AW assumption. We adopt the concept of the tracking-by-detection algorithm from the field of multi-target tracking, focusing on tracking Key Local MW to construct AW. Extensive experiments demonstrate that our proposed method attains low-drift pose estimation in various general structural scenarios. In future research, we aim to develop robust plane detection algorithms for complex scenes and explore stable tracking strategies (perhaps using IMU to get accurate vertical direction) tailored for structured environments with a single wall and no ground, loop closure and relocalization will also be considered.

## REFERENCES


[1] B. Huang, J. Zhao, and J. Liu, "A Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Networks," 2020. [Online]. Available: http://arxiv.org/abs/1909.05214

[2] Z. Lin, Z. Gao, B. M. Chen, J. Chen, and C. Li, "Accurate LiDAR-Camera Fused Odometry and RGB-Colored Mapping," IEEE Robot. Autom. Lett., vol. 9, no. 3, pp. 2495–2502, 2024.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1052–1067, 2007.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, 2015.

[5] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 4503–4508.

[6] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up SLAM: Semantic Monocular Plane SLAM for Low-texture Environments," 2017. [Online]. Available: http://arxiv.org/abs/1703.07334

[7] Q. Sun, J. Yuan, X. Zhang, and F. Duan, "Plane-Edge-SLAM: Seamless Fusion of Planes and Edges for SLAM in Indoor Environments," IEEE Trans. Automat. Sci. Eng., vol. 18, no. 4, pp. 2061–2075, 2021.

[8] Y. Lu and D. Song, "Robust RGB-D Odometry Using Point and Line Features," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3934–3942.

[9] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-Plane SLAM Using Supposed Planes for Indoor Environments," Sensors, vol. 19, no. 17, p. 3795, 2019.

[10] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3d sensors in manhattan worlds," in Asian Conference on Computer Vision, Springer, 2016, pp. 3–19.

[11] P. Kim, B. Coltin, and H. J. Kim, "Indoor RGB-D Compass from a Single Line and Plane," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 4673–4680.

[12] J. P. Company-Corcoles, E. Garcia-Fidalgo, and A. Ortiz, "MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry," IEEE Robot. Autom. Lett., vol. 7, no. 2, pp. 2803–2810, 2022.

[13] K. Joo, T. -H. Oh, and F. Rameau et al., "Linear RGB-D SLAM for Atlanta World," in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 1077-1083.

[14] R. Yunus, Y. Li, and F. Tombari, "ManhattanSLAM: Robust Planar Tracking and Mapping Leveraging Mixture of Manhattan Frames," in 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, May 2021, pp. 6687–6693.

[15] D. Yan, H. Jiang and T. Li et al., "Efficient Vanishing Point Estimation for Accurate Camera Rotation Estimation in Indoor Environments," IEEE Robot. Autom. Lett., vol. 8, no. 11, pp. 6899–6906, 2023.

[16] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Trans. Robot., vol. 33, no. 5, pp. 1255–1262, 2017.

[17] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D Mapping With an RGB-D Camera," IEEE Trans. Robot., vol. 30, no. 1, pp. 177–187, 2014.

[18] J. Straub et al., "Real-time manhattan world rotation estimation in 3D," in 2015 International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 1913–1920.

[19] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "RGB-D SLAM with Structural Regularities," in 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 11581–11587.

[20] P. Kim, B. Coltin, and H. J. Kim, "Linear RGB-D SLAM for Planar Environments," in European conference on computer vision, Springer, 2018, pp. 350–366.

[21] K. Joo and T.-H. Oh, "Globally Optimal Inlier Set Maximization for Atlanta World Understanding," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 10, pp. 2656–2669, 2020.

[22] Y. Liu, G. Chen, and A. Knoll, "Globally Optimal Vertical Direction Estimation in Atlanta World," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 4, pp. 1949–1962, 2020,

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

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2564–2571.

[25] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 4, pp. 722–732, 2010.

[26] Y. Zhang et al., "ByteTrack: Multi-object Tracking by Associating Every Detection Box," in European conference on computer vision, Springer, 2022, pp. 1–21.

[27] P. Sun et al., "DanceTrack: Multi-Object Tracking in Uniform Appearance and Diverse Motion," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2022, pp. 20961–20970.

[28] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," IEEE Trans. Robot., vol. 35, no. 3, pp. 734–746, 2019.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 573–580.

[30] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 1524–1531.

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