2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

STL-SLAM: A Structured-Constrained RGB-D SLAM Approach to Texture-Limited Environments

Juan Dong, Maobin Lu, Chen Chen, Fang Deng and Jie Chen

Abstract— **Most RGB-D-based SLAM methods assume texture-rich environments, making them susceptible to significant tracking errors or complete failures in the absence of texture features. Moreover, many existing methods encounter substantial rotation estimation errors, leading to long-term drift in tracking. This paper proposes a novel structured-constrained RGB-D SLAM method (STL-SLAM) for texture-limited environments. Compared to the existing methods, STL-SLAM can deal with environments without abundant texture information and significantly reduce long-term drift caused by rotation estimation errors. We assess the distribution complexity of pixels in an image by calculating the information entropy and pre-processing accordingly. We also present an efficient Manhattan Frames (MF) detection strategy based on orthogonal planes and lines. If MF is detected, we decouple rotation and translation, estimate drift-free rotation based on the Manhattan World (MW) coordinate system, and then estimate translation by minimizing the re-projection error of point, line, and plane features. In non-Manhattan Frames, the 6-DoF pose estimation is performed holistically, with the incorporation of structural constraints of parallel and perpendicular planes, as well as parallel and vertical lines, into the optimization process. Finally, we evaluate our method on public datasets and in real-world environments, which shows that our proposed method achieves superior performance compared to its counterparts.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental technology for real-time pose estimation and environment construction in robotics. In visual SLAM, RGB-D cameras can process data faster and generate real-time 3D maps more efficiently than monocular and binocular cameras.

Most current visual SLAM methods are always for special types of environments and are difficult to adapt to texture-limited scenes. Texture-limited means that there are not enough point features in the environment, which makes the traditional point feature-based method ineffective. Methods based on point features such as [1], [2] are prone to alignment errors in scenes with insufficient texture. In recent years, researchers integrate deep learning into various frameworks to enhance the robustness of SLAM, as exemplified by [3], [4]. These methods often emphasize models tailored to specific objectives, necessitating separate training, storage, and execution for each model. Consequently, these methods exhibit limited generalizability. The method proposed by [5] is a semantic SLAM approach that leverages objects as landmarks to enhance point associations and reduce odometry errors. However, in certain scenarios where specific objects are scarce or absent in the environment, the accuracy of localization may be affected.

Compared with methodologies that depend exclusively on point features, strategies that incorporate a combination of point, line, and plane features provide a more holistic representation of environmental geometry. These multi-feature fusion methods are versatile across different types of scenes, facilitating the establishment of environment maps with greater efficiency. In summary, they possess the potential to significantly boost the performance and robustness of tracking.

Camera trajectory drift due to rotation estimation also leads to large cumulative errors [6]. Although global bundle adjustment is an effective optimization technique to reduce drift in processing large-scale maps, it has high computational costs. Loop closure can also correct errors in the map, but it only works when the robot revisits a specific location.

To address these challenges, we propose a structured-constrained RGB-D SLAM method (STL-SLAM). The main contributions are summarized as follows:

i) A real-time, high-accuracy, and robust indoor RGB-D SLAM framework for texture-limited environments is established. This framework is based on point, line, and plane features.
ii) An efficient Manhattan Frames (MF) detection method used to establish a Manhattan map for drift-free rotation estimation is proposed.
iii) An innovative pose estimation method based on structural constraints is developed, which minimizes parallel and perpendicular errors of lines and planes.

We evaluate our method on public datasets and in real-world environments. The experimental results show that our method performs well in texture-limited environments and outperforms its counterparts in tracking accuracy.

II. Related Work

ORB-SLAM2 [7] is a point-feature-based visual SLAM, which includes loop closure, relocation, and map reuse. It is

This work was supported in part by National Key R&D Program of China under Grant 2021ZD0112600, in part by National Natural Science Foundation of China under Grant 62373058, in part by Key Program of National Natural Science Foundation of China under Grant 61933002, in part by National Science Fund for Distinguished Young Scholars of China under Grant 62025301, in part by Natural Science Foundation of Chongqing under Grant 2021ZX4100036, in part by the Basic Science Center Programs of NSFC under Grant 62088101, and in part by Shanghai Municipal Science and Technology Major Project 2021SHZDZX0100.

Juan Dong, Maobin Lu, Chen Chen and Fang Deng are with the School of Automation, Beijing Institute of Technology, Beijing, China, and are also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. Jie Chen is with Beijing Institute of Technology, Beijing, China, and is also with Tongji University, Shanghai, China. E-mail: dongjuan@bit.edu.cn, lumaobin@bit.edu.cn, xiaofan@bit.edu.cn, dengfang@bit.edu.cn, chenjie@bit.edu.cn.

Corresponding author: Maobin Lu (lumaobin@bit.edu.cn).

the first open-source SLAM system supporting monocular, stereo, and RGB-D cameras, achieving real-time performance and high accuracy. Furthermore, ORB-SLAM3 [1] makes some improvements and extensions. Based on it, Park et al. [2] reduce computational and storage resources through spatial point sparsification, while improving the accuracy of pose estimation.

Compared to point features, line features contain more comprehensive geometric information. Klein et al. [8] address tracking failure in SLAM systems caused by image blurring due to rapid camera motion by integrating point and line features.

Pumarola et al. [9] introduce monocular PL-SLAM that builds upon [10], while Gomez-Ojeda et al. [11] present stereo PL-SLAM. Monocular and stereo PL-SLAM employ the LSD detection algorithm [12] to identify line features and then integrate them with point features at each stage of SLAM. These advancements enable the system to function effectively, even when a significant portion of point features becomes temporarily obscured or unavailable.

Furthermore, specific environments exhibit notable plane features that attract researchers' attention. Gee et al. [13] introduce a method integrating lines and planes into map construction to enhance real-time SLAM system mapping robustness. Li et al. [14] present a SLAM algorithm optimized for indoor environments, integrating points, lines, and planes. Structure PLP-SLAM in [15] is a modular multi-feature monocular SLAM system capable of adapting to various environments and tightly integrating semantic features. These methods primarily focus on minimizing plane re-projection errors to enhance pose estimation accuracy.

Significant drift often occurs in the rotation estimation of SLAM. In order to fully leverage the potential of plane features in the environment and reduce drift in rotation estimation, some algorithms are developed based on structural assumptions, exemplified by the Manhattan World (MW) model [16].

Building upon MW, OPVO [17] utilizes orthogonal plane structures to attain absolute, drift-free orientation estimation for an RGB-D camera. Plane features based on the MW assumption are also used in [18], [19], [20] to estimate drift-free rotations. It is noted that these methods require constant visibility of at least two orthogonal planes in the environment. Due to their over-reliance on the MW assumption, they are only applicable to specific scenarios, leading to frequent tracking failures in many common environments.

LPVO [21] alleviates this limitation by introducing orthogonal line detection. L-SLAM [22] extends upon the drift-free rotation estimation of LPVO, updating camera translation using a linear Kalman filter (KF) framework. SP-SLAM [23] enhances pose estimation stability and accuracy by incorporating constraints on parallel and perpendicular planes. ManhattanSLAM [24] categorizes frames into Manhattan and non-Manhattan Frames, employing distinct tracking strategies accordingly. It broadens the system's applicability to various environments.

III. METHOD

In this section, we present a comprehensive overview of our proposed RGB-D SLAM method, as depicted in Fig. 1.

A. Pre-processing

We propose an image pre-processing strategy based on one-dimensional entropy, which serves as a metric for evaluating image complexity and information uncertainty. If the entropy falls below a predefined threshold (set to 2.15 in our experiments), indicating insufficient texture in the image, we apply image enhancement techniques to improve image quality.

B. Feature Extraction and Matching

1) Points: We denote the pixel point as $p = [u, v]^T \in \Omega$, where $\Omega \in \mathbb{R}^2$ is the image domain. We denote the space point as $P = [X, Y, Z]^T$, which represents the back-projected point corresponding to pixel p in the camera coordinate. We define the operator $\gamma : \mathbb{R}^3 \to \Omega$ as $\gamma(P) = p$.

2) Lines: We employ the LSD algorithm [12] for line detection. This algorithm identifies the two endpoints p_a and p_b of each line in the image, corresponding to spatial points P_a and P_b in the camera coordinate system, where $\gamma(P_a) = p_a$, and $\gamma(P_b) = p_b$. Through the two endpoints, the lines I in the image can be computed as: $I = p_a \times p_b / ||p_a \times p_b||$.

3) Planes: The agglomerative hierarchical clustering (AHC) algorithm [25] is an efficient method for plane detection. We represent the plane in Hessian form as $\pi = (n^T, d)^T$, where $n = (n_x, n_y, n_z)^T$ denotes the unit normal vector, and d signifies the distance from the origin.

C. MF Detection and Matching

A frame successfully detecting and matching the MW coordinate system or weak MW coordinate system is referred to as a Manhattan Frame (MF) [24]. We develop a novel method for detecting MF.

If three plane normals denoted as n_1, n_2, and n_3, are mutually orthogonal within a frame, then a MW coordinate system is established. If only two orthogonal plane normals exist within this frame, they form a weak MW coordinate system. In such cases, the third orthogonal plane normal, n_3, can be computed as: $n_3 = n_1 \times n_2 / ||n_1 \times n_2||$.

In situations where there are no orthogonal relationships between all planes, the most reliable lines are selected from all of the lines in this frame.

If three orthogonal spatial lines, denoted as v_1, v_2, and v_3, are detected, a MW coordinate system is established. In cases where only two orthogonal spatial lines, v_1 and v_2, are detected, a weak MW coordinate system is formed. The third orthogonal line, v_3, can be calculated using the cross product: $v_3 = v_1 \times v_2 / ||v_1 \times v_2||$. The process of establishing the MW coordinate system is depicted in Fig. 2.

D. Tracking with Structural Constraints

As mentioned in [24], there are two situations to be considered: MF and Non-MF.

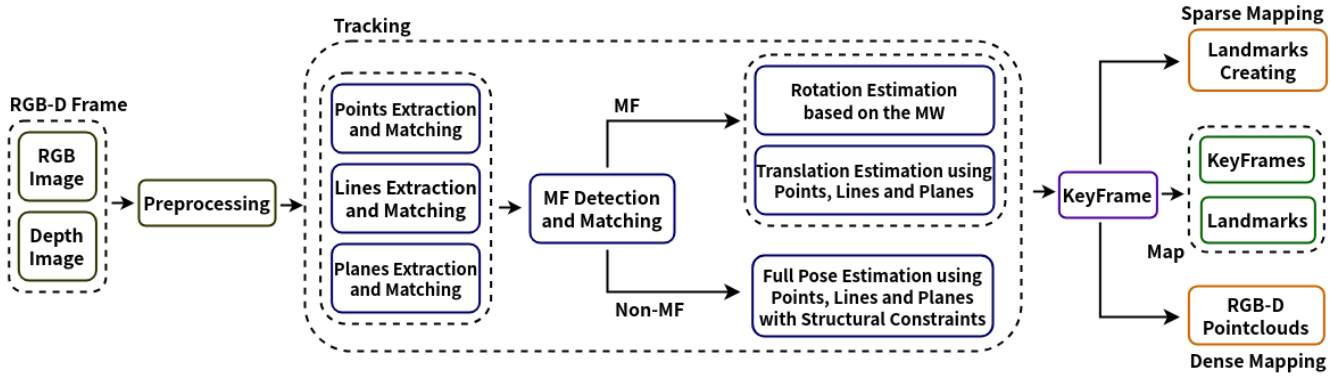

Fig. 1. Our STL-SLAM framework, which is structured-constrained and based on point, line and plane features.

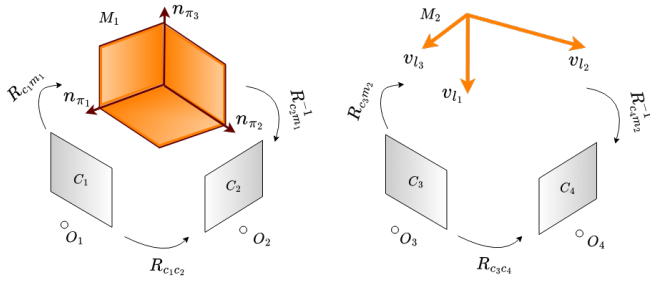

Fig. 2. Establishment of the MW coordinate system. Left: MW coordinate system established by plane normals. Right: MW coordinate system established by spatial lines.

1) Rotation Estimation for the MF: Instead of directly tracking the camera's movement from one frame to another, this estimation is achieved by modeling indoor environments as a MW, effectively mitigating the rotation drift that might occur in frame-to-frame tracking.

2) Translation Estimation for the MF: After obtaining the rotation matrix R^*_{wc}, a non-linear optimization method minimizes the re-projection errors of points, lines, and planes, ultimately obtaining the camera translation t_{cw}. We use P_w to represent the world coordinate of a map point, corresponding to the observed pixel point p_o. We define the operator φ, which converts non-homogeneous coordinate $\left[R^{*-1}_{wc}, t_{cw}\right]$ to homogeneous coordinate.

The point re-projection error e_p is defined as the difference between the observed position and the projected position of the map point:

$$e_p = ||p_o - \gamma(\varphi(\left[R^{*-1}_{wc}, t_{cw}\right])P_w)|| \tag{1}$$

The observed line in the image, denoted as I, corresponds to spatial lines with endpoints P_{wa} and P_{wb} in the world coordinate system. The line re-projection error e_l is calculated as the sum of the distances from the projected positions of the two endpoints to I:

$$e_l = ||I^T\gamma(\varphi(\left[R^{*-1}_{wc}, t_{cw}\right])P_{wa})|| + ||I^T\gamma(\varphi(\left[R^{*-1}_{wc}, t_{cw}\right])P_{wb})|| \tag{2}$$

To prevent over-parameterization, we define the operator ρ and q. The plane is expressed in the following form: $q(\pi) = (\rho(n)^T, d)^T$, where $\rho(n) = \phi = arctan(n_y/n_x)$, and $\psi = arcsin(n_z)^T$. In the camera coordinate system, the observed plane is denoted as π_o in Hessian form, with the world coordinate of the corresponding map plane expressed as π_w. The re-projection error of the plane is defined as the distance between the observed plane and the map plane:

$$e_\pi = ||q(\pi_o) - q(\varphi(\left[R^{*-1}_{wc}, t_{cw}\right])^{-T}\pi_w)|| \tag{3}$$

Minimize the cost function to solve for the optimal camera translation vector t^*_{cw}:

$$t^*_{cw} = \arg\min_{t_{cw}}(\sum(H_p(e_p) + H_l(e_l) + H_\pi(e_\pi))) \tag{4}$$

where H is the Huber robust kernel function that enhances the robustness of the optimization process.

3) Pose Estimation for the Non-MF: If no MF could be matched in the global map, a nonlinear optimization technique is utilized to concurrently estimate the 6-DoF transformation T_{cw}, which encompasses both rotation and translation components.

The point re-projection error e_p is defined as the difference between the observed position and the projected position of the map point:

$$e_p = ||p_o - \gamma(T_{cw}P_w)|| \tag{5}$$

where p_o is the observed point and P_w is the map point.

The line re-projection error e_l is calculated as the sum of the distances from the projected positions of the two endpoints to observed line I:

$$e_l = ||I^T\gamma(T_{cw}P_{wa})|| + ||I^T\gamma(T_{cw}P_{wb})|| \tag{6}$$

where P_{wa} and P_{wb} are the endpoints of spatial line corresponding to I.

The re-projection error of the plane is defined as the distance between the observed plane and the map plane:

$$e_\pi = ||q(\pi_o) - q(T^{-T}_{cw}\pi_w)|| \tag{7}$$

where π_o is the observed plane and π_w is the map plane.

We denote the direction vector of the spatial line as $v = (v_x, v_y, v_z)^T$. To prevent over-parameterization, similarly to the plane, we denote the line as $\rho(v) = (\phi = arctan(v_y/v_x), \psi = arcsin(v_z))^T$. Our method introduces

Fig. 3. Real-world environments. Top left: sequence-1. Top right: sequence-2. Bottom left: sequence-3. Bottom right: sequence-4.

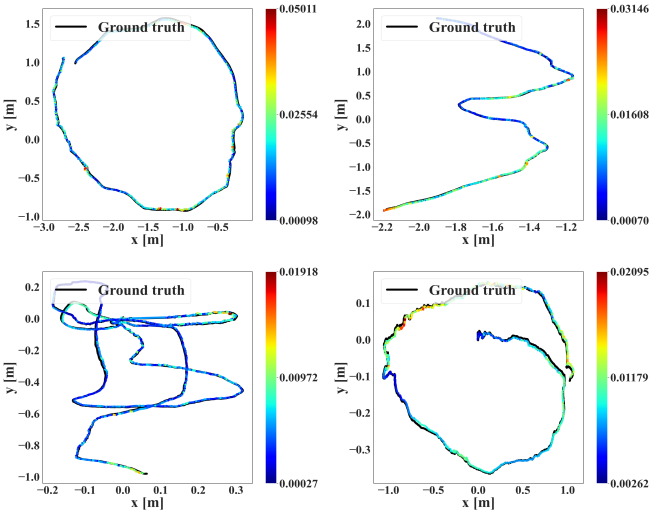

Fig. 4. Estimated trajectory (colored) and ground truth (black) in the fr3/cabinet, fr3/str_tex_near of TUM RGB-D dataset and living_room_kt0, living_room_kt3 of ICL-NUIM RGB-D dataset.

structural constraints of parallel lines and perpendicular lines, as well as parallel planes and perpendicular planes in the map.

Parallel and perpendicular lines are identified by the angle difference of the direction vectors. The errors are:

$$\begin{cases} e_{l\parallel} = ||\rho(v_o) - \rho(R_{cw} v_w)|| \\ e_{l\perp} = ||\rho(R_\perp v_o) - \rho(R_{cw} v_w)|| \end{cases} \tag{8}$$

where v_o is the observed line and v_w is the map line.

Likewise, we search for parallel and perpendicular planes within the map. The corresponding errors are expressed as:

$$\begin{cases} e_{\pi\parallel} = ||\rho(n_o) - \rho(R_{cw} n_w)|| \\ e_{\pi\perp} = ||\rho(R_\perp n_o) - \rho(R_{cw} n_w)|| \end{cases} \tag{9}$$

Here, n_o represents the observed plane normal, n_w represents the map plane normal, and $R_\perp$ denotes the rotation matrix for 90° angle.

Minimize the cost function to solve for the transformation matrix T^*_{cw}:

$$T^*_{cw} = \arg\min_{T_{cw}} \Big(\sum (H_p(e_p) + H_l(e_l) + H_{l\parallel}(e_{l\parallel}) + H_{l\perp}(e_{l\perp}) + H_\pi(e_\pi) + H_{\pi\parallel}(e_{\pi\parallel}) + H_{\pi\perp}(e_{\pi\perp})) \Big) \tag{10}$$

IV. Evaluation

All experiments are conducted using an Intel Core i5-10200U CPU @ 2.40GHz, without utilizing any GPU. We evaluate our systems in real-world environments, as well as on public datasets including the TUM dataset [26], and ICL dataset [27].

A. Real-world Environments

To evaluate our method in real-world scenarios, we create a dataset by capturing data with a handheld D435 camera at 30 FPS. This dataset comprises sequences recorded in typical indoor environments such as corridors and conference rooms, which are characterized by fewer distinctive point and line features.

The environments in which the dataset is captured are illustrated in Fig. 3. Similar to [28], our dataset does not provide ground truth. Each sequence in the dataset forms a closed loop. To assess the drift of different methods, we disable loop closure detection in SLAM and evaluate the algorithm's accuracy by measuring the Euclidean distance between the starting and ending positions.

We compare our method with other state-of-the-art approaches. The experimental results are shown in Table I,

TABLE I

Comparison of the accumulated drift (m) in real-world environments. × represents tracking failure.

Sequence	STL-SLAM	[24]	[7]	[15]	Total Frames
sequence-1	**0.71**	1.04	×	0.87	1076
sequence-2	**1.25**	1.89	×	2.02	2497
sequence-3	**0.83**	×	×	×	1246
sequence-4	**1.18**	1.62	3.43	1.22	1872
Average	**0.99**	1.38	3.43	1.37	1673

and the minimum error for each sequence is marked in bold. It can be found that STL-SLAM achieves the smallest cumulative drift compared to other approaches. Especially in sequence-3, for some frames of this sequence, there are cases where the rotation angle is large or the features are very sparse.

STL-SLAM achieves stable tracking by using drift-free rotation estimation based on plane and line MF detection and pose estimation based on structured features. In real-world environments, it processes frames (640×480) on average in 42ms.

B. TUM RGB-D Dataset

The TUM RGB-D dataset comprises sequences captured in various environments, making it valuable for SLAM re-

TABLE II

COMPARISON OF ATE RMSE (M) FOR TUM AND ICL-NUIM RGB-D SEQUENCES. × REPRESENTS TRACKING FAILURE.

		Methods							Frames	
Dataset	Sequence	STL-SLAM	[1]	[15]	[5]	[24]	[9]	[20]	MF	Total
TUM RGB-D	fr1/xyz	**0.010**	0.013	0.011	0.013	**0.010**	0.013	×	2	798
	fr1/desk	0.028	0.031	0.020	**0.016**	0.038	0.020	×	12	613
	fr3/cabinet	**0.019**	×	0.095	×	0.022	×	0.024	985	1111
	fr3/large_cabinet	**0.062**	×	0.072	×	0.108	0.088	0.107	200	983
	fr3/str_notex_far	0.033	×	0.035	×	0.042	0.053	**0.028**	686	793
	fr3/str_notex_near	**0.013**	×	0.016	×	0.023	×	0.020	795	1053
	fr3/str_tex_far	0.014	0.011	**0.009**	0.011	0.024	0.013	×	582	906
	fr3/str_tex_near	**0.011**	0.012	0.012	0.015	0.013	0.040	×	566	1056
ICL NUIM	living_room_kt0	**0.005**	×	0.006	×	0.008	0.012	0.007	1232	1510
	living_room_kt1	**0.010**	0.062	×	×	0.011	0.051	0.014	681	967
	living_room_kt2	**0.018**	0.204	0.024	0.027	0.019	0.022	0.025	770	882
	living_room_kt3	0.011	0.137	**0.009**	0.062	0.013	0.053	0.033	1041	1242
	office_room_kt0	0.033	0.121	0.035	0.081	0.026	**0.021**	0.036	1502	1510
	office_room_kt1	0.022	0.364	0.027	0.069	**0.017**	0.620	0.556	945	967
	office_room_kt2	**0.019**	0.614	0.024	0.033	0.020	0.027	0.023	788	882
	office_room_kt3	**0.011**	0.073	0.017	0.020	0.013	0.012	0.028	1207	1242

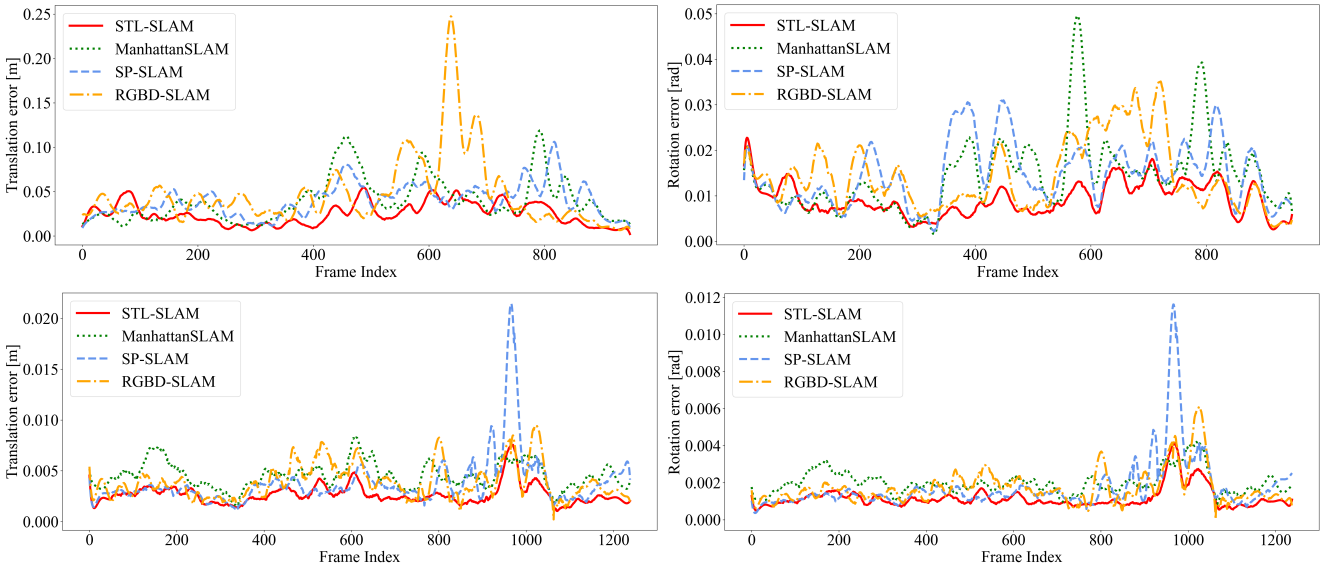

Fig. 5. Absolute translation error and rotation error in the fr3/large_cabinet (top) of TUM RGB-D dataset and office_room_kt3 (bottom) of ICL-NUIM RGB-D dataset for STL-SLAM and other algorithms.

search. We assess the method's performance across different environments, particularly in texture-limited scenarios such as 'fr3/cabinet' (with a single cabinet) and 'fr3/str_notex_far' (featuring a few planes and almost no texture), by comparing the root mean square error (RMSE) of the absolute trajectory error (ATE). The minimum error for each sequence is highlighted in bold in Table II.

The top-right section of Fig. 4 depicts our STL-SLAM trajectory in the 'fr3/str_tex_near' sequence, with the black curve representing the ground truth trajectory and the colored curve representing our tracking trajectory.

In 'fr1/xyz' and 'fr3/str_tex_near', there are adequate point features but insufficient structural features. Our method shows strong performance, outperforming the method used in [20], which requires frame-by-frame alignment with the MW coordinate system for pose estimation, resulting in tracking failures in cluttered scenes lacking clear planes.

In the 'fr3/cabinet' sequence, the scene is primarily composed of a single cabinet. Apart from a few corner points of the cabinet, there are hardly any other prominent point features in the environment. Line features are mainly detected along the edges of the cabinet, with a limited number in each frame. Methods that rely entirely on point features, such as [1] and [5], fail to track. Although [9] incorporates line feature optimization, it still experiences tracking failures, whereas our method consistently maintains errors within a small range. The trajectory of STL-SLAM on this sequence is depicted in the top-left section of Fig. 4.

The upper part of Fig. 5 illustrates the real-time translation and rotation errors of ManhattanSLAM [24], SP-SLAM [23], RGBD-SLAM [20], and STL-SLAM on the 'fr3/large/underline cabinet' sequence. As can be seen, STL-SLAM has the smallest average error in rotation and translation estimation.

C. ICL-NUIM RGB-D Dataset

The ICL-NUIM RGB-D dataset includes eight sequences recorded in indoor environments. Experimental results are illustrated in Table II.

As shown in the lower-left portion of Fig. 4, the absolute error peak of STL-SLAM in the 'living_room_kt0' sequence does not exceed 0.020. In the 'living_room_kt2' sequence, there are abundant features in the environment, including point, line, and plane features. Our method demonstrates competitive performance.

In certain segments of 'living_room_kt3', the scene from far to near fails to detect any planes, leading to obvious errors for SP-SLAM and RGBD-SLAM. Our method maintains errors below 0.011 in over half of the frames within this sequence, with the maximum less than 0.021, as illustrated in the lower-right portion of Fig. 4.

In the lower section of Fig. 5, the absolute translation and rotation error curves of several multi-feature fusion methods on the 'office_room_kt3' sequence are presented.

V. Conclusion

In this paper, we present a novel RGB-D SLAM method tailored for texture-limited indoor environments. By detecting and matching MF, we decouple rotation and translation estimation for MF, thereby reducing the accumulation of rotation errors. Additionally, we incorporate structural constraints on the parallelism and perpendicularity of lines and planes. Experimental results demonstrate that our proposed method ensures robust tracking in texture-limited indoor scenes, significantly enhancing pose estimation accuracy. Our method applies to scenarios with both limited texture and normal scenes. In the future, we will further explore the applicability of our method in outdoor texture-limited environments and environments with significant illumination changes, and consider extensions in multi-sensor fusion.

References

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multi-map slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] Y. Park and S. Bae, "Keeping less is more: Point sparsification for visual slam," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7936–7943.

[3] W. Chen, W. Li, A. Yang, and Y. Hu, "Active visual slam based on hierarchical reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7155–7162.

[4] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airvo: An illumination-robust point-line visual odometry," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3429–3436.

[5] Y. Wang, C. Jiang, and X. Chen, "Voom: Robust visual object odometry and mapping using hierarchical landmarks," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 298–10 304.

[6] P. Kim, B. Coltin, and H. J. Kim, "Linear rgb-d slam for planar environments," in *2018 European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 333–348.

[7] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[8] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," in *2008 European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 802–815.

[9] 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.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] Pumarola, Albert and Vakhitov, Alexander and Agudo, Antonio and Sanfeliu, Alberto and Moreno-Noguer, Francese, "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.

[12] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[13] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 980–990, 2008.

[14] H. Li, Z. Hu, and X. Chen, "Plp-slam: a visual slam method based on point-line-plane feature fusion," *Robotics*, vol. 39, no. 2, pp. 214–220, 2017.

[15] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2105–2112.

[16] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *1999 IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 1999, pp. 941–947.

[17] Kim, Pyojin and Coltin, Brian and Kim, Hyoun Jin, "Visual odometry with drift-free rotation estimation using indoor scene regularities," in *2017 British Machine Vision Conference (BMVC)*, vol. 2, no. 6, 2017, p. 7.

[18] L. Wang and Z. Wu, "Rgb-d slam with manhattan frame estimation using orientation relevance," *Sensors*, vol. 19, no. 5, p. 1050, 2019.

[19] R. Guo, K. Peng, W. Fan, Y. Zhai, and Y. Liu, "Rgb-d slam using point–plane constraints for indoor environments," *Sensors*, vol. 19, no. 12, p. 2721, 2019.

[20] 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. 11 581–11 587.

[21] Kim, Pyojin and Coltin, Brian and Kim, H Jin, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7247–7253.

[22] Kim, Pyojin and Coltin, Brian and Kim, H Jin, "Linear rgb-d slam for planar environments," in *2018 European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 333–348.

[23] 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.

[24] 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, 2021, pp. 6687–6693.

[25] 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.

[26] 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 (IROS)*. IEEE, 2012, pp. 573–580.

[27] 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.

[28] Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 3934–3942.