D-FUSION SLAM: A VSLAM SYSTEM OPTIMIZED BY INTEGRATING DEPTH INFORMATION AND RGBD FEATURE POINTS

Qingchun ZHENG [1,2], Moudong WU [1,2], Peihao ZHU [1,2,3], Bin YANG[1,2], Shubo LI [1,2]

This paper introduces D-Fusion SLAM, an advanced VSLAM system based on ORB-SLAM2. D-Fusion SLAM integrates depth information to enhance feature selection, aiming to improve accuracy and speed in visual SLAM applications. The system mainly comprises two modules: an image grid filtering module based on grayscale information and a feature selection module integrating depth information. These modules effectively eliminate redundant and useless features, ensuring high-quality features for pose estimation. We tested D-Fusion SLAM on the TUM dataset and compared it with ORB-SLAM2. Experimental results demonstrate that D-Fusion SLAM outperforms ORB-SLAM2, significantly enhancing the system's accuracy and robustness.

Keywords: Simultaneous Localization and Mapping, Depth Integration, RGB-D Camera, Feature Selection

1. Introduction

Visual SLAM (VSLAM) is a computer vision technology enabling real-time localization and environment mapping using image data from visual sensors, without external maps[1]. It is widely used in robotic autonomous navigation[2] and unmanned systems[3], providing them with autonomy and environmental perception[4]. Thus, improving the speed and accuracy of VSLAM systems for quick and precise environmental information acquisition is essential.

Two primary approaches enhance the VSLAM speed and accuracy[5]. The first is a hardware-based improvement, progressing from monocular to stereo and depth cameras, which allows for faster and more accurate environmental perception and map construction. Notable examples include the ORB-SLAM series by Mur-Artal R et al.: ORB-SLAM[6] for monocular cameras, ORB-SLAM2[7] for stereo and depth cameras, and ORB-SLAM3[8], which integrates multi-sensor information including LiDAR.

[1] Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, School of Mechanical Engineering, Tianjin University of Technology, Tianjin, 300384, China.

[2] National Demonstration Center for Experimental Mechanical and Electrical Engineering Education (Tianjin University of Technology).

[3] Corresponding author: Peihao ZHU. e-mail: zhupeihao_gp@163.com

The second approach to enhance VSLAM involves innovations in advanced geometric features based on environmental structures, such as points, lines, planes, and edges. ORB-SLAM and ORB-SLAM2[7] estimate camera poses using FAST corners and ORB features. Gomez-Ojeda et al.[9] integrate point and line features for pose estimation in low-texture environments, enhancing the SLAM robustness. Sun et al.[10] introduce the STING-SM method for plane matching, achieving complete 6-DoF camera pose estimation suitable for handheld cameras and mobile robots. Li et al.[11] utilize point and line features with depth cameras to improve field scene reconstruction in low-texture outdoor environments. Despite these advancements improving VSLAM accuracy, challenges like feature redundancy and computational speed remain. As shown in Fig. 1, after converting RGB images to grayscale, detected point and line features display a high level of redundancy.

To address the issues above, researchers have proposed a series of improvements. Zhang et al.[12] proposed a concise ray-to-ray residual model to replace the popular point-to-line model, enhancing SLAM accuracy and robustness through line feature optimization. Yu et al.[13] employed a hypothesis testing framework to resolve rotational ambiguities arising when matching vanishing directions with 3D directions, improving camera pose estimation accuracy. Zhang et al.[14] developed TTT SLAM, a feature-based bathymetric SLAM framework that extracts and matches terrain gradient features from submaps, enhancing robustness and efficiency. Yang et al.[15] proposed a unified multi-feature framework for the mutual association of point-line-plane features, integrating them to improve environmental information utilization, positioning accuracy, and robustness. Yuan et al.[16] calculated the uncertainty of 3D position estimates of map points in depth measurements within the ORB-SLAM2 framework, implementing selection strategies for keyframes and map points to achieve higher localization and mapping accuracy.

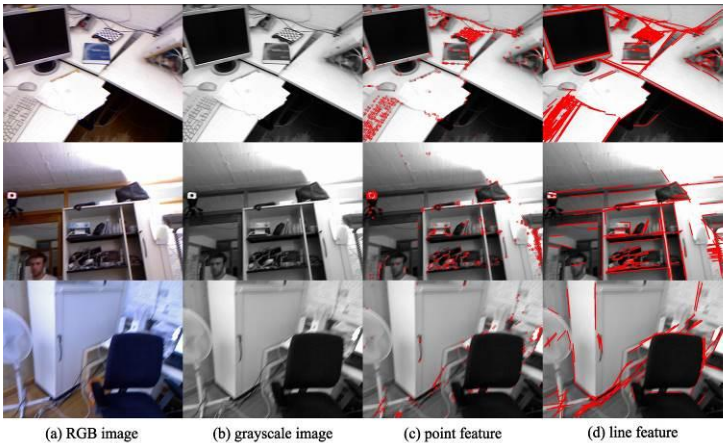

(a) RGB image (b) grayscale image (c) point feature (d) line feature

Fig. 1. Common point features and line features detection

This paper introduces D-Fusion SLAM, a VSLAM algorithm that enhances point feature processing by integrating depth information from a depth camera. This approach allows for explicit analysis of 3D positions of feature points in unknown environments, filtering out abnormal and redundant features. Our research significantly improves VSLAM accuracy, with key contributions as follows:

(1) Image Grid Filtering Module: We developed a grayscale-based image grid filtering module to obtain feature-rich images for detection.

(2) Feature Selection Module: Our feature selection module integrates depth information, converting 2D feature points into 3D spatial points and improving their quality.

(3) Experimental Validation: We validated D-Fusion SLAM's superiority in localization and mapping through experiments on the public TUM dataset, showing it outperforms the state-of-the-art ORB-SLAM2 system.

The paper is organized as follows: Section 2 details the D-Fusion SLAM framework and modules; Section 3 discusses experimental parameter settings and results; Section 4 concludes with future research directions.

2. System Overview

This paper presents the D-Fusion SLAM algorithm by introducing a feature selection module that integrates depth information. Incorporating depth information enables the system to accurately generate 3D point clouds from 2D images, enhancing spatial understanding. To address the increased computational demand of the feature selection module, we propose an image grid filtering module based on grayscale information, which preprocesses images by masking indistinct areas, significantly improving the computational efficiency of the VSLAM system. The overall system framework is illustrated in Fig. 2.

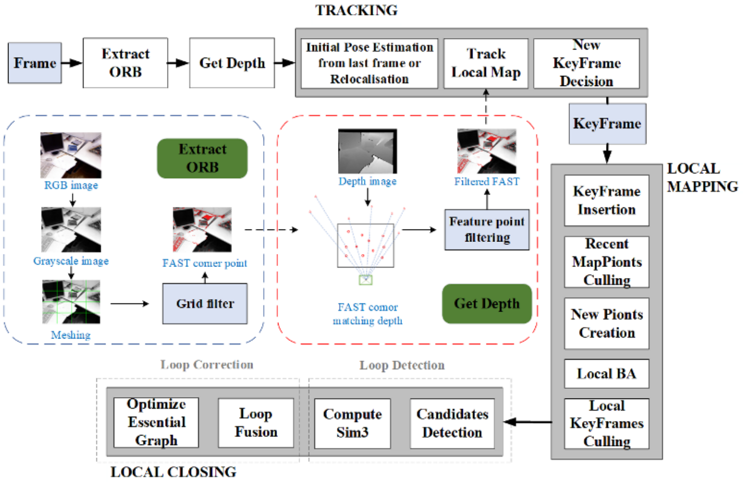


Fig. 2. System Overview Diagram

2.1 Grayscale Image Grid Filtering Module

As shown in Fig. 3, during VSLAM experiments, performing feature detection grid by grid can become computationally burdensome due to some grids lacking significant features. To address this issue, we propose an image grid filtering module based on grayscale analysis. By comparing the grayscale range and variance within each grid, this module improves computational efficiency. The calculation process is as follows:

1) **Image Initialization:** The RGB image input from the sensor is first converted into a grayscale image, and the grayscale value of each pixel is obtained.
2) **Grid Division:** Based on the size of the grayscale image, the image is divided into multiple small grids, and the grayscale values of pixels within each grid are obtained. The divided grids are denoted as D_k, D_{k+1}…, as shown in Fig. 3(a).

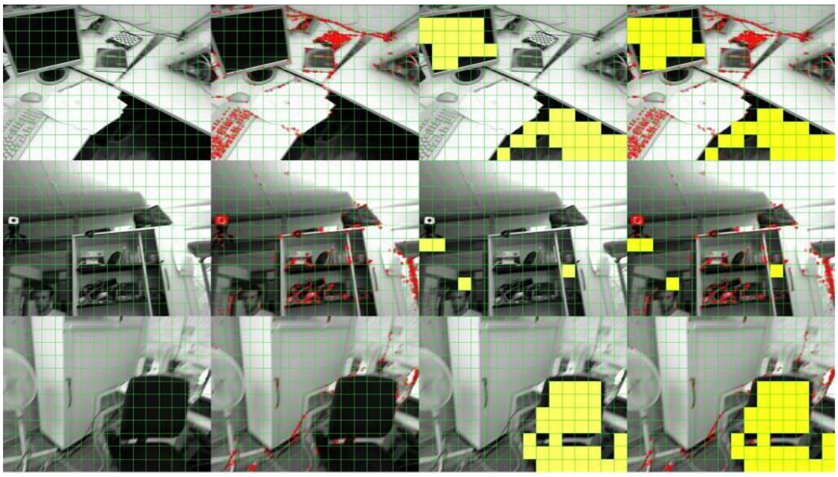

(a) grid diagram (b) detect feature (c) mask processing (d) re-detection

Fig. 3. Image Grid Filtering Module Execution Steps Diagram

3) **Grid Grayscale Analysis:** Use grayscale values to determine the overall feature significance of each grid. The specific steps are as follows:

Grayscale Range Calculation: Calculate the grayscale range of each grid, as shown in Equation (1). A more extensive grayscale range indicates the grid has strong contrast, meaning its features are significant.

$$Range_{\mathrm{k}} = \max(D_{\mathrm{k}}) - \min(D_{\mathrm{k}}) \tag{1}$$

where $Range_k$ is the grayscale range of grid k, and D_k represents the grayscale values of all pixels within grid k.

Grayscale Variance Calculation: Calculate the grayscale variance of each grid, as shown in Equation(2). Variance reflects the degree of variation in grayscale values within the image, with larger variance indicating more significant features within the grid.

$$Var_{\mathrm{k}} = \frac{1}{N_k} \sum_{i=1}^{N_k} (D_{i,k} - \overline{D_k})^2 \tag{2}$$

where Var$_k$ is the grayscale variance of grid k, N$_k$ is the number of pixels within grid k, D$_{i,k}$ is the grayscale value of the i-th pixel within grid k, and $\overline{\mathrm{D}}$ is the mean grayscale value of grid k.

4) **Grid Filtering**: Filter the grids based on the results of the grayscale analysis. The pseudocode is shown in *Table 1*. If the grayscale range of a grid is less than the RangeThreshold and the grayscale variance is less than the VarThreshold, the grid is deemed unnecessary for feature detection. The determination result *w* is expressed as:

$$w_k = \begin{cases} false, Range_k < \ RangeThreshold \text{ and } Var_k < \ VarThreshold \\ true, otherwise \end{cases} \tag{3}$$

where w$_k$ indicates whether grid k needs feature detection, with true representing yes and false representing no.

Table 1

Algorithm for Grid filter model

Input	Feature analysis of the gray grid Range$_k$ and Var$_k$
Output	Filter results for gray grid w
1:	for i←1 to m do
2:	w = True
3:	if (Range$_k$ < RangeThreshold)&&(Var$_k$ < VarThreshold)
4:	then w = False
5:	end
6:	Return w

5) Masking Process: Apply masking to the grids that do not require feature detection (similar to the method for handling potential dynamic points in dynamic environments). After processing, proceed with feature detection, as shown in Fig. 3(c). For grids that require feature detection, proceed with standard feature detection, as shown in Fig. 4.

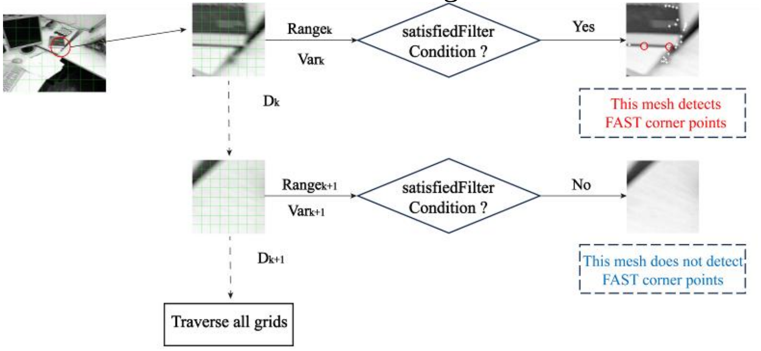

Rangek
Vark
satisfiedFilter Condition ?
Yes
This mesh detects FAST corner points
Dk
Rangek+1
Vark+1
satisfiedFilter Condition ?
No
This mesh does not detect FAST corner points
Dk+1
Traverse all grids

Fig. 4. Image grid filtering module decision framework diagram

The effect of the image grid filtering module is shown in Fig. 3(d), significantly improving computational efficiency in feature detection. The values of RangeThreshold and VarThreshold affect image preprocessing: large values fail to remove insignificant grids, while small values may filter out grids with valid features. The selection of appropriate values will be discussed in Section 3.2.

2.3 Depth Filtering Module

The feature detection process is straightforward, but not all feature points are suitable for pose estimation and matching. Two exceptional cases include Useless Feature Points and Redundant Feature Points, as shown in Fig. 1. Using high-quality features is essential for effective matching and pose estimation.

In the baseline system, we use an RGB-D camera as the external sensor to acquire color and accurate depth information from the surrounding environment. The depth filtering module, as shown in Fig. 5, processes the depth map to obtain distance information for each feature point. It then eliminates useless and redundant feature points, supplying high-quality features for tasks such as feature matching and pose estimation in the tracking thread.

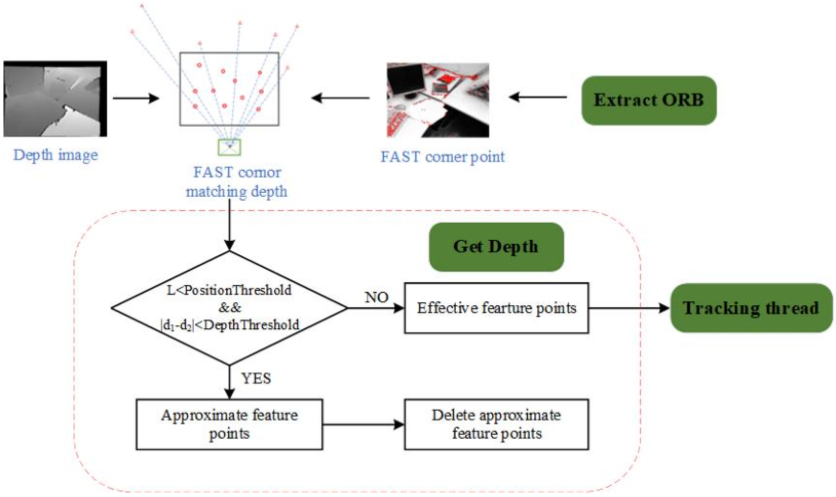

Depth image
FAST comor matching depth
FAST comer point
Extract ORB
Get Depth
L<PositionThreshold && |d₁-d₂|<DepthThreshold
NO
YES
Effective fearture points
Tracking thread
Approximate feature points
Delete approximate feature points

Fig. 5. Adopting Deep Information and Camera Pose Tracking

First, we obtain the depth information of feature points based on their coordinates in the image coordinate system, as shown in Fig. 6. To eliminate useless feature points, the depth information *d* must meet the detection range conditions of the RGB-D camera. The detection range constraint is as follows:

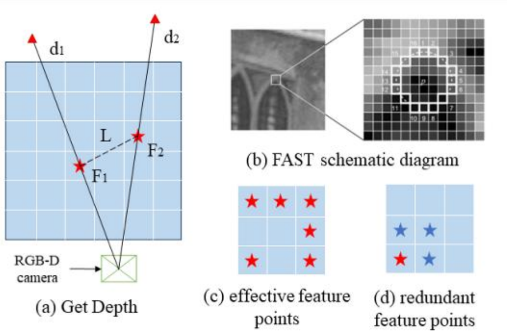
$$d_{\min} < d < d_{\max} \tag{4}$$

where $\mathrm{d_{min}}$ and $\mathrm{d_{max}}$ are the RGB-D camera's minimum and maximum detection distances, respectively. If the depth information d of a feature point is not within this range, the feature point is considered useless and is eliminated.

To address the redundant feature points shown in Figure 1, t we calculate the distance between feature points using their 2D-pixel coordinates in the camera coordinate system. As shown in Fig. 6(a), the distance *L* between feature point F_1 and feature point F_2 is calculated as follows:

$$L = \sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2} \tag{5}$$

where (u_1,v_1) and (u_2,v_2) are the 2D pixel coordinates of feature points F_1 and F_2 in the camera coordinate system, respectively.



(a) Get Depth

(b) FAST schematic diagram

(c) effective feature points

(d) redundant feature points

Fig. 6. Deep Filtering Principle Diagram

If the distance L between feature points F_1 and F_2 is sufficiently small, and their depth information d_1 and d_2 are nearly identical, they can be considered redundant feature points. To quantify redundant feature points, we define the distance threshold PositionThreshold and depth threshold DepthThreshold as follows:

1.PositionThreshold: When L is less than this threshold, the distance between feature points F_1 and F_2 is considered sufficiently small.

2.DepthThreshold: When $|d_1\text{-}d_2|$ is less than this threshold, the depth information of feature points F_1 and F_2 is considered nearly identical.

Based on the above thresholds, feature points F_1 and F_2 can be considered redundant if the following conditions are met:

$$L < PositionThreshold \,\&\&\, |\, d_1 - d_2 \,| < DepthThreshold \tag{6}$$

The values of the PositionThreshold and DepthThreshold can be further determined based on the type of feature points and the system's application scenario. To avoid deleting valid feature points near the detection boundary, we retain those points while only removing redundant feature points around the pixel p, as shown in Fig. 6. Therefore, the PositionThreshold is set to 1.5, i.e.:

$$PositionThreshold = 1.5 \tag{7}$$

ORB-SLAM2 applications include UAV navigation, robot navigation, autonomous driving, and industrial automation, typically requiring sub-meter accuracy. In SLAM systems, the depth accuracy of RGB-D cameras is often in meters, so the DepthThreshold is set to 0.1 meters, i.e.:

$$DepthThreshold = 0.1\text{m} \tag{8}$$

This setting ensures that the depth filtering module effectively removes redundant feature points with slight differences in depth information, enhancing the system's accuracy and robustness of pose estimation and feature matching.

3. Experiments and Discussion

In this section, we validated the D-Fusion SLAM system's effectiveness through experiments on a computer with a 2.30GHz Intel Xeon(R) Gold 5118 CPU, Quadro P4000 GPU, and 64GB of memory. We introduced the TUM dataset, which includes various indoor scenes and motion patterns, serving as a benchmark for SLAM evaluation. We analyzed the impact of the grayscale threshold (RangeThreshold) and variance threshold (VarThreshold) in the grid filtering module on pose estimation, determining their optimal values to enhance feature detection efficiency and accuracy. Finally, we compared our D-Fusion SLAM system with the baseline ORB-SLAM2 framework using the Absolute Trajectory Error (ATE) metric to measure the difference between estimated and ground truth trajectories.

3.1 Dataset Introduction

The TUM RGB-D dataset is a new benchmark for evaluating SLAM systems and is widely used for testing and validation in indoor scenes. Six handheld SLAM scene video sequences were selected for this experiment, as shown in *Table 2*.

Table 2

Datasets

Video Sequences	Content and Function
freiburg1_360（fre1_360）	It performed a 360-degree rotation within an office environment to assess the system's robustness against rotational motion.
freiburg1_floor（fre1_floor）	Scanned the wooden floor of the office to validate the system's capability to detect texture features.
freiburg1_desk（fre1_desk）	Scanned the four desks in the office to assess the system's reliability in handling translational motion.
freiburg1_room（fre1_room）	It records the office scene to test the SLAM system's loop closure detection capability. The video lasts 48.9 seconds and features relatively fast motion.
freiburg2_desk（fre2_desk）	It rotates around a desk with a video duration of 99.36 seconds and moves relatively slowly.
freiburg3_long_office_household（fre3_long）	Captured a loop-closure sequence featuring rich textures and intricate structures within the scene to validate the system's performance in complex environments.

The TUM dataset provides ground truth camera motion trajectories captured at 100Hz, essential for evaluating the ATE of SLAM systems. Testing D-Fusion SLAM across these six scenes enables a comprehensive assessment of its performance in diverse environments.

3.2 The influence of the grey value threshold and variance threshold on pose estimation

In this section, we analyze the impact of RangeThreshold and VarThreshold on the tracking time and pose estimation accuracy of D-Fusion SLAM using grid filtering. With the minimum threshold for detecting FAST corners set to minThFAST = 7, we set the RangeThreshold to 14 to ensure that the grayscale range within the grid meets the conditions for detecting FAST corners.

$$RangeThreshold = 14 \tag{9}$$

Next, we will experiment with VarThreshold ranging from 1 to 10, using tracking time and Absolute Pose Error (APE) as evaluation metrics. We selected the fre1_room and fre2_desk video sequences for analysis. To ensure reliable results, we conducted five experiments on each sequence and averaged the outcomes. The data obtained include time$_i$, mean$_i$, and rmse$_i$, representing the average tracking time, mean pose error, and root mean square error (RMSE) of pose estimation at VarThreshold=i. Next, we will proceed with a detailed analysis.

The time$_i$ As shown in Fig. 7, in the fre1_room, the time$_i$ decreases only when VarThreshold $\in \{1,2\}$. As VarThreshold increases, the time$_i$ fluctuates upward due to the need for continued feature point detection in many grids after filtering. Conversely, in the fre2_desk experiments, increasing VarThreshold consistently reduces the tracking time, improving the system's real-time performance. In summary, higher variance thresholds may increase tracking time in the fast-moving scenario of fre1_room, while in the slower-moving scenario of fre2_desk, they can significantly decrease tracking time.

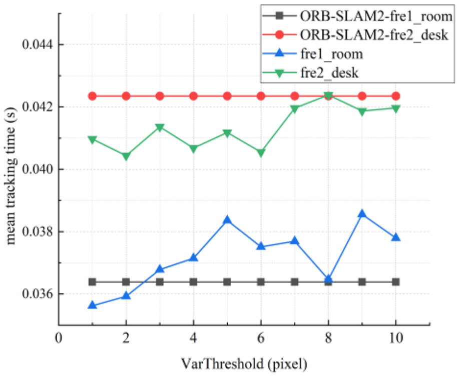
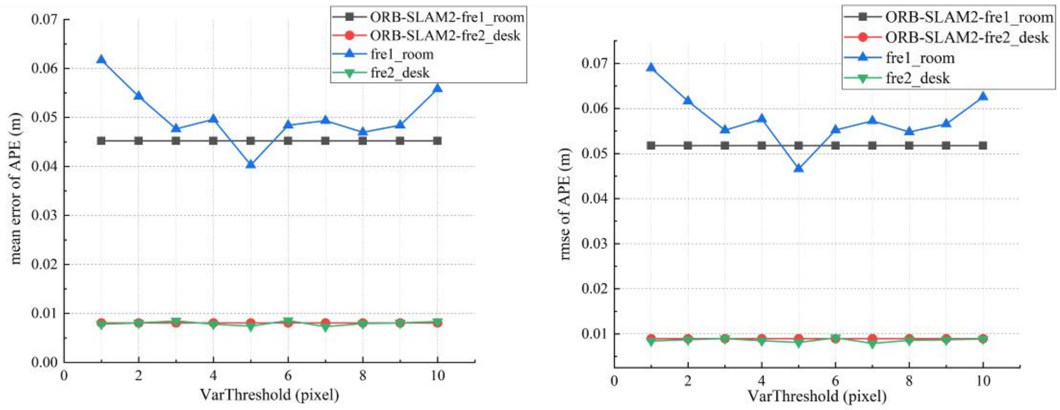


Fig. 7. Mean tracking time

The mean$_i$ and rmse$_i$ From Fig. 8(a) and Fig. 8(b), we observe that in the fre1_room video sequence, the RMSE of pose estimation initially decreases and then improves as VarThreshold rises. Filtering out feature points with uniform grayscale and less distinctive features reduces the error, with the minimum error achieved at VarThreshold=5. However, further increases lead to significant features being filtered out, making camera pose estimation more susceptible to noise and reducing accuracy. In the fre2_desk video sequence, the mean error and RMSE show little change with increasing VarThreshold due to slower camera motion, which reduces disparities between adjacent frames and facilitates feature matching. Notably, at VarThreshold=5 or 7, both mean error and RMSE decrease. In conclusion, setting VarThreshold to 5 is optimal for improving the accuracy of the VSLAM algorithm.

$$\mathrm{Var}Threshold = 5 \tag{10}$$



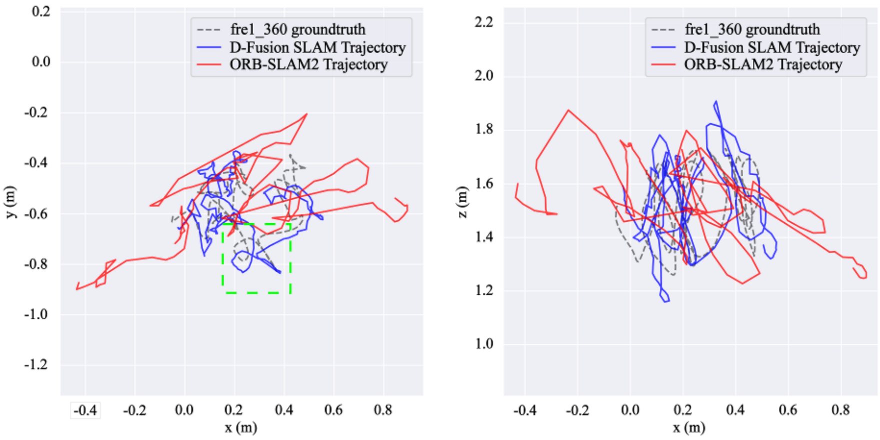
(a) mean error (b) rmse

Fig. 8. Trajectory Error Comparison Chart

3.3 Experiments

Firstly, the parameters RangeThreshold and VarThreshold are introduced into the D-Fusion SLAM system and compared with ORB-SLAM2. Using the TUM dataset examples fre1_360 and fre1_floor, we present the 3D trajectory visual results. Fig. 9 and Fig. 10 display the projected camera motion trajectories in the x-y and x-z planes for D-Fusion SLAM (blue lines) and ORB-SLAM2 (red lines), alongside the ground truth (grey dashed lines). Next, we analyze the errors of the predicted trajectories compared to the ground truth, including translational (xyz) and rotational (rpy) components. Detailed results are shown in Fig. 11 and Fig. 12.

Analysis of Spatial Trajectory in fre1_360 In Fig. 9, D-Fusion SLAM shows a trajectory closer to the ground truth than ORB-SLAM2, demonstrating higher accuracy and stability. Notably, in the green-highlighted area of Fig. 9(a), D-Fusion SLAM's predicted trajectory closely aligns with the ground truth, despite minor deviations.

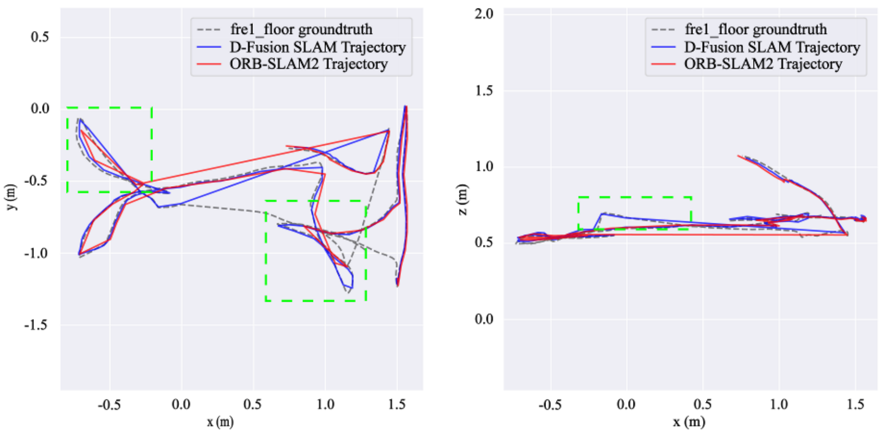


(a) x-y plane

(b) x-z plane

Fig. 9. Predicted Spatial Trajectory Projection of fre1_360

Analysis of Spatial Trajectory for fre1_floor As shown in Fig. 10, both D-Fusion SLAM and ORB-SLAM2 trajectories are near the ground truth, with overlaps indicating similar errors. However, in the green-highlighted area of Fig. 10(a), D-Fusion SLAM is more accurate. Additionally, Fig. 10(b) shows that D-Fusion SLAM can track the ground truth where ORB-SLAM2 fails, indicating greater robustness in this sequence.

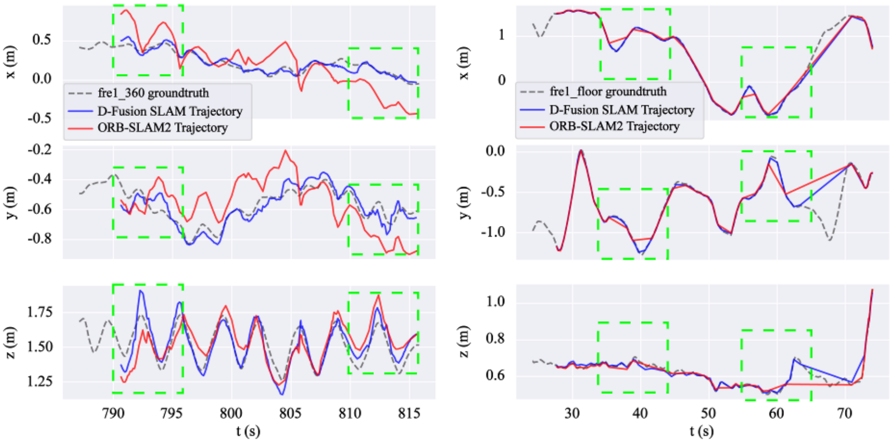


(a) x-y plane

(b) x-z plane

Fig. 10. Predicted Spatial Trajectory Projection of fre1_floor

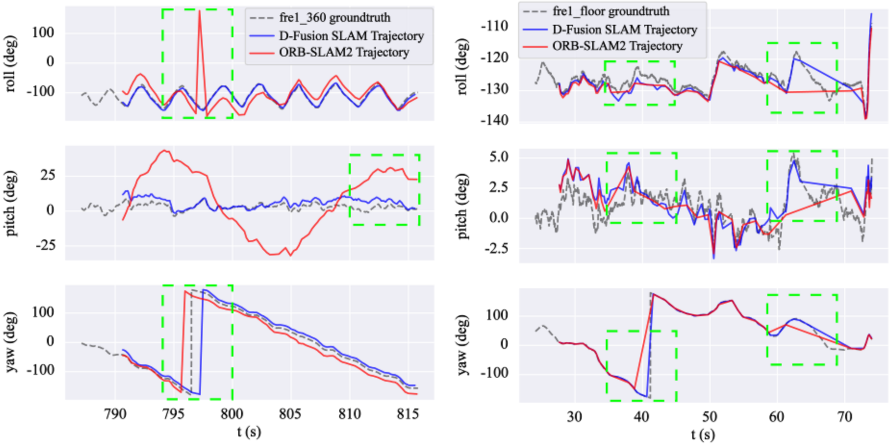
Analysis of Translational Component XYZ Errors Fig. 11 shows the errors in the world coordinate system along the xyz axes for the discussed spatial trajectories. Panels (a) and (b) correspond to fre1_360 and fre1-floor, respectively. In Fig. 11(a), D-Fusion SLAM's translational errors are closer to the ground truth in all directions, particularly in the highlighted green section, where it demonstrates reduced errors and improved localization. In Fig. 11(b), while most errors overlap between D-Fusion SLAM and ORB-SLAM2, the non-overlapping region (green boxed area) shows D-Fusion SLAM's trajectory variation closely matching the ground truth, indicating greater precision and consistency in this scenario.



(a) fre1_360 (b) fre1_floor

Fig. 11. Translational Component XYZ Error

Rotation component (rpy) error analysis Fig. 12 shows the rotational errors of the trajectories in the world coordinate system, including roll, pitch, and yaw, with panels (a) and (b) for fre1_360 and fre1_floor, respectively. In Fig. 12(a), the roll and pitch angles of D-Fusion SLAM closely overlap with the ground truth, while ORB-SLAM2 exhibits significant errors, especially in the highlighted green region where discrepancies are more pronounced. In Fig. 12(b), D-Fusion SLAM's trajectory, though not perfectly aligned with the ground truth, is closer than that of ORB-SLAM2, displaying smaller errors and higher accuracy, particularly in the green-highlighted area.



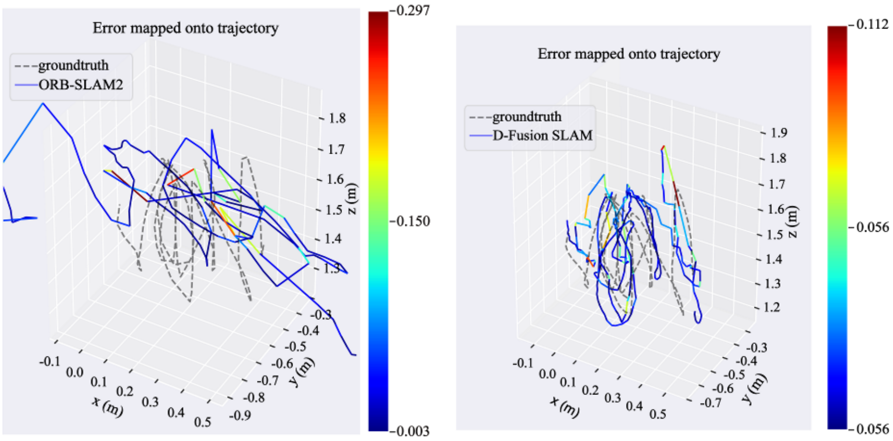
(a) fre1_360 (b) fre1_floor

Fig. 12. Rotational Component RPY Error

To further demonstrate D-Fusion SLAM's superiority, we provide visual results comparing errors. Fig. 13 and Fig. 14 show the relative pose error (RPE) between the trajectories from the ORB-SLAM2 and D-Fusion SLAM against the ground truth trajectories for the fre1_360 and fre1_floor sequences. Fig. 15 and Fig. 16 analyze RPE metrics, including mean error, median error, RMSE, and standard deviation (std), confirming D-Fusion SLAM's advantages in accuracy and stability.

In Fig. 13 and Fig. 14, the blue lines represent trajectories, while the grey dashed lines indicate ground truth. The colored vertical bars show error magnitudes at corresponding timestamps, with red indicating larger errors and deeper blue indicating smaller errors. In Fig. 15 and Fig. 16, black lines represent RPE where smaller values indicate more accurate predictions and more fluctuations suggest better alignment with ground truth timestamps.

RPE for fre1_360 In Fig. 13(a), ORB-SLAM2's maximum error is 0.297 meters, while D-Fusion SLAM's maximum is 0.112 meters, reflecting over a 50% reduction. Both systems have minimum errors, but D-Fusion SLAM's trajectories are overall closer to ground truth compared to ORB-SLAM2's more scattered trajectories.

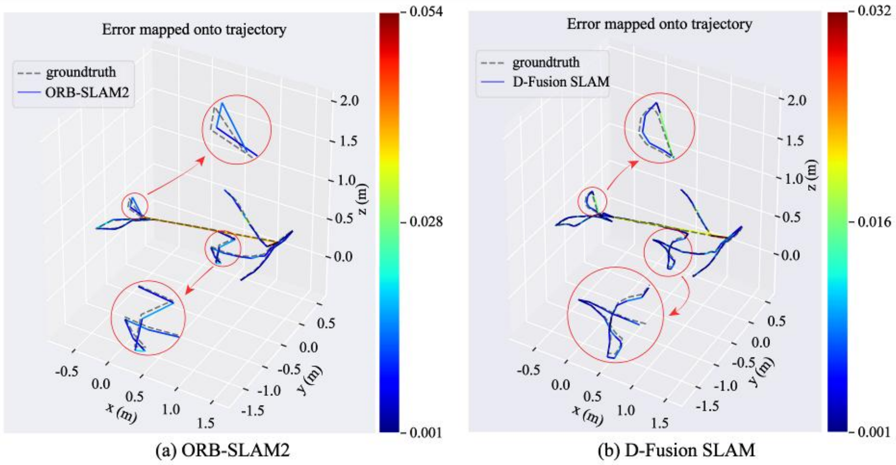


(a) ORB-SLAM2

(b) D-Fusion SLAM

Fig. 13. Predicted Trajectory Comparison of fre1_360

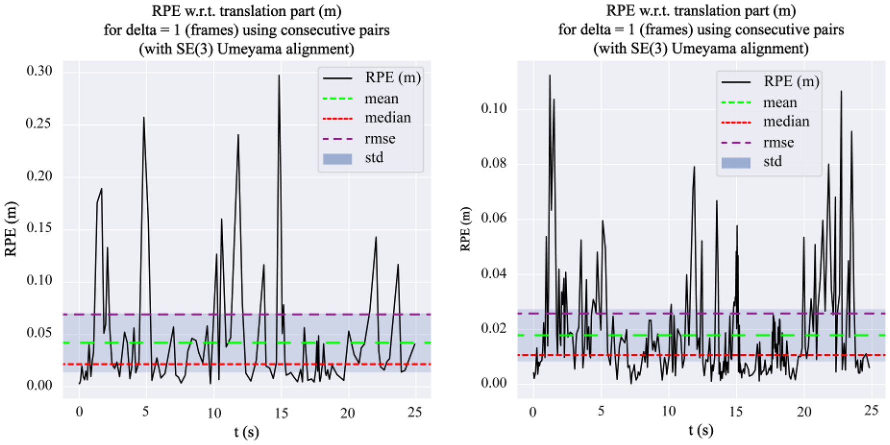
RPE for fre1_floor In Fig. 14, ORB-SLAM2 shows a maximum error of 0.054 meters versus D-Fusion SLAM's 0.032 meters, with both systems having a minimum error of 0.001 meters. D-Fusion SLAM captures more timestamps and camera poses, demonstrating its advantage in environmental information retrieval.



(a) ORB-SLAM2 (b) D-Fusion SLAM

Fig. 14. Predicted Trajectory Comparison of fre1_floor

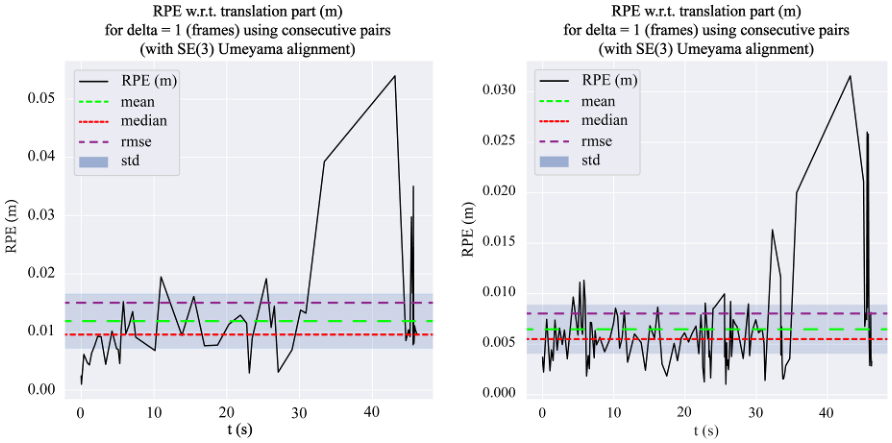
RPE analysis of fre1_360 In Fig. 15(a), ORB-SLAM2's maximum RPE is about 0.30 meters, while D-Fusion SLAM's is about 0.11 meters, with significantly lower average and median RPE values for D-Fusion SLAM. D-Fusion SLAM matches 262 timestamps compared to ORB-SLAM2's 141, indicating better environmental information capture.



(a) ORB-SLAM2 (b) D-Fusion SLAM

Fig. 15. RPE Analysis of fre1_360

RPE analysis of fre1_floor In Fig. 16(a), ORB-SLAM2's maximum RPE exceeds 0.05 meters, while D-Fusion SLAM's is around 0.03 meters, also showing lower average and median RPE values. D-Fusion SLAM matches 135 timestamps against ORB-SLAM2's 54, further highlighting its capability to capture more environmental information, explaining the trajectory differences observed in Fig. 14.



(a) ORB-SLAM2

(b) D-Fusion SLAM

Fig. 16. RPE Analysis of fre1_floor

D-Fusion SLAM exhibits smaller RPE than ORB-SLAM2, indicating higher accuracy in predicting trajectories closer to the ground truth. Additionally, D-Fusion SLAM shows more frequent error fluctuations, allowing it to match more timestamps and gather environmental information, thus demonstrating greater robustness. We provide quantitative evaluation metrics, including average tracking time and maximum, minimum, mean, and RMSE of RPE. Each video sequence was run ten times, and average values were calculated to minimize the impact of system uncertainties. The overall results are presented in *Table 3*.

Table 3

RPE

Sequence		fre1_360	fre1_floor	fre1_desk	fre1_room	fre2_desk	fre3_long
ORB-SLAM2	Time(s)	0.0297741	0.02748794	0.03718386	0.03612249	0.04055405	0.04524937
	Max(m)	0.7518778	0.119234	0.1143055	0.2095349	0.0454511	0.0429191
	Min(m)	0.3736492	0.0289177	0.0186178	0.0278146	0.0118293	0.0059256
	Mean(m)	0.5065064	0.064056	0.0473497	0.0909519	0.0267483	0.0185017
	Rmse(m)	0.5177966	0.0671173	0.0512749	0.1005869	0.0274813	0.0197656
Ours	Time(s)	0.0347389	0.0320578	0.03741437	0.03971084	0.04359422	0.04558183
	Max(m)	0.4622209	0.1082471	0.0897817	0.1920585	0.0435841	0.040376
	Min(m)	0.1809041	0.0165013	0.0128381	0.0220597	0.0109587	0.0058269
	Mean(m)	0.270993	0.0593027	0.0357424	0.0898052	0.0251579	0.0178062
	Rmse(m)	0.2779717	0.063894	0.0390499	0.0965865	0.0257948	0.0188213
Improvement of our approach against ORB-SLAM2	Time(s)	-16.67%	-16.63%	-0.62%	-9.93%	-7.5%	-0.73%
	Max(m)	**38.52%**	**9.21%**	**21.45%**	**8.34%**	**4.11%**	**5.93%**
	Min(m)	**51.58%**	**42.94%**	**31.04%**	**20.69%**	**7.36%**	**1.67%**
	Mean(m)	**46.50%**	**7.42%**	**24.51%**	**1.26%**	**5.95%**	**3.76%**
	Rmse(m)	**46.32%**	**4.80%**	**23.84%**	**3.98%**	**6.14%**	**4.78%**

In the comparison experiments across six handheld video sequences, the average tracking time of D-Fusion SLAM increased by 8.32% compared to ORB-SLAM2. This increase is due to the addition of a depth filtering module alongside grayscale image grid filtering, enabling the system to gather more environmental information and increasing overall computation time. We also computed the accuracy improvement of D-Fusion SLAM relative to ORB-SLAM2, as shown in *Table 3*. The efficiency improvement calculation follows the formula:

$$\mathrm{Im\,p} = \left| \frac{\mathrm{error}_{D-\mathrm{Fusion}-SLAM}}{\mathrm{error}_{ORB-SLAM\,2}} - 1 \right| \times 100\% \tag{11}$$

As shown in Fig. 17, D-Fusion SLAM improved pose estimation accuracy compared to ORB-SLAM2 across six video sequences, with the most significant improvement in fre1_360, exceeding 45%. Other sequences show varying improvements: approximately 24% for fre1_desk, around 7% for fre1_floor and fre2_desk, while fre1_room and fre3_long exhibit slight increases, reflecting a gradient improvement trend. A detailed analysis of the six sequences revealed differences in camera movement speed and angular deviation. Specifically, fre1_360 has the fastest camera movement, followed by fre1_desk, with fre1_floor and fre2_desk at average speeds, while fre1_room and fre3_long are relatively slower. D-Fusion SLAM effectively enhances pose estimation accuracy during rapid movements by using grayscale image grid filtering and depth filtering modules to eliminate redundant and irrelevant feature points, capturing more valuable environmental information and reducing trajectory errors.

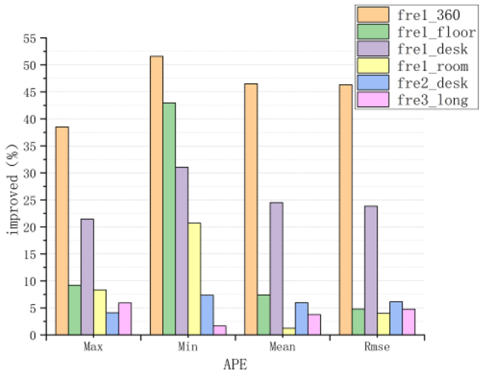


Fig. 17. Accuracy Improvement Diagram

4. Conclusions

This paper proposes a visual SLAM system optimized for feature detection called D-Fusion SLAM. This method utilizes depth information from depth maps to filter detected feature points, eliminating redundant features. We designed an image preprocessing module based on grayscale variations to highlight the image's most prominent and meaningful regions. Unlike existing feature optimization methods in SLAM, D-Fusion SLAM performs a comprehensive analysis and processing of feature points by combining pixel distances in 2D images with depth information in 3D space. This approach reduces computational costs while improving localization accuracy. Experiments on the TUM public dataset demonstrate that D-Fusion SLAM achieves higher localization accuracy and robustness, particularly in challenging scenarios such as rapid camera movements and changes in camera direction. Although the feature optimization process slightly increases tracking time, it does not compromise the overall real-time performance of the SLAM system.

D-Fusion SLAM also has the potential for further development into multi-feature fusion SLAM or visual-LiDAR fusion SLAM, enabling richer environmental and structural information acquisition to enhance localization accuracy and adaptability in complex environments. In future work, we will extend D-Fusion SLAM to incorporate higher-level geometric features, such as line and plane features, to apply to dynamic environments for environmental reconstruction and real-time localization.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62073239).

REFERENCES

[1] *C. Cadena, L. Carlone, H. Carrillo, Y. Latif, and D. Scaramuzza, et al.*, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age", IEEE Transactions on Robotics, **vol.32**, no.6, Dec. 2016, pp. 1309-1332.

[2] *A. Nüchter, and J. Hertzberg*, "Towards semantic maps for mobile robots", Robotics and autonomous systems, **vol.56**, no.11, Nov. 2008, pp. 915-926.

[3] *W. Chen, G. Shang, A. Ji, C. Zhou, and X. Wang, et al.*, "An Overview on Visual SLAM: From Tradition to Semantic", Remote Sensing, **vol.14**, no.13, Jun. 2022, pp. 3010.

[4] *H. Taheri, and Z. C. Xia*, "SLAM; definition and evolution", Engineering applications of artificial intelligence, **vol.97**, Jan. 2021, pp. 104032.

[5] *A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel*, "A Comprehensive Survey of Visual SLAM Algorithms", Robotics, **vol.11**, no.1, Feb. 2022, pp. 24.

[6] *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, Aug. 2015, pp. 1147-1163.

[7] *R. Mur-Artal, and J. D. Tardos*, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras", IEEE Transactions on Robotics, **vol.33**, no.5, Jun. 2017, pp. 1255-1262.

[8] *C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos*, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM", IEEE Transactions on Robotics, **vol.37**, no.6, May 2021, pp. 1874-1890.

[9] *R. Gomez-Ojeda, F. Moreno, D. Zuniga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez,* "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments", IEEE Transactions on Robotics, **vol.35**, no.3, Apr. 2019, pp. 734-746.

[10] *Q. Sun, J. Yuan, X. Zhang, and F. Sun*, "RGB-D SLAM in Indoor Environments With STING-Based Plane Feature Extraction", IEEE/ASME Transactions on Mechatronics, **vol.23**, no.3, Nov. 2018, pp. 1071-1082.

[11] *Q. Li, X. Wang, T. Wu, and H. Yang*, "Point-line feature fusion based field real-time RGB-D SLAM", Computers & Graphics, **vol.107**, Oct. 2022, pp. 10-19.

[12] *C. Zhang, Z. Fang, X. Luo, and W. Liu*, "Accurate and robust visual SLAM with a novel ray-to-ray line measurement model", Image and Vision Computing, **vol.140**, Dec. 2023, pp. 104837.

[13] *H. Yu, W. Zhen, W. Yang, and S. Scherer*, "Line-Based 2-D-3-D Registration and Camera Localization in Structured Environments", IEEE transactions on instrumentation and measurement, **vol.69**, no.11, Jun. 2020, pp. 8962-8972.

[14] *Q. Zhang, and J. Kim*, "TTT SLAM: A feature-based bathymetric SLAM framework", Ocean Engineering, **vol.294**, Feb. 2024, pp. 116777.

[15] *H. Yang, J. Yuan, Y. Gao, X. Sun, and X. Zhang*, "UPLP-SLAM: Unified point-line-plane feature fusion for RGB-D visual SLAM", Information Fusion, **vol.96**, Aug. 2023, pp. 51-65.

[16] *J. Yuan, S. Zhu, K. Tang, and Q. Sun*, "ORB-TEDM: An RGB-D SLAM Approach Fusing ORB Triangulation Estimates and Depth Measurements", IEEE Transactions on Instrumentation and Measurement, **vol.71**, Feb. 2022, pp. 1-15.