Received 13 September 2024, accepted 13 October 2024, date of publication 16 October 2024, date of current version 29 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3481420

RESEARCH ARTICLE

Visual SLAM Based on Improved Line Filtering Decision and Weight Optimization

YIBO CAO, ZEHAO LUO, AND ZHENYU DENG
School of Software, South China Normal University, Guangzhou, Guangdong 510631, China

Corresponding author: Yibo Cao (422705879@qq.com)

This work was supported in part by the Scientific Research Innovation Project of Graduate School of South China Normal University, and in part by the Key Technology Research (Second Batch) Program of Ningbo.

ABSTRACT In recent years, visual SLAM (Simultaneous Localization and Mapping) based on line feature tracking has garnered widespread attention due to its provision of additional constraints for structured scenes. However, the current mainstream framework, PL-VINS, faces several challenges, such as overly simplistic line length pruning strategies and the utilization of fixed loss functions in point-line backend optimization. To address the former, we propose a novel line-length pruning strategy that dynamically determines pruning thresholds based on the average length of lines extracted from the current frame image. Regarding the latter, we introduce the concept of point-line weighting, which involves dynamically adjusting the size of the loss function based on the ratio of points to lines within a sliding window. Experimental results on public benchmark datasets demonstrate that, compared to the PL-VINS method, our approach achieves a 6.79% average improvement solely by employing the enhanced line length pruning strategy. Furthermore, by simultaneously adopting the improved line length pruning strategy and dynamic point-line weighting for backend optimization, our method outperforms the PL-VINS method with an average improvement of 23.60%. This indicates that our proposed enhancements elevate the accuracy of SLAM.

INDEX TERMS Localization, SLAM, visual camera sensors.

I. INTRODUCTION

In the fast-paced domain of robotics today, simultaneous localization and mapping (SLAM) stands as a fundamental challenge crucial for enabling autonomous navigation in mobile robots. At its core, SLAM empowers robots to autonomously traverse and construct maps in unfamiliar surroundings, relying on accurate environmental sensing and dependable self-localization [1], [2], [3]. Beyond its importance in scientific inquiry, SLAM carries extensive practical implications across industries such as manufacturing, healthcare, and defense, promising significant advancements in the realm of autonomous systems.

SLAM systems are primarily categorized into three types based on the type of sensors used: laser-based SLAM, visual-based SLAM, and multi-sensor fusion SLAM. Among these, visual SLAM (VSLAM) has garnered significant attention and research interest from both academia and industry due to its simple structure, low cost, strong recognition capabilities, and ability to capture rich texture information [4].

In the field of VSLAM, it can be further divided into three categories: feature-based visual SLAM, direct method visual SLAM, and deep learning-based visual SLAM. For feature-based visual SLAM, such as PTAM [5], VINS-MONO [6], and PL-VINS [7], they feature fast feature point extraction, making them suitable for environments with rich textures and real-time requirements. However, they may suffer from tracking degradation in environments with low texture or textureless regions. For direct-method visual SLAM, such as DSO [8], LDSO [9], and DSM [10], they directly use pixel values for tracking without the need for feature extraction, making them suitable for environments with weak textures and low-light conditions. However, they tend to be sensitive to lighting changes and image noise, and they require substantial computational resources. For deep learning-based visual SLAM, such as DeepVO [11]

The associate editor coordinating the review of this manuscript and approving it for publication was Adamu Murtala Zungeru.

and CNN-SLAM [12], they improve adaptability to complex scenes and changes but require training on large-scale datasets and may produce inexplicable results in some scenarios.

Although VSLAM technology has made significant progress, it still faces several challenges in practical applications. Feature-based visual SLAM relies on point features to estimate camera trajectories and build environment maps. However, in complex environments with weak textures or drastic lighting changes, the performance of these systems is often limited [13], [14]. For example, in indoor environments with uniformly colored walls or suboptimal lighting conditions, point features may not provide enough information to support precise positioning and map construction. In addition, rapid movement or occlusion in dynamic environments can lead to performance degradation. To overcome these challenges, researchers have begun to explore the use of line features in VSLAM. This is because line features provide more structural information and provide a more intuitive description of the geometry of the environment. Compared to point features, line features are more robust in complex environments.

In this paper, we propose an improved VSLAM method to enhance the performance and robustness of the system in complex environments through adaptive line extraction and point and line weight optimization. First, we adopt an adaptive thresholding strategy to optimize line feature extraction for different lighting and texture conditions. Second, we introduce a weight optimization mechanism to dynamically adjust the matching weights of the features according to the distribution number of features and the environmental structure information, so as to improve the positioning accuracy and the stability of map construction. The contributions of this paper are as follows:

- Removing non-dominant lines from the image reduces the number of lines, thus improving the performance and pose estimation of VSLAM.
- According to the point and line states of the local environment, dynamically adjust the optimization weights to improve accuracy of VSLAM in complex environments.
- Experiments on the benchmark dataset EuRoc [15] demonstrate that our method outperforms the PL-VINS method.

In the rest of the paper, Section II presents related work, and Section III describes the architecture of our proposed systematic approach. Section IV discusses our improved adaptive line length rejection strategy and point-line weight optimization. Section V describes the experimental setup and experimental results. Finally, we summarize our findings and future work in Section VI.

II. RELATED WORK

In the research progression of feature-based VSLAM, the development of feature extraction and optimization methods has been the core driving force behind technological advancement. Early works primarily focused on the extraction and matching of point features, such as Scale-Invariant Feature Transform (SIFT) [16], Speeded Up Robust Features (SURF) [17], Harris corner detection [18], and others. However, these methods exhibited poor performance in handling weak textures or illumination changes. To overcome these limitations, the Oriented FAST and Rotated BRIEF (ORB) algorithm [19] was proposed. It enhances the speed and robustness of feature extraction by combining FAST keypoint detection with BRIEF descriptors. Although corner detection techniques have matured significantly, challenges such as difficulty in acquiring corners and tracking loss still persist, especially in structured environments.

To address these issues, some studies have proposed methods that integrate both point and line features. References [20] and [21] introduced line constraints into stereo cameras. Reference [22] optimized RGB-D vision using point, line, and plane features. Reference [23] utilized the LoFTR network to generate dense point matching feature descriptors in low-texture scenes, filtering unreliable features using feature masking, and enhancing matching robustness using KNN strategies. References [24], [25], and [26] all enhanced localization by merging or suppressing unstable line segments, reducing the number of short lines to ensure the stability of line features during tracking. Reference [7] improved the real-time performance of the Line Segment Detector (LSD) algorithm for pose estimation problems and added additional constraints on scene structure. Reference [27] proposed adaptive combinations of point, line, and plane features for environmental adaptation, aiming to enhance tracking and mapping performance in scenes with varying texture richness. However, current research focuses on the quality of lines while neglecting the quantity issue. This paper introduces a novel adaptive line-filtering strategy to maintain an appropriate number of lines in the image.

Recently, some studies have started to focus on adaptive feature extraction and back-end optimization. For example, Zhou et al. [25] and Yu et al. [28] proposed an improved point-line feature VSLAM method. By adopting adaptive thresholds for point features and data association strategies, they enhanced the adaptability and robustness of the system in different environments. These works provide valuable insights, indicating that significant improvements in VSLAM system performance can be achieved by adaptively adjusting feature extraction strategies and optimizing back-end processing workflows.

Building upon previous research, this work further explores the application of adaptive line extraction and weight optimization in VSLAM. The goal is to provide a more reliable and efficient visual perception solution for the autonomous navigation of mobile robots in complex environments.

III. SYSTEM OVERVIEW

In this section, we will briefly outline our system. Our system is built upon the PL-VINS architecture and incorporates

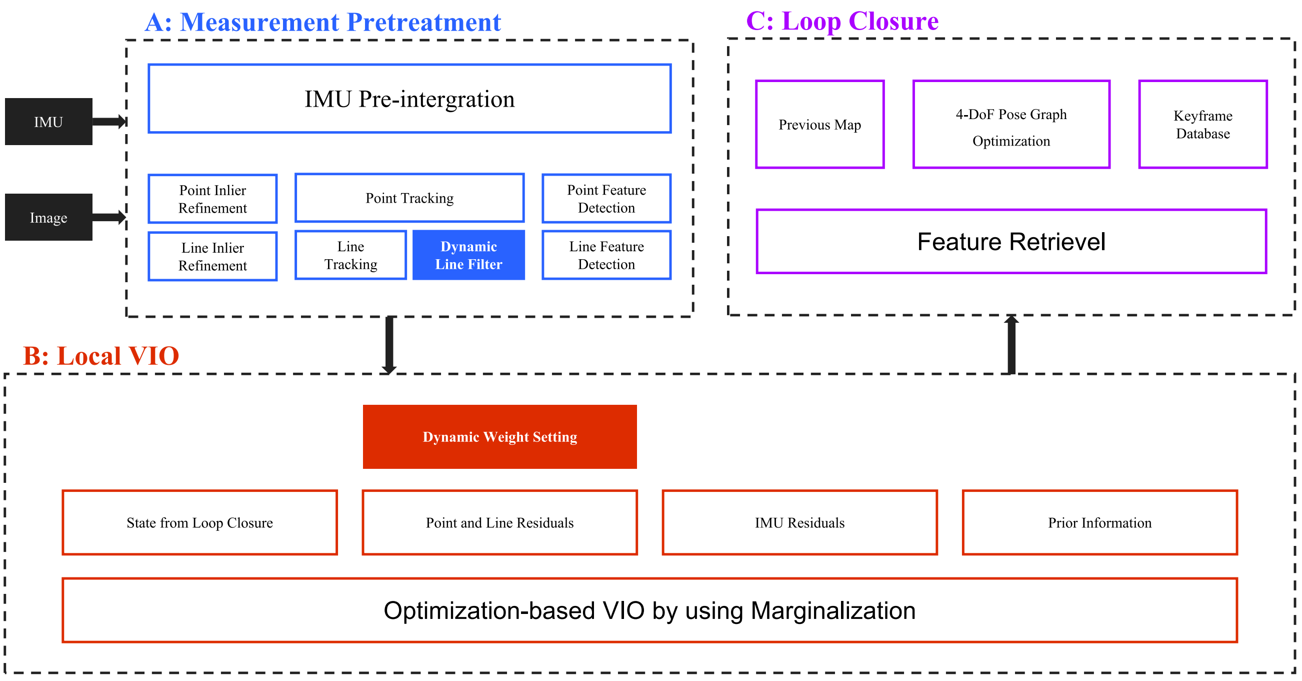


FIGURE 1. System Overview: Our system was developed based on the PL-VINS architecture with three threads: measurement preprocessing, local VIO (Visual Inertial Ranging) and closed loop. The highlighted parts are our improvement points.

dynamic line length trimming thresholds and dynamic point-line weighted back-end optimization techniques. This enhancement enables line features to play a more effective role in feature tracking and local map construction. The overall algorithm workflow is depicted in Figure 1. The system consists of three threads: measurement preprocessing, local VIO (Visual-Inertial Odometry), and loop closure.

A. IMAGE INPUT

Whenever an input image is received, the system performs point and line feature extraction. For point extraction, we use the Shi-Tomasi corner detector, which is an enhanced version of the Harris corner detection algorithm. This detector accurately identifies corners in the image and has strong noise resistance. For line segment extraction, we fine-tune the hidden parameters of the line segment detector to quickly extract line segments. We use a "wide-in, narrow-out" strategy to extract all lines in the image. Subsequently, we calculate the average length of line segments in the current image and dynamically determine the pruning threshold for line segments. This aspect will be further elaborated in Section IV.

B. SLIDING WINDOW OPTIMIZATION

After system initialization, a fixed-size sliding window is utilized to maintain a local map. The sliding window offers advantages in rapid optimization of key vectors such as local stability, pose, velocity, spatial points and lines, acceleration, and gyroscope bias. These key vectors are jointly optimized by multiple residual functions, as shown in Equation (9). The size of the loss function is dynamically adjusted based on the states of points and lines within the sliding window. This aspect will be discussed in detail in Section IV. Additionally, the sliding window determines whether a frame is a keyframe, and if so, it is passed to the loop closure thread.

C. LOOP CLOSURE

After local VIO, if the parallax between the current frame and the previous keyframe exceeds a predefined threshold or if the number of tracked features drops below a certain level, the current frame undergoes keyframe determination. Once a keyframe is selected, it activates the loop closure thread to search for potential loop closures. Upon detecting a loop closure, global pose optimization is performed using least squares functions, as shown in Equation (11). Our experiments, as well as those of PL-VINS, have verified that loop closure detection significantly improves the accuracy of SLAM.

IV. IMPROVED LINE METHOD

A. DYNAMIC REJECTION THRESHOLD

The current commonly used methods for line extraction are LSD and EDLines. However, they are primarily designed for scene representation rather than pose estimation. Therefore, they still have some potential for improvement in SLAM.

We notice that the improvements based on line extraction methods ([24], [25], [26]) mainly focus on merging and filtering short lines, rather than considering the number of lines. Although PL-VINS mentions extracting dominant lines as features because they may reappear in the nearest frame images, the issue of extracting an appropriate number of lines remains unresolved. We believe that too many lines can lead

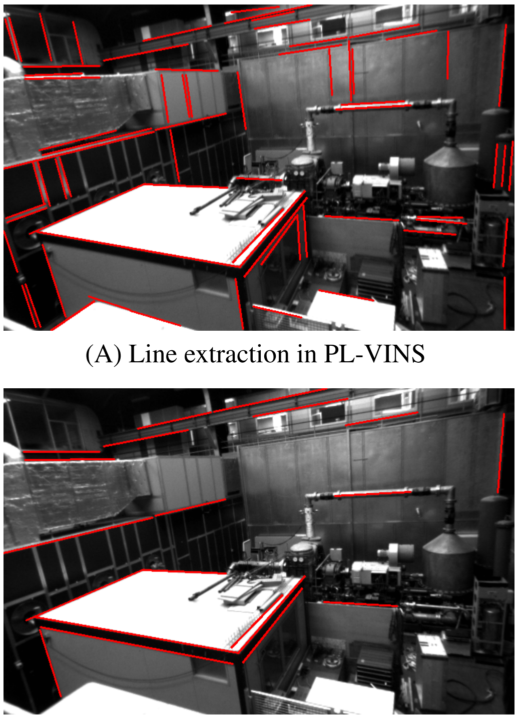

(A) Line extraction in PL-VINS
(B) Our improved line extraction

FIGURE 2. (A) describes the unimproved line extraction process, and (B) illustrates the enhanced line extraction process. Images are from the MH-04 sequence dataset. It can be clearly seen that our method results in a reduction in the number of lines.

to a significant number of matching errors, thereby reducing algorithm performance and localization accuracy. Therefore, based on the concept of dominant lines, when high-quality lines can be extracted from the image, the rejection threshold for lines should be increased to reduce their quantity.

In PL-VINS, the line length rejection strategy is oversimplified, calculated as follows:

$$L_{min} = \lceil \mu * \min(W_I, H_I) \rceil \tag{1}$$

Among them, L_{min} represents the minimum length of the line segment to be retained, $\min(W_I, H_I)$ represents the smaller value between the width W_I and height W_H of the input image, $\lceil \star \rceil$ represents the ceiling operation, and μ is a ratio factor.

For this formula, we have made two considerations: First, in structured environments where lines are clearly visible, after obtaining a sufficient number of dominant lines, this formula may retain non-dominant lines in the image, leading to an excessive number of lines. And in unstructured environments where lines are not clearly visible, this formula may result in extracting too few lines or even fail to extract any lines at all. Therefore, we propose a new dynamic line length elimination strategy, the experimental results of which are shown in Figure 2, and the formula is as follows:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)} \tag{2}$$

$$L_{min} = \lceil \mu * \frac{1}{n}\sum_{i=1}^{n} d_i \rceil \tag{3}$$

where $\lceil \star \rceil$ represents the ceiling operation, d denotes the Euclidean distance between two points, and μ is a scaling factor. In this context, x_i and y_i represent the coordinates of corresponding points on two line segments, with n being the number of points or dimensions considered in the distance calculation. During extensive experiments, we set μ to 1.25, considering generalizability.

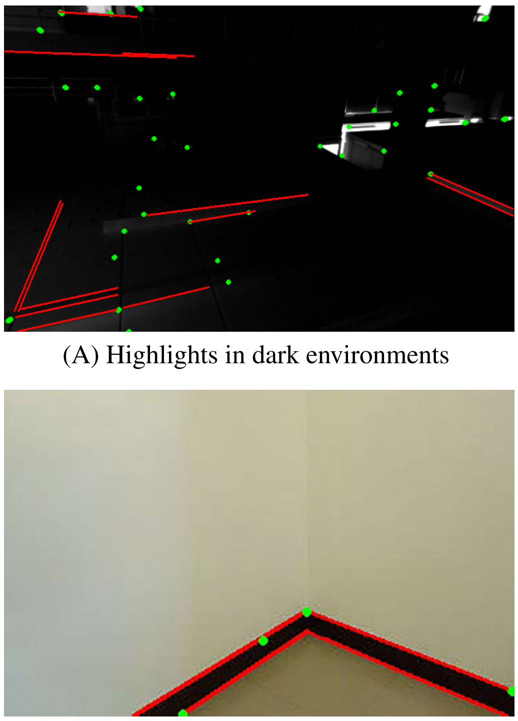

(A) Highlights in dark environments
(B) Structured and well-lit images

FIGURE 3. Both scenarios (A) and (B) in complex environments result in the extraction of too few points and lines by the SLAM system.

B. ADAPTIVE LOSS FUNCTION SIZE

We observe an interesting phenomenon: the number of dots and lines in an image decreases only in three cases. The first case is when there are no bright points in the dark image. We do not discuss this case because, in this case, VSLAM cannot extract the features of points and lines from the image. The second case is when there are bright parts of the image in a dark environment. In this case, accurate corner points can be obtained near the bright light, while line extraction becomes less accurate. The third case is that accurate line features can be extracted from bright structural images, while point features are difficult to obtain. This is illustrated in Figure 3.

We believe that in the third scenario, where accurate line extraction is feasible, the constraint imposed by line features would enhance the accuracy of the SLAM system. Therefore, it is advisable to increase the optimization weight for lines and decrease it for points. However, in the second scenario, although lines can still be extracted, they might be unstable compared to point features. Hence, it is preferable to increase the optimization weight for points as they provide more reliable constraints.

For the second and third scenarios, we distinguish them by using the local grayscale value to reflect the global grayscale value. We calculate the average grayscale value of the central circle in the image to represent the brightness level of the image. The formula is as follows:

$$\bar{P} = \frac{\sum_{(x,y)\in R} \omega(x,y)}{E} \tag{4}$$

where $\bar{P}$ represents the local grayscale value of the image, $\omega(x, y)$ represents the grayscale value of pixels, R represents the circle centered at the pixel in the image, and E represents the total number of pixels in the circle.

Inspired by [25], which used the number of points and lines in the current frame to set optimization weights, our approach takes a different direction. We believe that relying solely on the point and line count of a single frame lacks robustness. Instead, we use a sliding window approach that considers the number of points and lines across multiple recent frames. This method captures the most recent image data, providing a more accurate reflection of the current complex environment. The optimization weights are dynamically adjusted based on the aggregated data within the sliding window.

Additionally, by considering the resolution of different cameras, we can establish a normal range for the number of feature points in typical environments. If the actual feature point count falls below this range, it suggests the environment is either structured or poorly lit. Our algorithm then distinguishes between these scenarios using grayscale values and dynamically adjusts the optimization weights for points and lines accordingly. The corresponding formula is shown below.

$$m_i = \beta * (\frac{n_i}{\bar{n}_i + \bar{n}_j}) \tag{5}$$

$$m_j = \beta * (\frac{n_j}{\bar{n}_i + \bar{n}_j}) \tag{6}$$

where m_i represents the loss function value of points, m_j represents the loss function value of lines, β is the initial coefficient, which is usually set to 0.6 for generalization, n_i and n_j represent the number of points and lines in the current sliding window, and $\bar{n}_i$ and $\bar{n}_i$ are the averages of the points and lines in normal environment. The overall flowchart is shown in Figure 4.

We optimize using line reprojection residuals based on point-to-line distances. First, the projection line l is obtained by transforming L_c to the image plane. Then, assuming l_w represents the j-th spatial line ζ which is observed by the i-th camera frame c_i, the line reprojection error can be defined as:

$$l = [l_1, l_2, l_3]^T \tag{7}$$

$$r_L(z_{\zeta_j}^{c_i}, x) = d(m, l) = \frac{m^T l}{\sqrt{l_1^2 + l_2^2}} \in \mathbb{R} \tag{8}$$

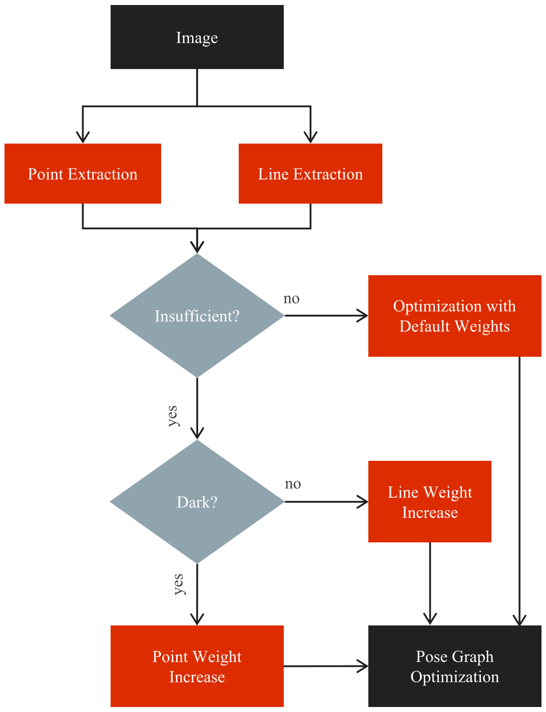


FIGURE 4. Flowchart of Adaptive Loss Function Based on Dual Estimation for Assessing Current Environment Conditions. Upon input of image data and completion of point and line extraction, if the number of points and lines is insufficient, a local grayscale assessment of the current environment is conducted. Finally, dynamic adjustment of weight optimization is performed. If the number of points and lines is normal, default weight optimization is carried out.

where $d(m, l)$ represents the distance function from a point to a line, and $\underline{m}$ represents the homogeneous coordinates of points in the line element. The corresponding Jacobian matrix can be obtained through the chain rule [29].

We perform local optimization within a sliding window, defining the full state vector X associated with point, line, and IMU measurement information in the sliding window.

$$X = [x_o, x_1, \ldots, x_{n_k}, \lambda_0, \lambda_1, \ldots, \lambda_{n_p}, o_0, o_1, \ldots, o_{n_l}] \tag{9}$$

$$x_k = [P_{b_k^w}, Q_{b_k^w}, V_{b_k^w}, b_a, b_g], \quad k \in [0, n_k] \tag{10}$$

The full state vector X is defined as follows: x_k includes the following components: the position $P_{b_k^w}$, orientation $Q_{b_k^w}$, velocity $V_{b_k^w}$, accelerometer bias b_a, and gyroscope bias b_g of the k-th IMU body. n_k, n_p, and n_l represent the total number of keyframes, spatial points, and lines in the sliding window, respectively. λ is the inverse distance of the point feature to its first observed keyframe. $o^T = (\theta^T, \theta)$represents the four-parameter orthogonal representation of the 3D line.

Next, we define the objective function for local optimization:

$$min(e_{prior} + e_{imu} + e_{point} + e_{line} + e_{loop}) \tag{11}$$

where e_{prior} represents the prior information obtained when marginalizing old frames in the sliding window, e_{imu} and e_{point} represent the residual of IMU measurements and point reprojection, respectively, e_{line} and e_{loop} represent the residual of line reprojection and loop closure constraints. As for the definition of e_{line}, it can be defined as:

$$e_{line} = \sum_{(i,j)\in\zeta} \left(p \left\| r_{\zeta}(z_{\zeta_j}^{c_i}), x \right\|^2_{\Sigma_{\zeta_j}^{c_i}}\right) \tag{12}$$

$$p(s) = \begin{cases} s, & s \leq 1 \\ 2\sqrt{s} - 1, & s > 1 \end{cases} \tag{13}$$

where $p(s)$ represents the Huber norm, used to suppress outliers.

TABLE 1. Comparison of Line Suppression Strategy and PL-VINS for Mean Line Length, Std Dev, and ATE.

Sequence	Mean Line Length		Line Std Dev		ATE	
	PL-VINS	**Our**	**PL-VINS**	**Our**	**PL-VINS**	**Our**
MH-01	98.611	**170.681**	46.420	**53.790**	0.198	**0.197**
MH-02	98.857	**175.978**	48.773	**58.374**	0.094	**0.092**
MH-03	99.989	**168.801**	46.845	**55.462**	0.116	**0.114**
MH-04	108.103	**190.375**	69.379	**90.596**	**0.181**	0.187
MH-05	108.996	**188.048**	67.138	**86.974**	0.325	**0.287**
V1-01	103.951	**178.156**	51.669	**62.718**	0.154	**0.151**
V1-02	107.621	**174.198**	51.989	**55.959**	0.082	**0.081**
V1-03	110.427	**179.432**	57.574	**62.208**	0.160	**0.136**
V2-01	105.306	**162.606**	45.831	**46.664**	0.087	**0.081**
V2-02	107.997	**183.191**	60.027	**72.517**	**0.110**	0.112
V2-03	118.162	**191.872**	72.171	**82.286**	0.287	**0.223**
Mean	106.183	**178.485**	56.165	**66.141**	0.162	**0.151(↓6%)**

V. EXPERIMENTS

In this section, we evaluate the experimental performance on the EuRoc benchmark dataset in terms of localization accuracy and real-time performance. The experiments are implemented on Ubuntu 20.04 and ROS Noetic. All experiments are conducted on an Intel Core i5-8300h CPU @2.30GHz. We compared the performance of all sequences in the EuRoc dataset, evaluating the absolute trajectory error (ATE). The formula for calculating Root Mean Square Error (RMSE) is as follows:

$$RMSE(T) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \breve{y}_i)^2}{n}} \tag{14}$$

where y_i and $\breve{y}_i$ represent the true pose and estimated pose of the mobile robot at time i, respectively.

A. REJECTION STRATEGY

Table 1 compares the Absolute Trajectory Error (ATE) of our method with PL-VINS, with bold values indicating better results. As shown in the table, the line length suppression strategy is crucial. When there are many long lines in the image, we prioritize retaining long lines while removing short ones; when there are fewer long lines, we retain short lines as much as possible. This adaptive thresholding strategy results in a higher mean line length and standard deviation compared to PL-VINS, thereby improving localization accuracy.

In terms of Absolute Trajectory Error, although PL-VINS performs better on certain datasets, the difference from our method is minimal. Notably, on the more challenging dataset V2-03, our method improves the loop closure test results by 22.30%.

Finally, from the overall average, we can see that the mean line length and standard deviation of our adaptive line strategy are higher than those of PL-VINS, resulting in a 6.79% improvement in Absolute Trajectory Error. This clearly demonstrates the importance of filtering out non-dominant lines.

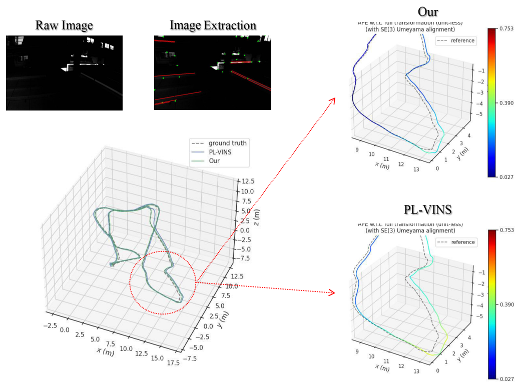


FIGURE 5. Low-light Environment of The MH-05 Sequence. In the red-highlighted portion, the environment corresponds to a dark setting with bright spots. The "Raw Image" images depict the environment, while the "Image Extraction" images illustrate the results of point and line extraction. The color of the trajectory lines represents the absolute position error.

B. REJECTION STRATEGY AND WEIGHT OPTIMIZATION

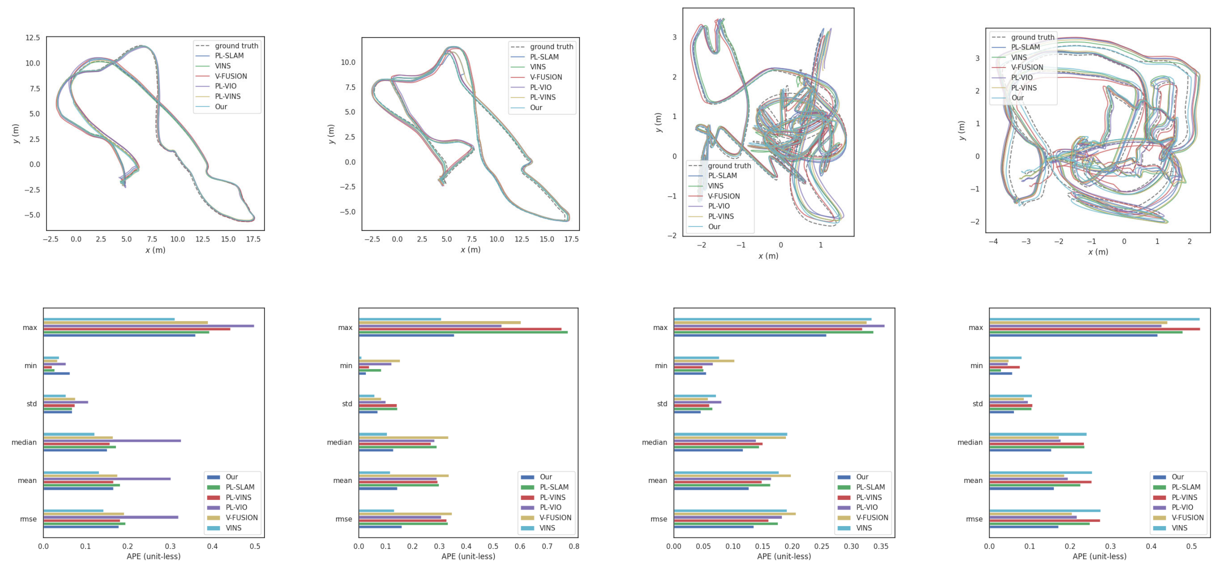
After optimizing the line length rejection strategy, we proceeded with dynamic weight optimization. To evaluate the localization accuracy on the EuRoC dataset, we conducted experimental comparisons with VINS, VINS-Fusion, PL-SLAM, PL-VIO, and PL-VINS, using Absolute Trajectory Error (ATE) as the metric.

Table 2 shows the ATE results, revealing that the VINS series, which employs Shi-Tomasi corner detection, offers better robustness in dark environments with bright points, such as those in the MH dataset. In these scenarios, point feature extraction and tracking outperform line feature methods, which is why VINS outperforms PL-SLAM, PL-VIO, and PL-VINS. Our approach, which uses point-line weighted optimization to minimize the influence of lines, achieves better accuracy than other point-line methods but still falls short of VINS in these specific environments.

Conversely, in brighter and more structured environments, such as those in the V1 and V2 datasets, our method excels. Line feature extraction and tracking are more reliable under these conditions, leading to superior performance. However, the MH sequences highlight a limitation of our method: the difficulty in accurately tracking line features in low-light conditions. This limitation underscores the need for further refinement in line feature extraction techniques to improve robustness across diverse lighting scenarios.

TABLE 2. Comparison of Line Length Filtering and Dynamic Weight Adjustment Optimization Algorithms with Other Open-Source Algorithms. (Bold Indicates Best ATE Performance).

Sequence	VINS	V-FUSION	PL-SLAM	PL-VIO	PL-VINS	Our
MH-01	**0.086**	0.191	0.230	0.192	0.198	0.137
MH-02	0.093	0.099	0.184	0.149	0.094	**0.090**
MH-03	**0.078**	0.117	0.234	0.265	0.116	0.110
MH-04	**0.141**	0.190	0.193	0.319	0.181	0.178
MH-05	**0.131**	0.345	0.331	0.306	0.325	0.160
V1-01	0.153	0.159	0.163	0.166	0.154	**0.152**
V1-02	0.098	0.333	0.108	0.083	0.082	**0.081**
V1-03	0.191	0.206	0.175	0.183	0.160	**0.135**
V2-01	**0.074**	0.137	0.100	0.112	0.087	0.081
V2-02	0.134	-	0.131	0.156	0.110	**0.105**
V2-03	0.327	0.204	0.248	0.216	0.274	**0.171**
Mean	0.132	0.198	0.190	0.195	0.162	**0.127**



(A) MH-04 (B) MH-05 (C) V1-03 (D) V2-03

FIGURE 6. PL-SLAM, PL-VIO, PL-VINS and our method are compared, and A-D are the MH-04, MH-05, V1-03 and V2-03 difficult sequence set trajectory maps.

Figure 5 from the MH-05 sequence illustrates this issue. In low-light environments, point extraction is concentrated near bright spots, making tracking easier, while line extraction faces significant challenges. To address this, our method adjusts the weight of lines and points dynamically, which is crucial for maintaining SLAM accuracy in environments where line features are less stable. On average, our method improves PL-VINS accuracy by 23.06% and outperforms other algorithms, especially in complex environments.

After analyzing the ATE results, we further compared the algorithms that incorporate line features by testing their runtime, as detailed in Table 3. In PL-SLAM, ORB-based corner point extraction and LSD-based line feature extraction consume significant time. While PL-VIO and PL-VINS use

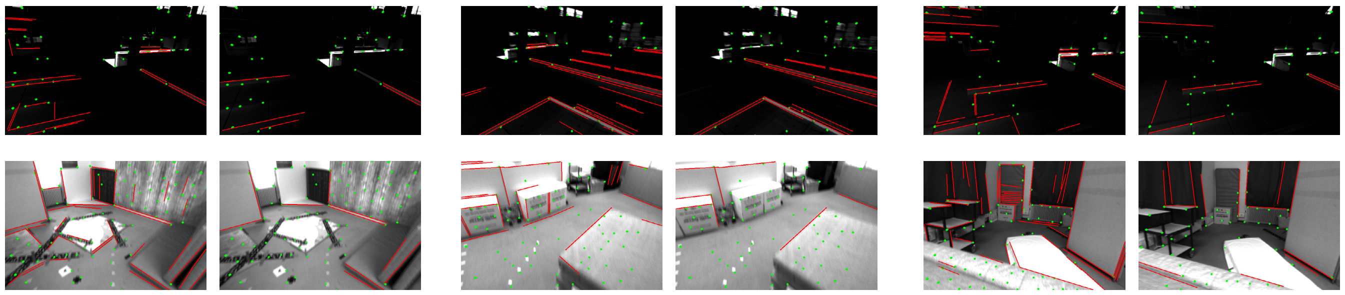


FIGURE 7. **Comparison of the algorithms in low-light and bright environments in the Euroc dataset: the PL-VINS algorithm on the left, the optimized algorithm on the right, line features in red and point features in green.**

Shi-Tomasi for point extraction, the original LSD used in PL-VIO remains time-consuming. PL-VINS improves line extraction efficiency by adjusting LSD's parameters, and our method further reduces runtime by decreasing the number of lines, speeding up line-matching.

TABLE 3. **Comparison of Computational Overhead Between Our Algorithm and Current Open-Source Feature Line Algorithms.**

Module	PL-SLAM	PL-VIO	PL-VINS	Our
Point Extraction Tracking(ms)	22.24	13.32	13.36	13.36
Line Extraction Tracking(ms)	80.24	87.78	32.23	30.14

Figure 6 demonstrates the visual comparison of 3D motion trajectories, while Figure 7 compares the line feature extraction results of our method with PL-VINS. Our method is able to extract high-quality line segments more stably in complex environments, reduce the tracking of drifting line segments, and guarantee the stability of dominant line segments through a dynamic line segment length rejection strategy. By dynamically evaluating the environmental conditions based on point and line data, our method adjusts the loss function in point and line optimization, which makes the SLAM system more adaptive and improves the overall performance.

VI. CONCLUSION

Our experiments demonstrate the successful performance of our method in various scenarios and produce satisfactory results. By optimizing the line length elimination strategy, the system can accurately extract a small number of high-quality, advantageous lines in complex environments. In addition, dynamic weighted optimization enables the system to adapt to different environmental conditions and set different optimization weights, thereby improving the accuracy and robustness of the SLAM system.

However, it is worth noting that line features are currently underutilized. For example, they are not fully exploited in loop closure detection, and there are challenges with line matching accuracy. Additionally, in partially bright and dark environments, the accuracy of lines is not as reliable as corner points, resulting in limited utilization of line features.

For future work, in addition to the aforementioned issues, we believe that single-sensor systems have fatal limitations. For instance, vision-based SLAM systems cannot operate in completely dark environments, while laser-based SLAM systems cannot capture rich environmental textures. Therefore, we plan to address scale, texture, and lighting issues in SLAM by utilizing multi-sensor fusion methods.

REFERENCES

[1] Y. Jia, X. Yan, and Y. Xu, "A survey of simultaneous localization and mapping for robot," in *Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Dec. 2019, pp. 857–861.

[2] A. R. Sahili, S. Hassan, S. M. Sakhrieh, J. Mounsef, N. Maalouf, B. Arain, and T. Taha, "A survey of visual SLAM methods," *IEEE Access*, vol. 11, pp. 139643–139677, 2023.

[3] S. Zhang, L. Zheng, and W. Tao, "Survey and evaluation of RGB-D SLAM," *IEEE Access*, vol. 9, pp. 21367–21387, 2021.

[4] K. Di, W. Wan, H. Zhao, Z. Liu, R. Wang, and F. Zhang, "Progress and applications of visual SLAM," *J. Geodesy Geoinf. Sci.*, vol. 2, p. 38, Jun. 2019.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[6] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[7] X. Ma and S. Ning, "Real-time visual-inertial SLAM with point-line feature using improved EDLines algorithm," in *Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Jun. 2020, pp. 1323–1327.

[8] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3923–3931.

[9] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.

[10] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1363–1370, Aug. 2020.

[11] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[12] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6565–6574.

[13] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-SLAM: Low-drift monocular SLAM in indoor environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6583–6590, Oct. 2020.

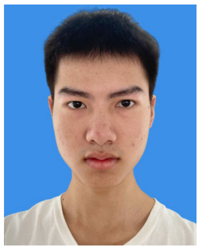
[14] Y. Zhang, M. Hsiao, Y. Zhao, J. Dong, and J. J. Engel, "Distributed client-server optimization for SLAM with limited on-device resources," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5336–5342.

[15] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.
[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[17] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, Aug. 1988, p. 5244.
[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[20] X. Xue and X. Lv, "Stereo visual inertial SLAM algorithm fusing point and line features," in *Proc. China Autom. Congr. (CAC)*, Nov. 2022, pp. 1687–1692.
[21] Z. Teng, B. Han, J. Cao, Q. Hao, X. Tang, and Z. Li, "PLI-SLAM: A tightly-coupled stereo visual-inertial SLAM system with point and line features," *Remote Sens.*, vol. 15, no. 19, p. 4678, Sep. 2023.
[22] H. Yang, J. Yuan, Y. Gao, X. Sun, and X. Zhang, "UPLP-SLAM: Unified point-line-plane feature fusion for RGB-D visual SLAM," *Inf. Fusion*, vol. 96, pp. 51–65, Aug. 2023.
[23] Q. Peng, Z. Xiang, Y. Fan, T. Zhao, and X. Zhao, "RWT-SLAM: Robust visual SLAM for highly weak-textured environments," 2022, *arXiv:2207.03539*.
[24] J. He, M. Li, Y. Wang, and H. Wang, "PLE-SLAM: A visual-inertial SLAM based on point-line features and efficient IMU initialization," 2024, *arXiv:2401.01081*.
[25] F. Zhou, L. Zhang, C. Deng, and X. Fan, "Improved point-line feature based visual SLAM method for complex environments," *Sensors*, vol. 21, no. 13, p. 4604, Jul. 2021.
[26] J. Lee and S.-Y. Park, "PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7033–7040, Oct. 2021.
[27] J. Yan, Y. Zheng, J. Yang, L. Mihaylova, W. Yuan, and F. Gu, "PLPF-VSLAM: An indoor visual SLAM with adaptive fusion of point-line-plane features," *J. Field Robot.*, vol. 41, no. 1, pp. 50–67, Jan. 2024.
[28] L. Yu, E. Yang, and B. Yang, "AFE-ORB-SLAM: Robust monocular VSLAM based on adaptive FAST threshold and image enhancement for complex lighting environments," *J. Intell. Robotic Syst.*, vol. 105, no. 2, p. 26, Jun. 2022.
[29] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, Apr. 2018.

YIBO CAO was born in Hengyang, Hunan, China, in 1971. He received the B.S. and M.S. degrees from the Department of Electrical and Mechanical Engineering, Xian University of Electronic Science and Technology, in 1994 and 1997, respectively, the Ph.D. degree from the School of Mechanical Engineering, South China University of Technology, Guangzhou, China, in 2007, and the Ph.D. degree from the Department of Precision Instruments, Tsinghua University, Beijing, China, in 2009.

He joined the Software College, South China Normal University, in 2009. He is currently an Associate Professor. His research interests include AI artificial intelligence technology research, optical electromechanical integration technology, the Internet of Things technology, pattern recognition, and simultaneous localization and mapping (SLAM).

ZEHAO LUO was born in Puning, Guangdong. China, in 2000. He is currently pursuing the degree with the School of Software, South China Normal University. His current research interests include computer vision, vision simultaneous localization, and mapping and indoor positioning.



ZHENYU DENG was born in Shaoyang, Hunan, China, in 2001. He is currently pursuing the degree with the School of Software, South China Normal University. His current research interests include computer vision, vision simultaneous localization, path tracking, mobile robots, and autonomous robot navigation.

• • •