IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 May 2024, accepted 3 June 2024, date of publication 5 June 2024, date of current version 26 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3409946

RESEARCH ARTICLE

Structured Line Feature and Merge Strategy Localization Algorithm Based on Constraints

XIGUANG ZHANG AND RENHAN WU
Zhongyuan-Petersburg Aviation College, Zhongyuan University of Technology, Zhengzhou 450007, China
Corresponding author: Renhan Wu (renhanwu1999@163.com)

This work was supported by the Key Research and Development Projects in Henan Province under Grant 231111221600.

ABSTRACT In scenes characterized by weak textures and high motion speeds, traditional visual inertial odometer (VIO) systems face challenges, including reduced accuracy and inadequate real-time performance. These deficiencies may cause the robot to lose some frames in actual operation and the actual error is too large. To address these degradation issues, we propose a VIO system based on point and line fusion features. The system incorporates novel methods for extracting both point and line features. Firstly, subpixel corner extraction is employed to enhance the accuracy of the point feature extraction algorithm. Secondly, for the line feature extraction algorithm, we utilize the FLD line feature extraction method which significantly improves its speed in most environments and enhances its real-time performance. Additionally, to ensure accurate and stable tracking of line features, we introduce a novel idea-combining approach after optimizing these features that reduces pose estimation errors and enhances overall algorithm precision. In the experiment, we found that in the process of online feature extraction, the same line was repeatedly detected in the previous frame and the next frame, and the detected line would become a new line, which would increase the pose estimation error. Therefore, we proposed an optimization method to delete redundant lines for the triangulated line features. Experimental results demonstrate that our proposed method outperforms PL-VINS in terms of both accuracy and speed of line feature extraction on commonly used EuRoC datasets.

INDEX TERMS PL-VINS, subpixel corner points, line feature extraction, redundant line removal.

I. INTRODUCTION

With the rapid development of unmanned vehicles and aerial vehicles, mobile robot technology continues to advance, and localization and mapping (SLAM) algorithm is also constantly studied. In the past few years, many monocular VIO algorithms for positioning and mapping through feature point extraction have been proposed, such as [1], [2], [3], [4], [5], [6], [7], and [8]. The SLAM algorithm that uses the point feature extraction algorithm alone for feature extraction has a good effect in common scenes, but in the environment with weak texture, fast motion speed and not obvious feature information, the VIO algorithm based on point feature extraction has some problems such as accuracy decline. Compared with point features, line features contain more information and provide additional constraints. Recently, the use of line features for the fusion of point-based VIO algorithms has received increasing attention, such as [9], [10], [11], and [12]. Pumarola et al. [13] proposed point-and-line real-time monocular visual SLAM(PL-SLAM) in order to solve the problem of precision degradation. The combination of camera sensor and PL-SLAM algorithm has some limitations

In practical application. The addition of inertial measurement unit (IMU) can solve the pose estimation problem well, overcome the shortcomings of pure visual SLAM and have great advantages in weight and composition. In the visual inertia odometer (VIO), the VINS-Mono algorithm performs better in real scenes [1], [14], [15], [16]. Although VIO algorithm based on point and line feature fusion is better than VIO algorithm based on point feature in challenging

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

environment, it still has some problems such as insufficient accuracy and real-time performance.

For PL-VIO algorithm, the accuracy and real-time performance of feature extraction and tracking algorithm is very important. Fu et al. [17] proposed real-time monocular Visual Inertial SLAM (PL-VINS) with point and line features. Although PL-VINS is one of the best performing visual inertial SLAM algorithms for point and line features, it still has some shortcomings in the environment of weak texture or fast motion speed.

For PL-VINS algorithm, the accuracy of point feature extraction algorithm and the speed of line feature extraction algorithm are not stable, especially in the scene with weak texture and fast motion speed. First of all, the point feature extraction algorithm has a large feature extraction error, resulting in the accuracy of PL-VINS algorithm is greatly reduced. To solve this problem, we propose to use subpixel corner extraction algorithm to replace the Harris and Stephens corner extraction algorithm in the original algorithm [18]. Using subpixel corner extraction algorithm to optimize point feature extraction can improve the accuracy of PL-VINS. Secondly, to improve the speed of line feature extraction, we propose to use the FLD line feature extraction algorithm [19] to replace the LSD line feature extraction algorithm [20] in the original algorithm. Using FLD line feature extraction algorithm to optimize the original algorithm, speeding up line feature extraction speed can improve the real-time performance of PL-VINS algorithm.

On the other hand, experiments show that in the process of line feature extraction by PL-VINS, the same line is repeatedly detected in the previous frame and the next frame, and the line repeatedly detected will become a new line, which will increase the pose estimation error when the algorithm performs line feature matching. The accuracy of PL-VINS algorithm is greatly reduced. To solve this problem, we propose a new matching method for line feature tracking. For the repeated lines in extraction, we will adopt the method of optimizing the line features first and merging the repeated lines to reduce the pose estimation error and improve the accuracy of PL-VINS algorithm.

Finally, by summarizing the above problems and solutions, the optimization strategy of PL-VINS algorithm is proposed. Its characteristics include:

1. Subpixel corner extraction algorithm is used to replace the corner extraction algorithm to iterate and improve the accuracy of the initial value, and subpixel image marginalization constraints are applied to the detection results. Firstly, a function is used to obtain the number of corner points that need to be optimized and the initialization result. Then, the algorithm of detecting subpixel corner points is used to iterate and improve the accuracy of the initial value, and image marginalization is constrained according to the detection result.

2. The LSD line feature extraction used in the original algorithm is replaced by FLD line feature extraction, which improves the speed of line feature extraction.

3. Delete redundant lines that appear in the process of line feature extraction to reduce the pose estimation error. In the process of online feature extraction, the same line is repeatedly detected in the previous frame and the next frame, and the detected line will become a new line, which will increase the pose estimation error. Our optimization method is to delete redundant lines for the triangulated line features. At the beginning, we considered two schemes to delete the redundant lines. The first one was to merge the redundant lines when the FLD line feature algorithm extracted the line features. This scheme would terminate the program directly, and it was inferred that the tracking was not satisfied due to insufficient line features at the beginning of the program. Therefore, we adopt the second scheme to triangulate the line features and then merge them, which can effectively reduce the pose estimation error. For some data sets, there are fewer initial line features, so parameter adjustment is carried out, and the adjusted parameter is the elimination of points outside the line. The elimination distance of points outside the line in the original method is 10, but it is modified to 20, considering that more line features are needed for merging. The main solution is that for data sets with fewer line features, such as easy class, the effect may be poor due to the small number of line features extracted. In the experiment, it is found that for different data sets, the corresponding parameters have different effects, and more lines should be used to merge data sets with fewer line features.

In the line feature merging idea, the line feature is first structured, that is, transferred to the Manhattan coordinate system, the three-dimensional position of the line feature is judged, and then the line feature is traversed. There are three conditions for the merging: 2d plane, three-dimensional distance judgment and the id of the line feature. First of all, the distance between two lines in the three-dimensional Manhattan coordinate system can be parameterized, which is set as 0.1 in this paper. Then the observation line id is detected, only the original id may be a duplicate line of the new id; finally, the distance of 2d plane lines is judged. After the detection is complete, the merged features are deleted to complete the deletion of redundant lines.

To sum up, for the first point, Harris corner extraction algorithm extracts corner points at the pixel level. Subpixel corner extraction is adopted to improve the accuracy of corner extraction and achieve better performance in the algorithm. For the second point, the FLD algorithm is mainly based on the gradient information of the image to quickly detect the line. It uses the edge information in the image to identify possible lines by calculating the gradient amplitude and direction. In contrast, the LSD algorithm requires a more complex computation process. It not only performs gradient calculations on images, but also performs operations such as segmented search and parameter space clustering, which are relatively more time-consuming processes. In the process of the third point line feature extraction, the same line is repeatedly detected in the previous frame and the next frame, and the detected line will become a new line, which will

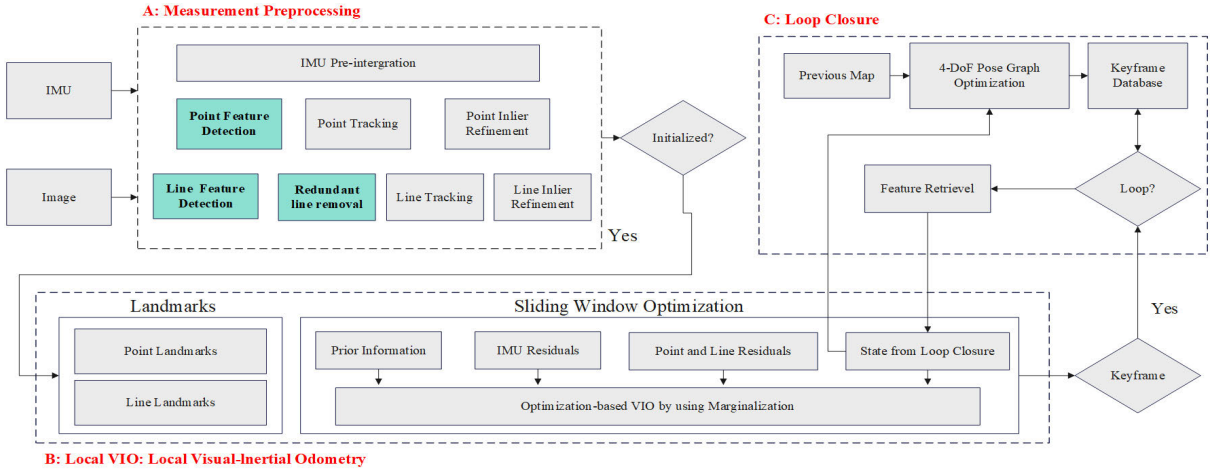


FIGURE 1. A block diagram illustrates the pipeline of the proposed VIO system. The blue dotted box and blue font are the improvements of PL-VINS algorithm.

increase the pose estimation error. Our optimization method is to delete redundant lines for the triangulated line features, so as to improve the accuracy. In addition, parameters need to be adjusted for some data sets to meet the number of line merges. For data sets with fewer line features, such as the easy class, the effect may be poor due to the small number of lines. Adjust parameters as needed.

The second part of this paper introduces the related work, and proposes the algorithm optimization for PL-VINS and the framework structure will be described in the third section. The fourth part is the comparison test and results between the original algorithm and the optimized algorithm. The fifth part is discussion and analysis. The sixth part is the conclusion.

II. RELATED WORK

Qin et al. [1] proposed to use a tightly-coupled, nonlinear optimization based method to obtain high precision visual inertia odometer called VINS-Mono, which is a robust and universal monocular visual inertia state estimator. In 2015, Mur-Artal et al. [21] proposed a system that is robust to severe motion clutter and includes fully automatic initialization, which is called ORB-SLAM. In 2017, Albert Pumarola et al. pointed out that only relying on point features in weak texture scenes is very deadly and proposed a visual SLAM with point and line features (PL-SLAM) based on ORB-SLAM, which has greatly improved accuracy and efficiency compared with ORB-SLAM. Later, Fu et al. [17] proposed a visual inertial SLAM based on the VINS-Mono algorithm using point and line features, it is proposed that LSD algorithm is designed for scene shape representation rather than attitude estimation, which is the bottleneck of real-time performance due to its high computational cost and proposed a length suppression strategy to improve the LSD algorithm to solve the problem of large computation. Gomez-Ojeda et al. [22] proposed PL-SLAM, a stereo vision SLAM system that combines dots and line segments to work robustly in a wider range of scenes, especially in scenes where mid-point features in images are scarce or unevenly distributed. Zheng et al. [9] pointed out that line features help improve the robustness of the system in challenging scenarios when point features cannot be reliably detected or tracked, such as low-texture environments or lighting changes. Therefore, a three-dimensional VIO system based on tightly coupled filtration using points and lines (Trifo-VIO) is proposed. Zou et al. [23] proposed a method (StructVIO) for applying structural regularities in artificial environments, d and described the regularities by using the Atlanta world model instead of the Manhattan world hypothesis. In 2018, He et al. [10] in order to solve the problem of estimating camera trajectories and constructing structural three-dimensional (3D) maps based on inertial measurement and visual observation, proposed a monocular visual inertial odometer (PL-VIO) with tightly coupled point and line features. Line features provide more structural information, making the PL-VIO method superior to the VIO algorithm with only a few features. In 2020, Wen et al. [12] in order to solve the problem of traditional point feature-based visual SLAM, it is difficult to find reliable point features to estimate the camera attitude in a weak texture long corridor environment. Visual SLAM alone can easily lose point features in the case of weak textures or fast motion, causing the system to crash. Monocular VIO has unobservable IMU scales at uniform motion, proposed an optimized point and line feature fusion stereo vision inertial odometer (PLS-VIO), which significantly improved its accuracy in environments such as weak textures. Lee and Park et al. [24] proposed a real-time monomial vision inertial synchronous positioning and mapping (PLF-VINS) with point-line fusion and parallel line fusion. The real-time positioning accuracy was improved when the two proposed residual errors were combined with sliding window optimization. Zhu et al. [25] pointed out that it is a great challenge for autonomous robots to efficiently estimate states and generate high-precision 3D maps in low-texture indoor scenes and proposed a visual inertial synchronous localization and mapping (SLAM) system

using point features, line features and depth information provided by RGBD cameras. The main advantage is that it can improve the accuracy of RGBD camera state estimation and intensive 3D mapping. Kuang et al. [26] in order to solve the problem of inaccurate positioning and frequent tracking loss of mobile robots in challenging scenes, the capability of point-based visual synchronous positioning and mapping (vSLAM) is beyond, proposed a real-time and robust point-line based monocular visual-inertial SLAM (VINS) system for smart city mobile robots oriented to 6G. The method could enable mobile robots to accurately position themselves in smart cities with complex environments. Luo et al. [27] pointed out the problems of image enhancement oversaturation and unreasonable weights in back-end optimization in VIO methods, and proposed that monocular VIO can improve the positioning accuracy and robustness of UAV navigation system through point and line fusion and back-end adaptive optimization. And compared with the pl-vins algorithm, the accuracy of the proposed method on the public EuRoc dataset is improved by 32.3% on average. This method is applied to UAV navigation in GNSS interference environment. Yang et al. [28] pointed out that when the texture information in the scene is missing or the image is blurred due to the fast movement of the camera, the number of point features is often small, thus affecting the accuracy of pose estimation. A visual inertial state estimator system (PLS-VINS) based on point-line features and structural constraints is proposed, which combines points and line segments to enhance the performance of feature extraction in a wider range of scenes and optimize the system state by jointly minimizing the pre-integral constraints of the inertial measurement unit (IMU). Zhao et al. [29] pointed out that in an indoor environment with low texture, the point feature-based visual SLAM system has poor robustness and low trajectory accuracy. Therefore, a visual inertial SLAM algorithm based on point-and-line feature fusion is proposed. This method can be used to improve the accuracy of indoor scenes.

III. METHOD

In this paper, a real-time optimized monocular visual Inertial SLAM method (PL-VINS) for optimizing point and line features is proposed, which improves the point features, line features and pose estimation errors respectively. The algorithms based on point feature include Harris corner extraction algorithm, Shi-Tomasi corner extraction algorithm and subpixel corner extraction algorithm. The algorithms based on line features introduced include FLD line feature extraction algorithm and LSD line feature extraction algorithm, as shown in Figure 1. Specifically, we first use subpixel corner extraction algorithm instead of corner extraction algorithm, then use FLD line feature extraction algorithm instead of LSD line feature extraction algorithm, and finally delete the repeated detection lines in the process of online feature extraction. From these three aspects to improve. As shown in Fig1.

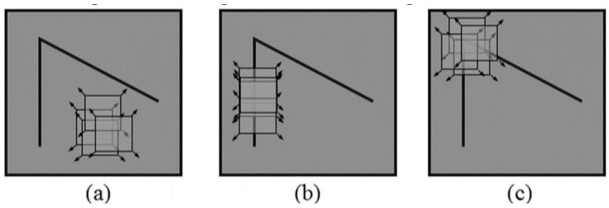


FIGURE 2. Harris corner point sliding window.

A. SUBPIXEL CORNER AND HARRIS CORNER

The image of the feature point detection is an important step, where the corner is the most common type of point features, Harris corner detection, extraction is pixel level. In the actual scene of the corner position is continuous, and the camera pixel is discrete, the two will have deviation, and compared with the conventional corner detection, subpixel corner detection with real numbers instead of integers to represent the corner coordinates, so the use of sub-pixel corner extraction will be more accurate, the matching results are also significant. Harris corner is a point feature extraction algorithm based on gray image; algorithm principle for the image of the corner point near the pixel points in the gradient direction or gradient amplitude are greater changes, edge points have a large horizontal or vertical gradient, while other points have smaller horizontal and vertical gradient. Therefore, only need to calculate the pixel gradient, can be determined according to the constraints of the corner. The basic idea of corner detection is to use a fixed window on the image along all directions of sliding, compare the sliding window before and after the gray point of the changes, if there is a large gray change in any direction of the sliding window, it is believed that there is a corner in the window; if no change in any direction, it is a uniform area; if the gray only changes in one direction, it may be the edge of the image. As shown in Fig 2.

Harris corner detection method is to set a window near the corner and observe the change of gray value in a certain direction by slowly moving the window. Assuming the window displacement (u, v), the covariance is used to represent the change in intensity of the gray value:

$$R = \sum (I(x + u, y + v) - I(x, y))^2 \tag{1}$$

The steps of Harris corner detection principle are as follows: First measure the horizontal direction gray value change in the window, and then measure the vertical direction gray value change. If the gray value of the horizontal and vertical directions changes greatly, it is a corner point.

The above process can be verified by expanding Taylor's formula:

$$\begin{aligned} R &\approx \sum \left((I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v - I(x, y) \right)^2 \\ &= \sum \left(\left(\frac{\partial I}{\partial x}u \right)^2 + \frac{\partial I}{\partial y}v^2 + 2\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}uv \right) \end{aligned} \tag{2}$$

The matrix form is:

$$R \approx [u, v] \begin{bmatrix} \sum \left(\frac{\delta I}{\delta x}\right)^2 & \sum \left(\frac{\delta I}{\delta x} \frac{\delta I}{\delta y}\right) \\ \sum \left(\frac{\delta I}{\delta x} - \frac{\delta I}{\delta y}\right) & \sum \left(\frac{\delta I}{\delta y}\right)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3)$$

By calculating the eigenvalues of the verifiable matrix:

$$Dst(x, y) = Det\left(C^{(x,y)}\right) - k \cdot \left(trC^{(x,y)}\right)^2 \quad (4)$$

where $Det\left(C^{(x,y)}\right)$ is the determinant of the matrix, $k \cdot \left(trC^{(x,y)}\right)^2$ is the direct trace of the matrix, and k is an empirical coefficient, usually in the range of 0.04-0.06, that exists to regulate the shape of the function.

In summary, Harris corner extraction algorithm extracts corner points at the pixel level. Subpixel corner extraction is adopted to improve the accuracy of corner extraction and achieve better performance in the algorithm. The main implementation methods are as follows: Firstly, the function is used to obtain the number of corners to be optimized and the initialization result; then, the subpixel corner algorithm is used to iterate and improve the accuracy of the initial value, and the boundary detection problem of the subpixel corner edge of the image is carried out according to the detection result.

B. LSD LINE DETECTION ALGORITHM AND FLD LINE DETECTION ALGORITHM

In this paper [20], in order to solve the problem that the LSD straight-line detection algorithm has a large amount of computation and is not real-time, a length suppression strategy is proposed to improve the LSD algorithm. LSD algorithm obtains the pixel set of the line through the local analysis of the image, and then verifies the solution through the hypothesis parameters, combines the pixel set and the error control set, and then adaptively controls the number of false detections. Under normal circumstances, the most basic idea of detecting the straight line in the image is to detect the pixel set with large gradient changes in the image, and LSD algorithm is to use the gradient information and the row line to detect the straight line. Gaussian blur is used first and then downsampling, in order to reduce the sawtooth effect. The gradient is calculated and the gradient size is defined as:

$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \quad (5)$$

The horizontal Angle is calculated as:

$$ang(x, y) = \tan^{-1}\left(\frac{g_x(x, y)}{-g_y(x, y)}\right) \quad (6)$$

Construct the rectangle and determine the center point of the rectangle:

$$c_x = \frac{\sum_{j \in Region} G(j) \cdot x(j)}{\sum_{j \in Region} G(j)}$$
$$c_y = \frac{\sum_{j \in Region} G(j) \cdot y(j)}{\sum_{j \in Region} G(j)} \quad (7)$$

where $G(j)$ is the gradient value of pixel j, and j is every pixel in the region.

$$M = \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix} \quad (8)$$

The Angle of the rectangle is set to be the Angle of the eigenvector, which is related to the minimum eigenvalue of the M matrix.

$$m^{xx} = \frac{\sum_{j \in Region} G(j) \cdot (x(j) - c_x)^2}{\sum_{j \in Region} G(j)}$$
$$m^{yy} = \frac{\sum_{j \in Region} G(j) \cdot (y(j) - c_y)^2}{\sum_{j \in Region} G(j)}$$
$$m^{xy} = \frac{\sum_{j \in Region} G(j) \cdot (x(j) - c_x)(y(j) - c_y)}{\sum_{j \in Region} G(j)} \quad (9)$$

FLD line detection algorithm is cited in this paper [19], and attempts to use straight line features to replace the original SURF point features for building identification. It is concluded that compared with point features, line features are easier to find and more robust, and line features are basically not affected by illumination, occlusion and Angle change.

The performance of the FLD line detection algorithm is similar to that of LSD, and the experimental results show that the speed of FLD line feature extraction can be significantly improved in most environments. Therefore, the FLD line detection algorithm is adopted to replace the original LSD line detection algorithm in PL-VINS to improve the real-time performance of the PL-VINS algorithm.

C. REDUNDANT LINES ARE DELETED

In the process of online feature extraction, the same line is repeatedly detected in the previous frame and the next frame, and the detected line will become a new line, which will increase the pose estimation error. Our optimization method is to delete redundant lines for the triangulated line features. At the beginning, we considered two schemes to delete the redundant lines. The first one was to merge the redundant lines when the FLD line feature algorithm extracted the line features. This scheme would terminate the program directly, and it was inferred that the tracking was not satisfied due to insufficient line features at the beginning of the program. Therefore, we adopt the second scheme to triangulate the line features and then merge them, which can effectively reduce the pose estimation error. For some data sets, there are fewer initial line features, so parameter adjustment is carried out, and the adjusted parameter is the elimination of points outside the line. The elimination distance of points outside the line in the original method is 10, but it is modified to 20, considering that more line features are needed for merging. The main solution is that for data sets with fewer line features, such as easy class, the effect may be poor due to the small number of line features extracted. In the experiment, it is found that for different data sets, the corresponding parameters have different effects, and more lines should be used to merge data sets with fewer line features.

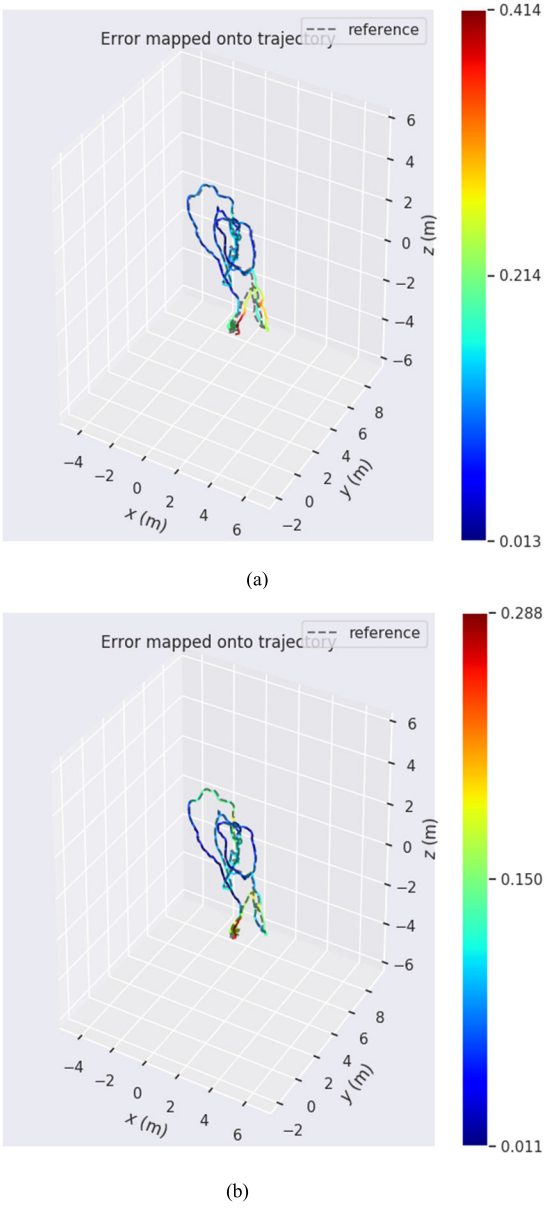


FIGURE 3. (a) shows the trajectory error of the original PL-VINS algorithm on the MH_01_easy dataset. (b) The trajectory error of the optimized PL-VINS on the data set (Scale unit is m).

In the line feature merging idea, the line feature is first structured, that is, transferred to the Manhattan coordinate system, the three-dimensional position of the line feature is judged, and then the line feature is traversed. There are three conditions for the merging: 2d plane, three-dimensional distance judgment and the id of the line feature. First of all, the distance between two lines in the three-dimensional Manhattan coordinate system can be parameterized, which is set as 0.1 in this paper. Then the observation line id is detected, only the original id may be a duplicate line of the new id; finally, the distance of 2d plane lines is judged. After the detection is complete, the merged features are deleted to complete the deletion of redundant lines.

TABLE 1. The absolute pose error (APE) of the original algorithm and the improved algorithm is compared (Unit: meter).

	Original	Improved
MH_01_easy	0.18	0.14
MH_02_easy	0.18	0.13
MH_03_medium	0.27	0.22
MH_04_difficult	0.36	0.30
MH_05_difficult	0.38	0.31
V1_01_easy	0.14	0.12
V1_02_medium	0.31	0.30
V1_03_difficult	0.31	0.25
V2_01_easy	0.12	0.12
V2_02_medium	0.25	0.20
V2_03_difficult	0.28	0.24

IV. EXPERIMENT

In this paper, the PL-VINS algorithm is compared with the improved PL-VINS algorithm in the common data set EuRoC. By printing line features feature extraction speed, and using the EVO trajectory measurement tool, it is more intuitive to see where the algorithm has improved. EVO is a trajectory evaluation tool for visual odometer and SLAM problems. The core function is the ability to plot the camera's trajectory or evaluate the error of the estimated trajectory from the true value. The absolute pose error (APE) is commonly used as the absolute trajectory error, which compares the estimated trajectory to the reference trajectory and calculates statistics for the entire trajectory, applicable to testing the global consistency of the trajectory. Relative attitude error (RPE) is not a comparison of absolute attitude, but a comparison of motion (attitude increment). Relative pose error can give local accuracy. Relative pose error (RPE) is divided into translation error and rotation error.

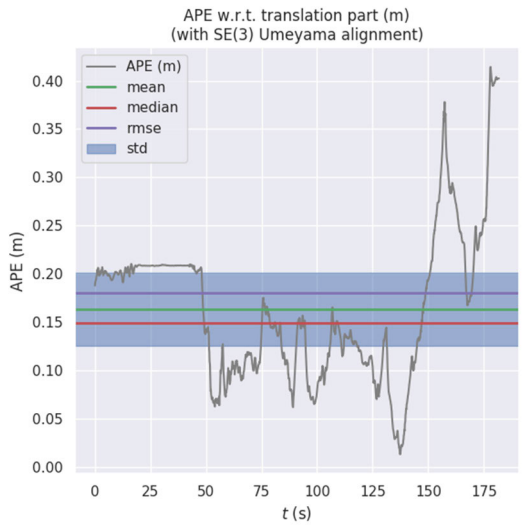
Take MH_01_easy in the EuRoC data set as an example to compare the trace error (Fig3.):

A. PRECISION COMPARISON OF ALGORITHMS

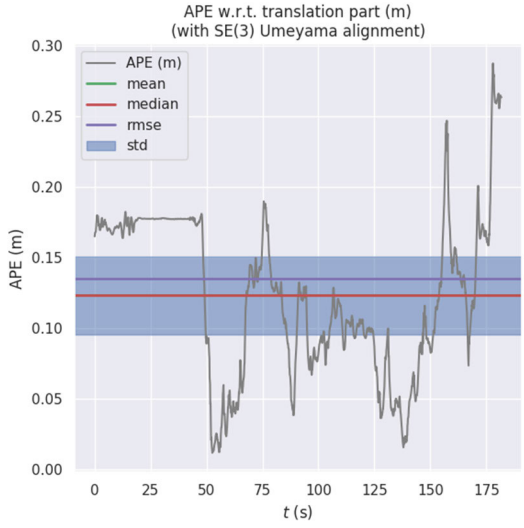
In the case of poor robustness and accuracy, subpixel corner algorithm and redundant line deletion can effectively solve the problem in the environment with weak texture and few key points. Therefore, subpixel corner algorithm is used to replace the Harris feature extraction algorithm, and redundant lines are deleted to improve the accuracy of the algorithm. The original PL-VINS algorithm and the optimized algorithm were verified on EuRoC data set, and the accuracy of the algorithm was compared. As shown in Table 1.

The original algorithm and the optimized algorithm are used to compare RMSE, translation error and trajectory on the EuRoC data set MH_01_easy. As shown in Fig 4, 5 and 6.

The original algorithm and the optimized algorithm are used to compare RMSE, translation error and trajectory on the EuRoC data set MH_03_medium. As shown in Fig7, 8 and 9.



(a)



(b)

FIGURE 4. (a) is the absolute attitude error (APE) of the original algorithm, and (b) is the absolute attitude error (APE) of the improved algorithm. The black line is the absolute pose error (APE), the blue line is the root mean square error (RMSE), the red line is the median error (median), the green line is the mean error (mean), and the blue area is the standard deviation (std).

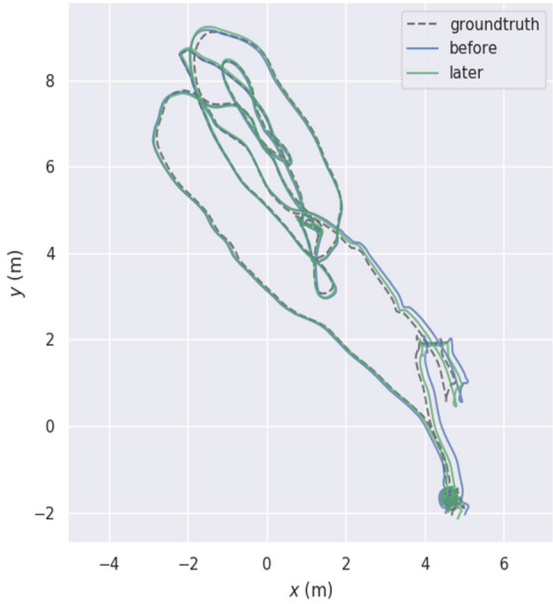


FIGURE 5. Compare the trajectories between the original algorithm and the optimized algorithm. The black line is the true trajectory, the blue line is the trajectory of the original algorithm, and the green line is the trajectory of the optimized algorithm.

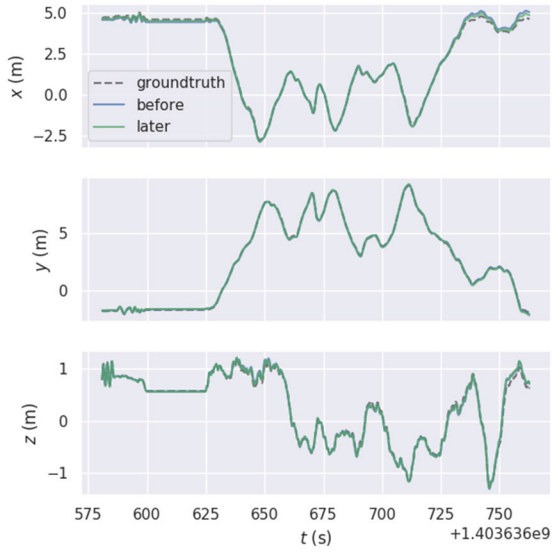


FIGURE 6. To the original algorithm compared with translation error of the optimized algorithm.

In the case of poor robustness and accuracy, subpixel corner algorithm and redundant line deletion can effectively solve the problem in the environment of weak texture and few key points. Therefore, subpixel corner algorithm is used instead of Harris feature extraction algorithm, and redundant lines are deleted to improve the accuracy of the algorithm. The PLS-VINS algorithm and the optimization algorithm are verified on the EuRoC data set, and the accuracy of the algorithm is compared. As shown in Table 2.

B. COMPARISON OF LINE FEATURE EXTRACTION SPEED

In terms of feature extraction speed, this paper uses FLD line feature extraction algorithm instead of LSD line feature extraction algorithm to improve the feature extraction speed and have better real-time performance in the actual scene. The original PL-VINS algorithm and the optimized algorithm were verified on the EuRoC data set, and the line feature extraction time of the original algorithm and the improved algorithm was printed out to reflect the speed of the comparison line feature extraction. As shown in Table 3.

The FLD algorithm is mainly based on the gradient information of the image to quickly detect the line. It uses the edge information in the image to identify possible lines by calculating the gradient amplitude and direction. In contrast, the LSD algorithm requires a more complex computation

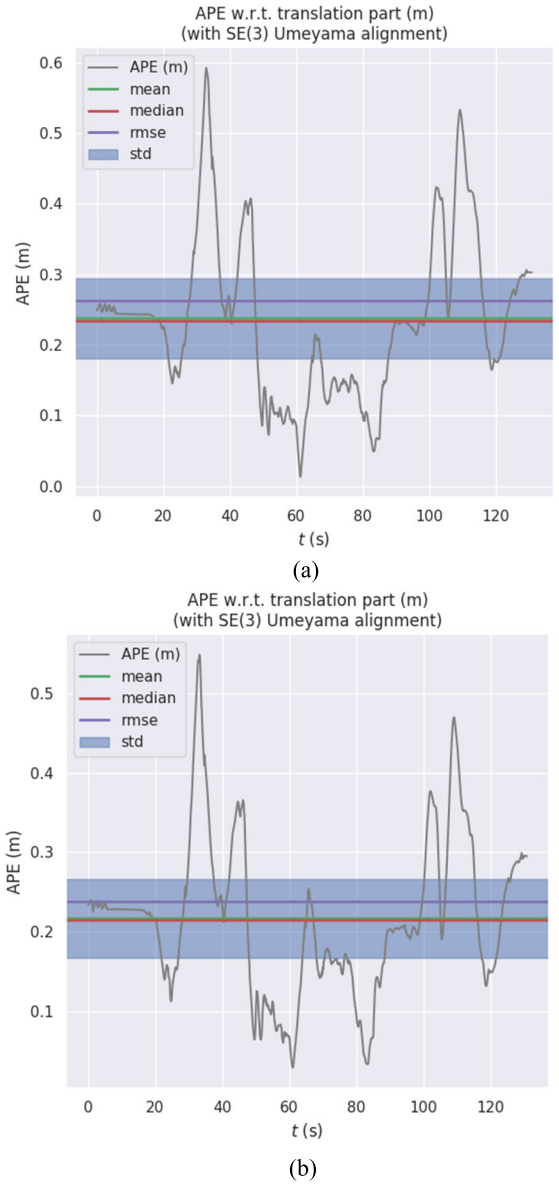


FIGURE 7. (a) is the absolute attitude error (APE) of the original algorithm, and (b) is the absolute attitude error (APE) of the improved algorithm. The black line is the absolute pose error (APE), the blue line is the root mean square error (RMSE), the red line is the median error (median), the green line is the mean error (mean), and the blue area is the standard deviation (std).

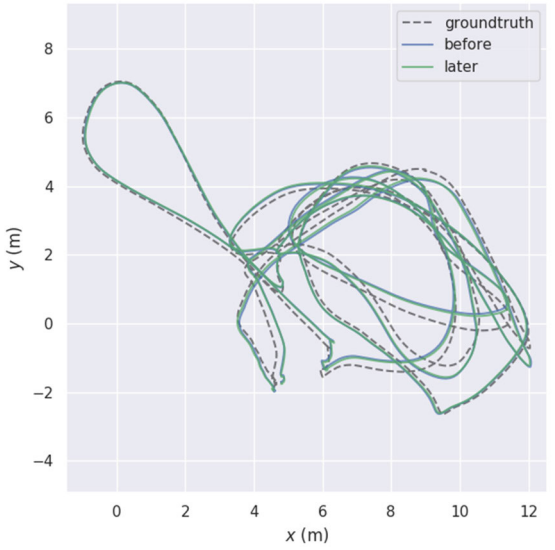


FIGURE 8. Compare the trajectories between the original algorithm and the optimized algorithm. The black line is the true trajectory, the blue line is the trajectory of the original algorithm, and the green line is the trajectory of the optimized algorithm.

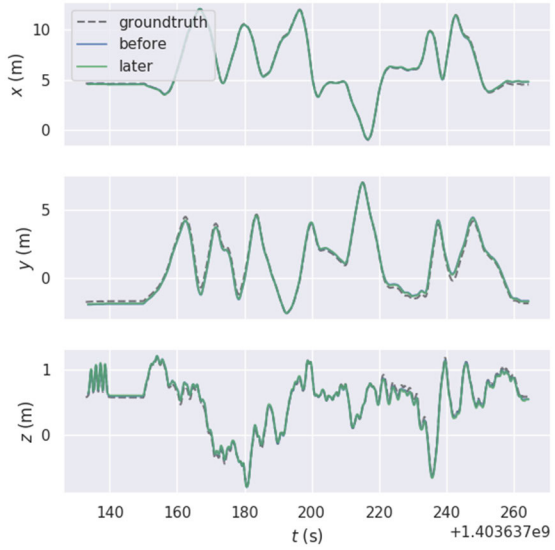


FIGURE 9. To the original algorithm compared with translation error of the optimized algorithm.

process. It not only performs gradient calculations on images, but also performs operations such as segmented search and parameter space clustering, which are relatively more time-consuming processes.

V. DISCUSSION AND ANALYSIS

This paper proposes a real-time optimized monocular visual-inertial SLAM method (PL-VINS) to optimize point and line features, which improves point features, line features and reduces pose estimation error. Specifically, we first use subpixel corner extraction algorithm instead of corner extraction algorithm, then use FLD line feature extraction algorithm instead of LSD line feature extraction algorithm, and finally delete the repeated detection lines in the process of online feature extraction.

The main realization method of using subpixel corner extraction algorithm to replace Harris corner extraction algorithm is as follows: Firstly, the function is used to obtain the number of corner points to be optimized and the initialization result, and then the subpixel corner algorithm is used to iterate and improve the accuracy of the initial value, and the edge detection of subpixel corner points is carried out according to the detection result. Since the Harris corner extraction algorithm extracts corner points at the pixel level, subpixel corner extraction is adopted to improve the accuracy

TABLE 2. The absolute pose error (APE) of the PLS-VINS algorithm and the improved algorithm is compared. (Unit: meter).

	PLS-VINS	Improved
MH_01_easy	0.16	0.14
MH_02_easy	0.14	0.13
MH_03_medium	0.24	0.22
MH_04_difficult	0.35	0.30
MH_05_difficult	0.35	0.31
V1_01_easy	0.12	0.12
V1_02_medium	0.31	0.30
V1_03_difficult	0.29	0.25
V2_01_easy	0.12	0.12
V2_02_medium	0.22	0.20
V2_03_difficult	0.25	0.24

TABLE 3. The speed of line feature extraction is compared between the original algorithm and the improved algorithm.

	Original	Improved
MH_01_easy	6.29	6.02
MH_02_easy	7.50	4.97
MH_03_medium	8.76	5.54
MH_04_difficult	7.12	5.63
MH_05_difficult	8.10	5.47
V1_01_easy	4.33	5.82
V1_02_medium	7.21	4.58
V1_03_difficult	3.66	3.32
V2_01_easy	4.24	5.11
V2_02_medium	5.62	4.86
V2_03_difficult	9.68	4.75

of corner extraction and achieve better performance in the algorithm.

In the main process of redundant line deletion, firstly, the line features are initially screened and their size is set. Later, the number of line merges can be adjusted and the line features are structured, that is, the three-dimensional position of the line features is judged under the Manhattan coordinate system, and then the line features are traversed. There are three conditions for setting merges: 2d plane, three-dimensional distance judgment and line feature id. First of all, the distance between two lines in the three-dimensional Manhattan coordinate system can be parameterized, which is set as 0.1 in this paper. Then the observation line id is detected, only the original id may be a duplicate line of the new id. Finally, the distance of 2d plane lines is judged. After the detection is complete, the merged features are deleted to complete the deletion of redundant lines. In the process of online feature extraction, the same line is repeatedly detected in the previous frame and the next frame, and the detected line will become a new line, which will increase the pose estimation error. Therefore, our optimization method is to delete redundant lines for the triangulated line features. At the beginning, we considered two schemes to delete the redundant lines. The first one was to merge the redundant lines when the FLD line feature algorithm extracted the line features. This scheme would terminate the program directly, and it was inferred that the tracking was not satisfied due to insufficient line features at the beginning of the program. Therefore, we adopt the second scheme to triangulate the line features and then merge them, which can effectively reduce the pose estimation error.

TABLE 4. The absolute pose error (APE) of the original algorithm and the improved algorithm is compared (Unit: meter).

	Original	Improved	Difference value
MH_01_easy	0.18	0.14	0.04
MH_02_easy	0.18	0.13	0.05
MH_03_medium	0.27	0.22	0.05
MH_04_difficult	0.36	0.30	0.06
MH_05_difficult	0.38	0.31	0.07
V1_01_easy	0.14	0.12	0.02
V1_02_medium	0.31	0.30	0.01
V1_03_difficult	0.31	0.25	0.06
V2_01_easy	0.12	0.12	0
V2_02_medium	0.25	0.20	0.05
V2_03_difficult	0.28	0.24	0.04
Average	\	\	0.04

A comparison experiment between the PL-VINS algorithm and the optimization algorithm is carried out on the open data set EuRoC. Firstly, the accuracy of the original algorithm and the optimized algorithm is compared. In most EuRoC data sets, the accuracy of the optimized algorithm is higher than that of the original algorithm, and the overall accuracy is improved by 0.04. As shown in Table 4.

According to the data in the table, it can be seen that Harris corner algorithm is replaced by subpixel corner algorithm and redundant lines are deleted, which effectively improves the accuracy of the algorithm. The data show that the accuracy of the optimized algorithm has been significantly improved in most scenarios. It can be seen that the accuracy of MH_05_difficult data set is the highest, which is improved by 0.07.

A comparison experiment between the PLS-VINS algorithm and the optimization algorithm is carried out on the open data set EuRoC. As shown in Table 5.

According to the data in the table, it can be seen that Harris corner algorithm is replaced by subpixel corner algorithm and redundant lines are deleted, which effectively improves the accuracy of the algorithm. The data show that the accuracy of the optimized algorithm has been significantly improved in most scenarios. It can be seen that the accuracy of MH_04_difficult data set is the highest, which is improved by 0.05.

In summary, it can be found that the improved algorithm has significantly improved in difficult data sets, which indicates that the positioning accuracy of the improved algorithm will be higher than that of the original algorithm in the case of weak texture and fast motion speed. In practical applications, robots often encounter harsh environments and scenes with

TABLE 5. The absolute pose error (APE) of the PLS-VINS algorithm and the improved algorithm is compared (Unit: meter).

	PLS-VINS	Improved	Difference value
MH_01_easy	0.16	0.14	0.02
MH_02_easy	0.14	0.13	0.01
MH_03_medium	0.24	0.22	0.02
MH_04_difficult	0.35	0.30	0.05
MH_05_difficult	0.35	0.31	0.04
V1_01_easy	0.12	0.12	0
V1_02_medium	0.31	0.30	0.01
V1_03_difficult	0.29	0.25	0.04
V2_01_easy	0.12	0.12	0
V2_02_medium	0.22	0.20	0.02
V2_03_difficult	0.25	0.24	0.01
Average	\	\	0.02

TABLE 6. The speed of line feature extraction is compared between the original algorithm and the improved algorithm.

	Original	Improved	Difference value
MH_01_easy	6.29	6.02	0.27
MH_02_easy	7.50	4.97	2.53
MH_03_medium	8.76	5.54	3.22
MH_04_difficult	7.12	5.63	1.49
MH_05_difficult	8.10	5.47	2.63
V1_01_easy	4.33	5.82	-1.49
V1_02_medium	7.21	4.58	2.63
V1_03_difficult	3.66	3.32	0.34
V2_01_easy	4.24	5.11	-0.87
V2_02_medium	5.62	4.86	0.76
V2_03_difficult	9.68	4.75	4.93
Average value	\	\	1.5

fast motion speed. Therefore, the improved algorithm can make the robot better used in the actual scene.

On the public data set EuRoC, compared with the original algorithm and the optimized algorithm, the overall average speed of line feature extraction is shortened by 1.5ms. As shown in Table 6.

According to the data in the table, it can be concluded that using FLD line feature extraction algorithm instead of LSD line feature extraction algorithm can improve the speed of line feature extraction and shorten the extraction time. With better real-time. Faster line feature extraction times mean that the robot can process sensor data faster and extract line features in the environment. And the improvement of real-time can help robots make decisions and plans faster, thus improving their ability to cope in dynamic environments. In scenes with a large number of straight lines, such as urban environments or indoor environments, faster line feature extraction can help robots build environment maps and plan paths more quickly. Faster line feature extraction allows the robot to respond more quickly to changes in the environment, such as quickly detecting emerging obstacles or the movement of obstacles. A faster response to changes in the environment helps improve the safety and adaptability of the robot, enabling it to better move and interact in dynamic environments.

FLD algorithm is mainly based on the gradient information of the image to quickly detect the line. It uses the edge information in the image to identify possible lines by calculating the gradient amplitude and direction. And the FLD algorithm usually outputs the parameterized equation of the line (such as slope and intercept), which is simple and clear. FLD algorithms generally have fewer parameters to adjust and are therefore easier to use. It may only need to set some simple thresholds to control the results of the line detection. In contrast, the LSD algorithm requires a more complex computation process. It not only performs gradient calculations on images, but also performs operations such as segmented search and parameter space clustering, which are relatively more time-consuming processes. And the LSD algorithm outputs the line segment directly, which means it needs to maintain the end points of the line segment during detection, not just the parameters of the line. This increases the complexity and computational burden of the detection process.

In the experiment, it is found that for the redundant line removal operation, the direct merging after the online feature extraction will lead to the direct termination of the program. It is inferred that the program does not meet the tracking requirements due to insufficient line features at the initial stage. Some parameters need to be adjusted on some data sets, here you can adjust the parameters according to your own experience, mainly considering adding more line features. In a smooth scene, you can merge by modifying the parameters to add more line features.

VI. CONCLUSION

PL-VINS algorithm has good performance in the point-and-line VIO algorithm, but in some special complex scenes, there are some shortcomings in real-time and accuracy. The purpose of this study is to solve the shortcomings of the current PL-VINS algorithm, propose an optimization scheme for the PL-VINS algorithm, and verify the feasibility of the scheme through comparative experiments.

In this paper, the first measure to optimize PL-VINS is to increase the speed of line feature extraction. Using FLD line feature extraction algorithm instead of LSD line feature extraction algorithm, it makes up for the shortcoming of slow feature extraction speed, and improves the real-time performance of PL-VINS algorithm significantly. Second, we use subpixel corner extraction algorithm instead of Harris corner extraction algorithm to improve the recognition accuracy of the algorithm. The third step is that in the process of online feature extraction, the same line is repeatedly detected in the previous frame and the next frame, and the detected line will become a new line, which will increase the pose estimation error. Our optimization method is to delete redundant lines for the triangulated line features to improve the accuracy of the algorithm.

In this paper, the public data set EuRoC is used to carry out comparative experiments, and the results show that the optimized algorithm has a better speed of online feature

extraction and recognition accuracy. In the future, we will compare each point-and-line visual inertial SLAM algorithm with PL-VINS algorithm in all aspects, find out the shortcomings of PL-VINS algorithm, and optimize it. At the same time, we will carry out practical application research of the optimized algorithm, find out the existing problems in the algorithm, and optimize it. Further, in future experiments, the optimized algorithm will be applied to more scenarios, from which the shortcomings of the algorithm under more constraints will be found. In view of these shortcomings, a possible solution is proposed and the algorithm is further studied.

REFERENCES

[1] 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.

[2] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[3] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017.

[4] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2510–2517.

[5] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 422–429, Apr. 2020.

[6] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1689–1696.

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[8] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4666–4672.

[9] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu, "Trifo-VIO: Robust and efficient stereo visual inertial odometry using points and lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 3686–3693, doi: 10.1109/IROS.2018.8594354.

[10] 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.

[11] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Visual-inertial odometry with point and line features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2447–2454.

[12] H. Wen, J. Tian, and D. Li, "PLS-VIO: Stereo vision-inertial odometry based on point and line features," in *Proc. Int. Conf. High Perform. Big Data Intell. Syst.*, May 2020, pp. 1–7.

[13] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4503–4508.

[14] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4225–4232.

[15] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, "Monocular visual-inertial state estimation for mobile augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2017, pp. 11–21.

[16] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1197–1204.

[17] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, and H. Zhang, "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," 2020, *arXiv:2009.07462*.

[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15, 1988, p. 5244.

[19] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 5550–5557.

[20] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[22] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019, doi: 10.1109/TRO.2019.2899783.

[23] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "StructVIO: Visual-inertial odometry with structural regularity of man-made environments," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 999–1013, Aug. 2019.

[24] 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.

[25] Y. Zhu, R. Jin, T.-S. Lou, and L. Zhao, "PLD-VINS: RGBD visual-inertial SLAM with point and line features," *Aerosp. Sci. Technol.*, vol. 119, Dec. 2021, Art. no. 107185.

[26] Z. Kuang, W. Wei, Y. Yan, J. Li, G. Lu, Y. Peng, J. Li, and W. Shang, "A real-time and robust monocular visual inertial SLAM system based on point and line features for mobile robots of smart cities toward 6G," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1950–1962, 2022, doi: 10.1109/OJCOMS.2022.3217147.

[27] H. Luo, G. Li, D. Zou, K. Li, X. Li, and Z. Yang, "UAV navigation with monocular visual inertial odometry under GNSS-denied environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 1001615, doi: 10.1109/TGRS.2023.3323519.

[28] G. Yang, Q. Wang, P. Liu, and C. Yan, "PLS-VINS: Visual inertial state estimator with point-line features fusion and structural constraints," *IEEE Sensors J.*, vol. 21, no. 24, pp. 27967–27981, Dec. 2021, doi: 10.1109/JSEN.2021.3123973.

[29] Z. Zhao, T. Song, B. Xing, Y. Lei, and Z. Wang, "Pli-vins: Visual-inertial slam based on point-line feature fusion in indoor environment," *Sensors*, vol. 22, no. 14, p. 5457, 2022.

XIGUANG ZHANG was born in January 1977. He is currently an Associate Professor and a Master Tutor. He is the Director of the Department of Software Engineering, a Henan Province Excellent Teacher, a Henan Province Civilization Teacher, and a Henan Province Excellent Dissertation Advisor. His research interests include software engineering, artificial intelligence, intelligent information processing, and visualization.

RENHAN WU received the bachelor's degree in software engineering from Zhongyuan University of Technology, in 2017, and the master's degree in computer technology from Ningxia University, in 2023. He has published an article in SCI journals. His current research interests include machine vision and robot navigation.