International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 3, Year 2026
DOI: https://doi.org/10.62051/ijcsit.v8n3.15
Journal homepage: https://wepub.org/index.php/IJCSIT/index

A Visual SLAM System Based on Point-Line Fusion

Lanqing Zhang, Lili Yuan *

School of Physics and Electronic Information, Henan Polytechnic University, Jiaozuo, China
*Corresponding Author

ABSTRACT

To address the issues of insufficient feature points, drift in pose estimation, and poor tracking stability in pure point-based visual Simultaneous Localization and Mapping (SLAM) systems under conditions such as low texture, varying lighting, and high-speed camera motion, this paper proposes a visual SLAM system that integrates point and line features. By leveraging the strong robustness of the SuperPoint feature extractor and the strong structural capabilities of end-to-end bounding box parsing, the system compensates for feature information loss through a multi-scale feature pyramid and an adaptive feature filtering mechanism, thereby improving the accuracy of pose estimation. Multiple comparative experiments were conducted on public standard datasets such as EuRoC, Tartanair, and UMA, the results demonstrate that the proposed algorithm can stably extract sufficient features and perform continuous tracking in scenarios with weak texture, fluctuating illumination, and rapid camera motion. Compared to other mainstream systems with similar performance, the absolute trajectory error and relative pose error are significantly reduced, and the overall localization accuracy and robustness of the system are effectively improved, better meeting the autonomous localization and mapping needs of mobile robots in indoor structured environments.

KEYWORDS

Visual SLAM; Point-line feature fusion; Line feature extraction; Pose estimation

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a core technology for enabling autonomous perception, navigation, and interaction in fields such as autonomous mobile robots, unmanned aerial vehicles (UAVs), autonomous driving, augmented reality, and virtual reality [1]. SLAM technology enables a platform to simultaneously estimate its own pose and construct a map of its surroundings in unknown environments without relying on external positioning infrastructure, serving as the fundamental foundation for intelligent autonomous movement. Based on the type of sensors used, SLAM can be broadly categorized into two types: laser SLAM and visual SLAM. Laser SLAM relies on LiDAR to acquire depth information about the environment; while it offers stable positioning, it is characterized by high equipment costs and a lack of textural information in the data; Visual SLAM uses monocular, stereo, or RGB-D cameras as its primary sensors. It offers advantages such as low cost, lightweight design, rich texture information, and strong scene adaptability, making it the mainstream positioning solution for devices such as indoor robots and small drones. It has been widely applied and studied in scenarios such as industrial inspections, home services, and smart spaces [2].

Currently, most mainstream visual SLAM systems are based on point features. They extract corner features such as FAST, ORB, SIFT, and SURF to perform inter-frame feature matching, motion estimation, back-end optimization, and loop closure. Among these, ORB-SLAM3, as the most

representative open-source visual SLAM framework, integrates monocular, stereo, RGB-D, and visual-inertial tightly-coupled modes. It offers high accuracy, strong robustness, relocalization capability, and support for multiple maps, and performs exceptionally well in texture-rich static environments [3]. However, in real-world indoor structured environments, areas such as walls, corridors, stairwells, and empty rooms commonly exhibit weak texture, resulting in a severe shortage of extractable corner features. Additionally, issues such as sudden changes in illumination intensity, shadows, reflections, and motion blur caused by high-speed camera movement further lead to failed feature extraction and increased matching error rates. These issues directly cause drift in the system's pose estimation and a decline in tracking accuracy, and may even result in tracking loss or mapping failure, severely limiting the practicality and reliability of visual SLAM in complex static scenes [4][6].

To address the limitations of pure point-based visual SLAM, researchers have incorporated line features into the SLAM framework. As structural features that are abundant in man-made environments, line features exhibit characteristics such as invariance to lighting, stability across viewpoints, and strong edge continuity. They can be reliably extracted even in low-texture regions, effectively supplementing constraint information when point features are missing, and thereby improving the system's tracking stability and localization accuracy in complex scenes [7][9]. Early point-line fusion SLAM algorithms typically employed the LSD algorithm to extract line features and the LBD descriptor for feature matching. However, traditional line feature methods suffer from issues such as high computational complexity, redundancy in short line segments, poor endpoint stability, and sensitivity to lighting changes. Direct fusion significantly increases the system's computational load and reduces real-time performance. Furthermore, simply superimposing point and line features leads to an imbalance in optimization weights, preventing the full enhancement potential of line features from being realized [10][12].

Currently, point-line fusion visual SLAM has emerged as a key approach for improving localization performance in low-texture and variable-illumination scenarios; however, existing methods still suffer from issues such as insufficient robustness in line feature extraction, simplistic point-line fusion strategies, and low optimization efficiency. To address these issues, this paper proposes an improved point-line fusion visual SLAM algorithm.

2. RELATED WORK

Research on point-line fusion visual SLAM focuses on three core areas: point-based SLAM, line feature extraction, and point-line fusion strategies. In recent years, a large number of improved methods have emerged, continuously enhancing the system's performance in complex static scenes characterized by low texture and varying lighting conditions.

2.1. Research on Point-Based Visual SLAM

Pure point-based visual SLAM relies on corner features for frame-to-frame matching and pose estimation, serving as the foundational approach for visual SLAM. Early representative algorithms such as MonoSLAM and PTAM were the first to achieve real-time monocular localization and mapping, laying the groundwork for subsequent research [13]. ORB-SLAM2 introduced ORB features and established a three-thread architecture comprising tracking, local mapping, and loop closure detection, achieving breakthroughs in both accuracy and real-time performance. ORB-SLAM3 further supports tight visual-inertial coupling, map reuse, and multimodal fusion, making it the most widely used open-source baseline system today. Although pure point-feature SLAM performs excellently in texture-rich scenes, under conditions such as low texture, uneven lighting, and motion blur, point features are prone to missing or mis-matching, leading to pose drift and making it difficult to meet the demands of complex environments [14].

2.2. A Study on Line Feature Extraction and Matching Algorithms

Line features effectively compensate for the limitations of point features in structured environments, and the quality of their extraction and matching directly determines the performance of point-line fusion SLAM. LSD (Line Segment Detector) is a classic real-time line segment detection algorithm capable of extracting line segments without relying on parameter tuning; however, it tends to generate a large number of redundant short line segments and is sensitive to changes in lighting conditions [15]. LBD (Line Band Descriptor) is a commonly used line feature descriptor capable of performing line segment matching; however, its high computational complexity impacts system real-time performance. To enhance line feature performance, the ELSED algorithm optimizes edge connection and line segment fitting, significantly improving detection speed and line segment continuity; however, it still suffers from detection omissions in scenarios with drastic changes in lighting conditions. Subsequent research has improved line feature extraction through strategies such as adaptive gradient thresholding, local brightness compensation, short-line merging, and length filtering, making it better suited to the real-time and robustness requirements of SLAM.

2.3. Research on Point-Line Fusion Visual SLAM

Point-line fusion significantly improves system performance in low-texture environments by combining the localization accuracy of point features with the structural stability of line features. PL-SLAM was the first to apply point-line feature fusion to stereo SLAM, using a combination of LSD and LBD for line feature processing; however, this approach incurs high computational overhead, making it difficult to run in real time on low-end devices [16]. PL-VIO introduces point-line fusion based on VINS-Mono, improving the robustness of visual-inertial odometry; however, errors in the endpoints of line features still affect the optimization results [17]. To address the issue of weight imbalance, some researchers have proposed adaptive point-line fusion strategies that dynamically adjust the weights of line features based on the number of point features, strengthening the constraints of line features when point features are insufficient, thereby effectively mitigating tracking drift. In recent years, point-line fusion schemes based on improved ELSED, adaptive thresholding, and fast line segment triangulation have emerged, further improving localization accuracy in low-texture and variable-illumination scenarios. However, there is still room for improvement in terms of computational efficiency, high stability, and strong generalizability.

In summary, the integration of point and line features is an effective approach for improving the performance of visual SLAM in scenes with weak static textures and varying lighting conditions. While existing methods have demonstrated the importance of line features, there is still room for improvement in the collaborative design of point and line features. Therefore, this paper focuses on static scenes as a constraint and conducts research on the optimization of line features and the efficient integration of point and line features to develop a visual SLAM algorithm with higher accuracy and robustness.

3. SYSTEM ARCHITECTURE

3.1. System Workflow

This point-line fusion visual SLAM system uses left and right images captured by a stereo camera as its raw input. The images first enter the front-end feature processing module, where joint extraction of point and line features is performed in parallel. This yields sparse point features based on vertices and line segment features based on edge structures. The system then calculates descriptors for both types of features and performs matching operations to establish geometric correspondences between point and line features across frames and between stereo views; Based on the matched point-line feature pairs, the system proceeds to the pose tracking thread. It first performs an initial pose estimate

for the current frame using feature projection constraints, then associates the observed features with existing map points and lines in the local map. By minimizing the joint re-projection error of points and lines, the system achieves local map tracking and fine-tunes the pose. Subsequently, key frames are generated by filtering based on the magnitude of pose changes and feature tracking quality to control computational load; After new keyframes are generated, triangulation is performed on the matched point and line features to generate new 3D map points and 3D map lines and update the local map. Simultaneously, historical keyframes are retrieved via database queries, and the transformation between the current frame and historical frames is calculated to perform loop closure detection; When loop closure is detected or optimization conditions are met, the system performs global beam adjustment (BA) optimization on the set of keyframes associated via the common view, jointly minimizing the reprojection errors of point and line features to achieve global consistency correction of pose and map; This ultimately completes the map update, providing more stable local and global constraints for tracking subsequent frames, thereby enabling high-precision, highly robust visual simultaneous localization and mapping (SLAM) in complex static scenes characterized by weak textures and changing lighting conditions.

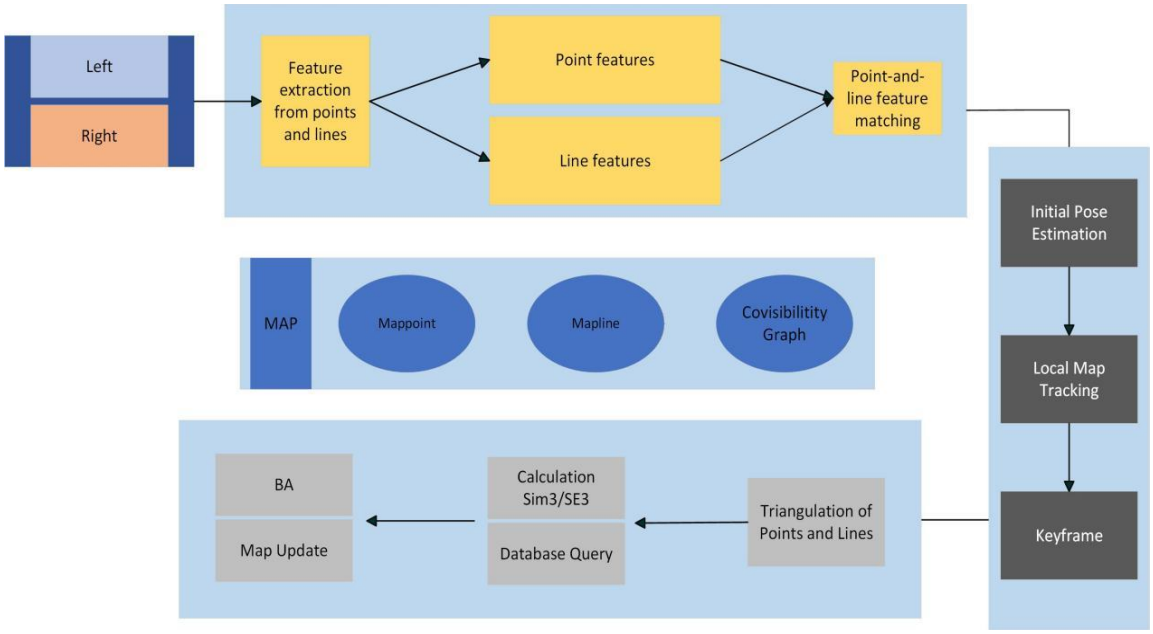

Figure 1. System Architecture Diagram

3.2. Point-line Fusion Method

To extract rich feature information in low-texture and dynamic environments, we have designed a new point-line fusion method. This method consists of two main components: point feature extraction, which includes point detection and descriptor generation; The second component is a bounding box parsing method, which is used to detect line segments in the image and extract line features. As shown in Figure 2, the entire point-line fusion method combines a self-supervised framework for point extraction with an end-to-end bounding box parsing method, enabling the system to effectively extract both point and line features and ensuring accurate tracking.

In the pursuit of building efficient systems, it is crucial to reduce the difficulty of data acquisition and enhance the system's generalization ability and robustness. Therefore, this paper adopts the SuperPoint [18] architecture to design the backbone network, utilizing an encoder with a reduced number of deep layers based on VGG [19] to perform dimensionality reduction on images, outputting a feature map with a resolution of $H/8 \times W/8 \times 65$. For the point detection component, a feature point detection head with an explicit decoder is employed. Through a hierarchical structure of upsampling and convolution, the feature map is restored to a resolution close to that of the input image. (The

descriptor generation module establishes feature point matching relationships by constructing transformations between images, utilizing bicubic interpolation and L2 normalization to achieve effective representation and association of features.) Descriptor generation involves constructing geometric transformations between images, performing pixel resampling using bicubic interpolation, and standardizing feature vectors via L2 normalization. This process accurately establishes matching mappings between feature points across images, enabling semantic representation and structured association of features.

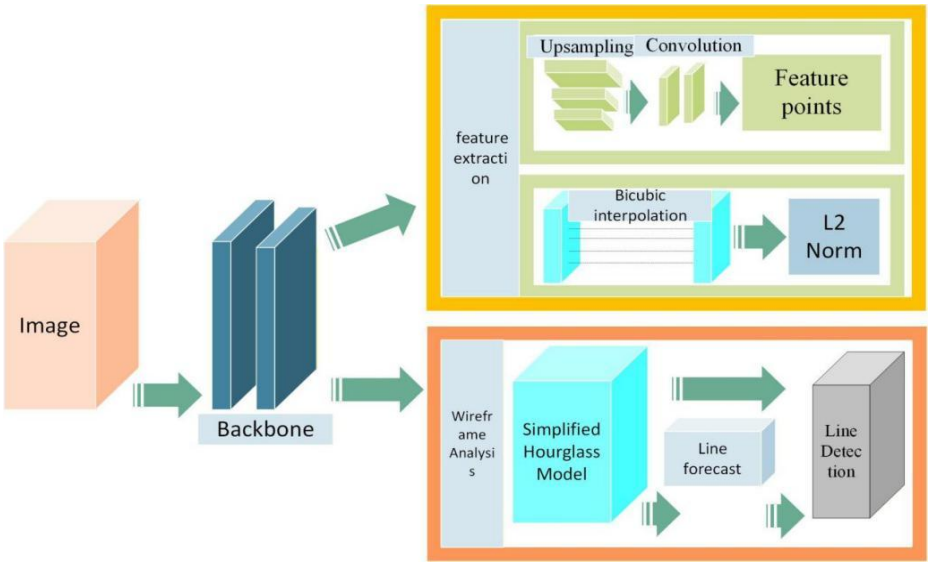

Figure 2. Diagram of the Point-Line Fusion System

3.3. Line Feature Extraction

The line frame analysis method not only detects line segments but also identifies the endpoints on these segments, thereby fully capturing the line frame information in the image and providing richer geometric data. To better integrate point and line features, the line frame analysis in this paper combines the endpoint decoupling of EPD LOIAlign with the line sampling of L-CNN, and improves the system's line feature extraction speed and information content through structural optimization. The wireframe analysis method is illustrated in Figure 3 and consists of three components: a simplified hourglass network, line prediction, and line detection.

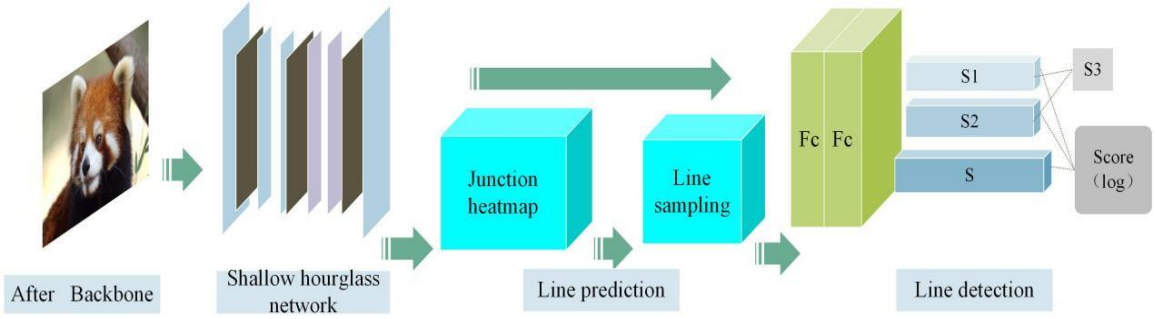

Figure 3. Wireframe Analysis Framework Diagram

The overall depth of an hourglass network increases with the number of stacked hourglass layers, which negatively impacts the system's real-time performance. However, since this model largely follows a process of restoring high-resolution, detailed features from low-resolution, high-level features, this paper opts to use a shallow hourglass network. The network employs a balanced architecture consisting of three downsampling layers and three upsampling layers. The number of parameters has been reduced from 5-10 million to 0.7-1.5 million, significantly lowering computational complexity and improving inference speed. The use of a shallow Hourglass network

not only reduces the number of downsampling and upsampling layers but also maintains a certain level of multiscale feature extraction capability.

Research indicates that most endpoint points of line segments in line detection models are also identified as keypoints by keypoint detection models. Based on this, this paper proposes a novel line construction method. This method integrates keypoints from point detection, connection point heatmaps, line sampling, and line detection techniques for line construction, prediction, and feature extraction. The method not only selects more representative line segments but also effectively fuses point features with line features.

After passing through the backbone, the system extracts features to generate a feature map. This feature map is then processed through a shallow hourglass model to produce a connection point heatmap. When predicting connection points, the image with resolution H×W is divided into Hb×Wb regions. Within each region, a convolutional neural network (CNN) is used to select a connection point and output its position. Mathematically, the output of this process includes a feature map J of connection point likelihood probabilities and a position offset map D, as shown in Equations (1) and (2). Here, b represents the center coordinates of the region, V is the index of the set of connection points, and Pi denotes a point within the set of connection points.

$$J(b)=\begin{cases}1 & \exists i\in V, Pi\in b\\ 0 & otherwise\end{cases} \tag{1}$$

$$D(b)=\begin{cases}\dfrac{(b-Pi)}{W_b} & \exists i\in V, Pi\in b\\ 0 & otherwise\end{cases} \tag{2}$$

b represents the coordinates of the regional center, V is the index of the set of connection points, and Pi is a point in the set of connection points.

In the prediction of J, this is treated as a classification problem, calculated using the average binary cross-entropy loss, and redundant prediction points are removed using non-maximum suppression (NMS) (Equation (3)). For the prediction of D, an L2 regression method is used to ensure that the offset range falls within [-1/2,1/2)×[-1/2,1/2). The prediction yields the optimal set of candidate connection points, T. Here, N(b) denotes the 8 regions surrounding b.

$$J^{'}(b)=\begin{cases}J(b) & J(b)=\max_{b^{'}\in N(b)}J(b^{'})\\ 0 & otherwise\end{cases} \tag{3}$$

Subsequently, line sampling is performed based on the set T, connecting pairs of nodes in T to form a set of line segments L(T1, T2), where T1 and T2 represent the endpoints of the line segments and T1, T2∈T. However, most of the line segments generated in this way are negative samples, which reduces the efficiency of the line detection component. To address this issue, this paper employs a method that combines static and dynamic sampling for pre-training. An equal number of connection points are sampled from both positive and negative samples for training. Additionally, a certain number of negative samples that overlap with positive samples are retained to enhance robustness.

In previous studies on online detection, features at non-grid intersections were often quantified; this operation tends to cause error accumulation, leading to a decrease in detection accuracy. To obtain precise line features, this paper directly inputs the endpoints of line segments for line verification processing. Specifically, this section receives the set of line segments L output from line sampling

and the feature maps processed by the shallow hourglass network. It then takes the two endpoints of the line segment as input, decouples the relationship between the endpoints and the midpoint, and uses bilinear interpolation to obtain a feature vector with a corrected length. The feature vector is then fed into a network consisting of two fully connected layers (FC) for processing. After processing, the network outputs three feature vectors S1, S2, and S, which represent different relationships between the endpoints and the midpoint. Subsequently, a score for evaluating the line segment is calculated using Equations (4) and (5). Finally, representative line segments are retained using NMS.

$$Score = L[MLP(S3) + MLP(S)] \quad S3 \in \{S1, S2\} \tag{4}$$

$$L = -[y\log(p) + (1-y)\log(1-p)] \tag{5}$$

In the equation, y represents the true label, p represents the predicted probability, MLP stands for Multi-Layer Perceptron, and L denotes a linear transformation. By taking the logarithm of p and 1-p, we transform these probabilities to avoid numerical instability during computation. The point-line fusion method proposed in this paper leverages the fact that key points in point extraction coincide with the endpoints of line segments in traditional line detection models. By fusing point extraction and line extraction, this method not only facilitates faster and more efficient line construction but also enriches the extracted feature information, thereby ensuring the system's accuracy and robustness.

4. EXPERIMENT

All experiments in this paper were conducted on a computer configured with 32 GB of RAM, an Intel Core i9-14900HX processor, and an NVIDIA GeForce RTX 4060 Laptop GPU, running Ubuntu 20.04.For the datasets, we selected the UMA, Tartanair, and EuRoc datasets to evaluate the system's performance in low-texture and dynamic environments. To ensure the stability of the experimental results, each sequence was tested multiple times. For evaluation, Absolute Trajectory Error (ATE), Relative Position Error (RPE), and trajectory comparison plots were used as metrics to assess system performance. ATE measures the global deviation between the estimated trajectory and the ground truth trajectory, making it suitable for evaluating the cumulative error of a SLAM system after prolonged operation. RPE represents the error in pose changes between adjacent frames and is used to assess the system's drift over time. In the experiments, an "F" indicates that the system did not complete the experiment for that sequence, and bold text indicates that the system with that score performed best in that sequence. The table uses RMSE values to evaluate performance; the smaller the value, the better the performance.

Table 1. Comparison Results of Various Systems on the UMA Dataset

Algorithm Sequence	ORB-SLAM3	ORB-LINE-SLAM	ORB&ELSED	AIRSLAM	Ours
	ATE(RMSE)				
class-csc1	F	0.0517	0.0764	0.0241	0.0239
Class-csc2	F	0.0518	0.0509	0.0824	0.0408
Corridor	0.2152	0.0279	0.0208	0.0227	0.0198
Fantasy-csc1	0.0909	0.0598	0.0367	0.0202	0.0179
Hall-rev	0.2888	0.1033	0.0604	0.0973	0.0726

First, for low-texture scenes, this paper selected five sequences from the UMA dataset—Class-csc1, Class-csc2, Corridor, Fantasy-csc1, and Hall-rev to conduct comparative experiments. The experimental results are shown in Table 1. Compared with ORB-SLAM3, ORB-LINE-SLAM, ORB&ELSED, and AIRSLAM, the system proposed in this paper achieved the best accuracy on the

first four sequences, with a particularly significant advantage in the Class-csc2 sequence (see Figure 4 for details). This sequence simulates a scenario of a person walking back and forth in a low-texture corridor environment, which poses significant challenges for the feature extraction stage. Experimental data indicate that even in such complex, noisy scenarios, the proposed system maintains stable performance, achieving a 51% improvement in localization accuracy over ORB&ELSED and a 69% improvement over AIRSLAM. Figure 5 displays the sparse point cloud generated from experiments on the UMA dataset, showing complete and continuous trajectories with clear terrain contours, further validating the system's high-precision localization capabilities.

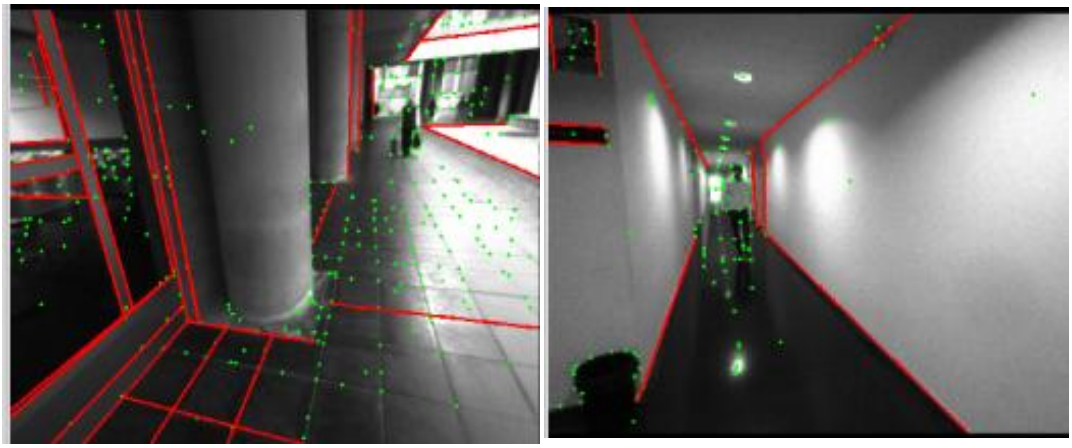

Figure 4. Rendering of point-and-line extraction

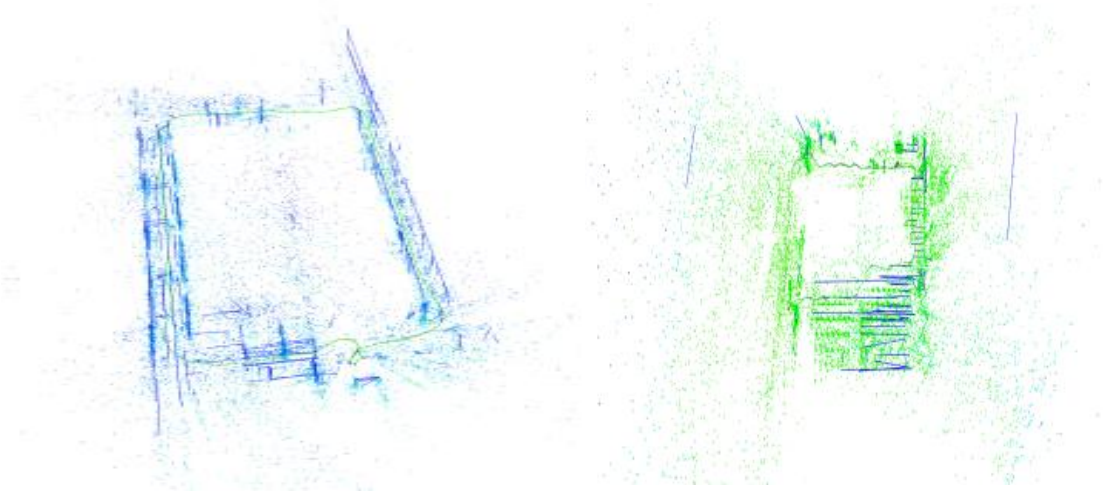

Figure 5. Point Cloud Rendering

Ablation Experiment: To evaluate the performance of the point-line fusion method proposed in this system in low-texture static environments, we conducted experiments using nine sequences from the EuRoc static dataset. The corresponding results are shown in Table 2.

As shown by the data comparison in Table 2, our system achieved the best performance across the six sequences, with an accuracy improvement of 28% over ORB-SLAM3, 20% over PL-SLAM, 58% over UV-SLAM, and 8.3% over AIRSLAM. We conducted experimental comparisons between our system and the recently popular AIRSLAM system, with the results shown in Figure 6.

Table 2. Comparison results of various systems on the EuRoc dataset

Algorithm Sequence	ORB-LINE-SLAM	ORB-SLAM3	PL-SLAM	UV-SLAM	AIRSLAM	Ours
	ATE(RMSE)					
MH01	0.038	0.036	0.0416	0.161	0.028	0.027
MH02	F	0.033	0.0522	0.179	0.035	0.031
MH03	0.041	0.047	0.0399	0.176	0.042	0.038
MH04	0.044	0.052	0.0641	0.291	0.051	0.049
MH05	0.045	0.082	0.0697	0.189	0.064	0.052
V102	F	0.069	0.0523	0.071	0.045	0.042
V103	F	0.142	0.0826	0.094	0.070	0.072
V201	0.061	0.077	0.0659	0.078	0.062	0.053
V202	0.058	0.093	0.0568	0.085	0.060	0.055

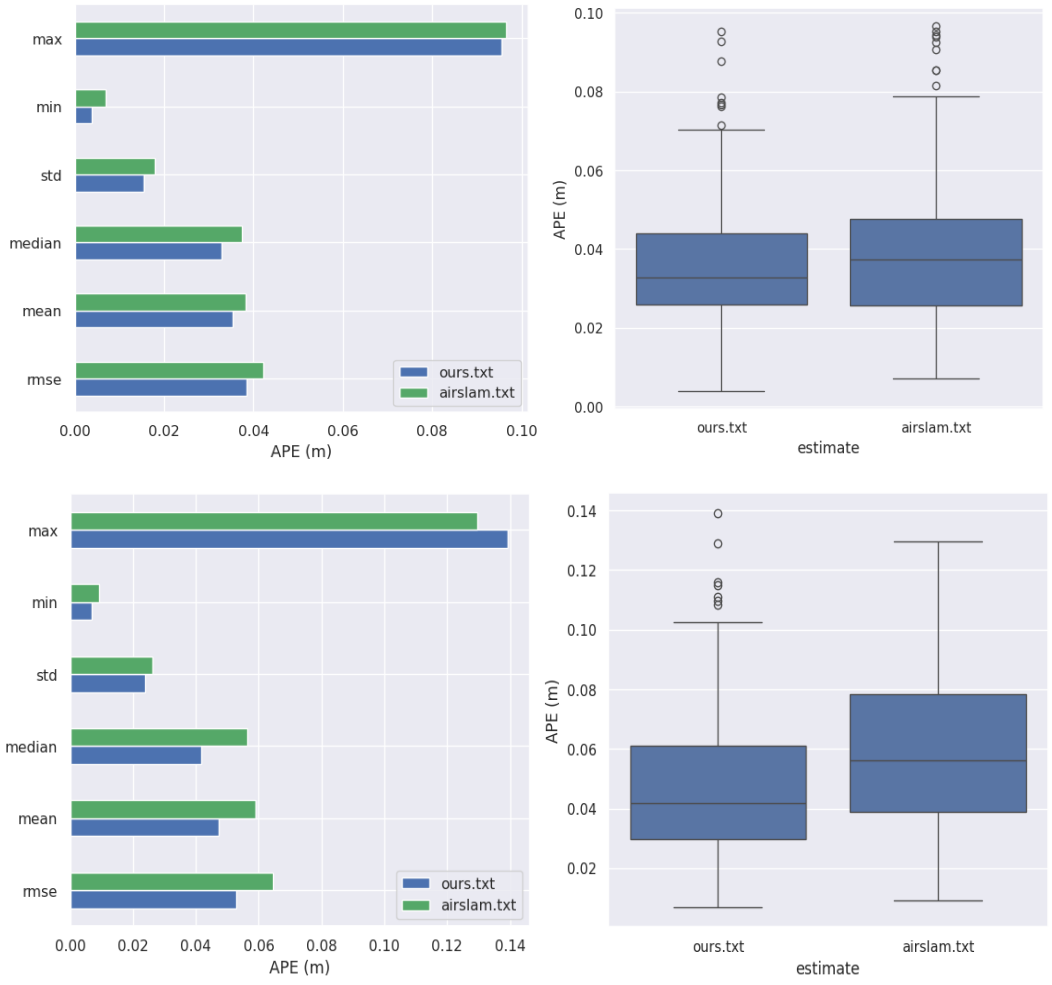

Figure 6. Graph of Experimental Results

The experimental results show that, when comparing the six APE metrics—max, min, std, median, mean, and rmse—the system described in this paper (ours.txt) significantly outperforms AirSLAM (airslam.txt), and box plots further validate the stability and superiority of the overall error distribution. This fully confirms that the point-line fusion feature extraction method proposed in this paper can effectively capture richer geometric constraint information in low-texture scenes, providing reliable support for accurate pose estimation and thereby significantly improving the system's localization accuracy and robustness.

5. SUMMARY

This paper addresses the issues of insufficient features and pose drift in pure point-based visual SLAM systems under low-texture scenes by proposing a visual SLAM system that integrates point and line features. By improving the feature extraction and fusion strategies for points and lines, the system compensates for the information loss inherent in single-point features, thereby enhancing the system's localization accuracy and robustness. Comparative experiments on public datasets such as UMA and EuRoC demonstrate that the proposed system performs exceptionally well in low-texture sequences: in the Class-csc2 highly noisy sequences of the UMA dataset, the system achieves a 51% improvement in accuracy over ORB&ELSED and a 69% improvement over AIRSLAM; on the EuRoC dataset, accuracy improved by 28% compared to ORB-SLAM3, and all error metrics outperformed the comparison algorithms, with the generated sparse point cloud trajectories being complete and clearly defined. This system effectively addresses the shortcomings of traditional pure point-feature SLAM, providing reliable technical support for autonomous localization and navigation of mobile robots in indoor structured environments. Future work can focus on further optimizing algorithm lightweighting and adaptability to dynamic scenes.

REFERENCES

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," in IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874-1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[2] Merchán-Cruz, Emmanuel A., Samuel Moveh, Oleksandr Pasha, Reinis Tocelovskis, Alexander Grakovski, Alexander Krainyukov, Nikita Ostrovenecs, Ivans Gercevs, and Vladimirs Petrovs. 2025. "Smart Safety Helmets with Integrated Vision Systems for Industrial Infrastructure Inspection: A Comprehensive Review of VSLAM-Enabled Technologies" Sensors 25, no. 15: 4834. https://doi.org/10.3390/s25154834

[3] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611-625, 1 March 2018, doi: 10.1109/TPAMI.2017.2658577.

[4] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 15-22, doi: 10.1109/ICRA.2014.6906584.

[5] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 2007, pp. 225-234, doi: 10.1109/ISMAR.2007.4538852.

[6] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[7] R. Gomez-Ojeda, F. -A. Moreno, D. Zuñiga-Noël, D. Scaramuzza and J. Gonzalez-Jimenez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," in IEEE Transactions on Robotics, vol. 35, no. 3, pp. 734-746, June 2019, doi: 10.1109/TRO.2019.2899783.

[8] H. Wen, J. Tian and D. Li, "PLS-VIO: Stereo Vision-inertial Odometry Based on Point and Line Features," 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), Shenzhen, China, 2020, pp. 1-7, doi: 10.1109/HPBDIS49115.2020.9130571.

[9] J. P. Company-Corcoles, E. Garcia-Fidalgo and A. Ortiz, "MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2803-2810, April 2022, doi: 10.1109/LRA.2022.3142900.

[10] R. Grompone von Gioi, J. Jakubowicz, J. -M. Morel and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 722-732, April 2010, doi: 10.1109/TPAMI.2008.300.

[11] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," Journal of Visual Communication and Image Representation, vol. 24, no. 7, pp. 794–805, 2013, doi: 10.1016/j.jvcir.2013.05.006.

[12] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," Pattern Recognition Letters, vol. 32, no. 13, pp. 1633–1642, 2011, doi: 10.1016/j.patrec.2011.06.001.

[13] I. Suárez, J. M. Buenaposada and L. Baumela, "ELSED: Enhanced line SEgment drawing," Pattern Recognition, vol. 127, pp. 108619, 2022, doi: 10.1016/j.patcog.2022.108619.

[14] M. Burri, J. Nikolic, P. Gohl, et al., "The EuRoC micro aerial vehicle datasets," International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163, 2016, doi: 10.1177/0278364915620033.

[15] R. G. V. Gioi, J. Jakubowicz, J. M. Morel, et al., "LSD: A line segment detector," Image Processing On Line, vol. 2, pp. 35–55, 2012, doi: 10.5201/ipol.2012.gjmr-lsd.

[16] R. Gomez-Ojeda, F. -A. Moreno, D. Zuñiga-Noël, D. Scaramuzza and J. Gonzalez-Jimenez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," in IEEE Transactions on Robotics, vol. 35, no. 3, pp. 734-746, June 2019, doi: 10.1109/TRO.2019.2899783.

[17] Y. He, Z. Ji, Y. Guo et al., "PL-VIO: Tightly-Coupled Monocular Visual–Inertial Odometry Using Point and Line Features," Sensors, vol. 18, no. 4, p. 1159, 2018, doi: 10.3390/s18041159.

[18] D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018, pp. 230-241, doi: 10.1109/CVPRW.2018.00037.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015, pp. 1–14.