PLL-VO：An Efficient and Robust Visual Odometry Integrating Point-Line Features and Neural Networks

Leyang Zhao[1], Yanguang Yang[2], Ding Ma[1], Xing Lin[1], Weixi Wang[1]
([1] School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China
[2] School of Geomatics and Geographical Sciences, Liaoning Technical University, Fuxin 123000, China)

Keywords: Visual Odometry, Line Detection, Neural Networks, Self-Supervised Learning, Lighting Conditions.

Abstract

Visual odometry is crucial for the navigation and planning of autonomous robots, but low-light conditions, dramatic lighting changes, and low-texture scenes pose significant challenges to odometry estimation. This paper proposes PLL-VO, which integrates point-line features and deep learning. To overcome the impact of complex lighting conditions, a self-supervised learning method for interest point detection and a line detection algorithm that combines line optical flow tracking with cross-constraints is presented. After selecting keyframes based on point feature counts and line feature overlap angles, we integrate convolutional neural networks (CNNs) and graph neural networks (GNNs) to enhance sparse matching, thereby improving both accuracy and computational efficiency. PLL-VO system are evaluated in multiple datasets under various lighting conditions, demonstrating a 6.3% reduction in absolute trajectory error for pose estimation compared to state-of-the-art (SOTA) algorithms, the average computation time for visual odometry (43ms) shows a 29.74% decrease compared to the state-of-the-art (SOTA) algorithms.

1 . INTRODUCTION

In Visual Simultaneous Localization and Mapping (VSLAM) systems, the front-end visual odometry (VO) subsystem determines the robot's current position and orientation by processing camera images and estimating camera motion. Traditional corner detection strategies based on optical flow rely on the intensity constancy assumption, selecting feature points based on image intensity values and performing 2D-2D feature matching using epipolar constraints to recover the camera or robot's pose. However, in visually challenging environments, such as low-light conditions, dramatic lighting changes, and weak-texture scenes, traditional intensity-based feature extraction methods do not consider the structural and semantic information of the image. The challenges in accurately recognizing and correlating feature information within the current surroundings can precipitate instability in performance and, in more intricate and shifting scenarios, potentially lead to the loss of tracking. This underscores the necessity for robust mechanisms to ensure consistent feature association and tracking accuracy across varying environmental conditions.

Deep learning-based methods have shown great potential in specific scenarios. Deep learning models can automatically learn complex features from large amounts of image data, capturing multi-level information from simple edges to complex textures and shapes. Automatic feature learning capability significantly reduces the dependence on manual feature engineering and enhances the generalizability and adaptability of feature extraction. Deep learning also enables end-to-end feature extraction and matching, directly mapping from raw images to matched features. This end-to-end approach simplifies system design and improves overall performance through joint optimization of all relevant steps.
However, existing deep learning-based visual odometry methods still have certain limitations. Initially, these methods demand a substantial dataset for training, along with significant computational resources and storage space. This requirement often results in suboptimal real-time performance on CPUs. Secondly, end-to-end deep learning approaches encounter interpretability issues, models are frequently likened to "black boxes," characterized by their lack of transparency in decision-making processes. Lastly, despite their potential, these models have not yet reached their full potential in consistently delivering accurate and robust performance across diverse scenarios, indicating a need for further refinement and optimization. For example, SupSLAM (Quach et al., 2021) replaces traditional feature point extraction with deep feature points but continues to use conventional methods for tracking these features. SuperGlue (Sarlin et al., 2020), while effectively outputting the matching relationships between feature points and descriptors in images, relies heavily on local feature extraction, which may not be robust in scenes with low texture or significant viewpoint changes.
A robust and resilient deep learning-based Visual Odometry (VO) system is proposed. Initially, introduce a line feature extraction algorithm that employs line optical flow tracking and cross-constraint strategies to overcome the limitations of point feature extraction in visually challenging scenarios. Subsequently, incorporate a self-supervised feature point detector based on the SuperPoint network (DeTone et al., 2018), which integrates point and line features to achieve robust feature extraction under adverse conditions such as low-light environments, dynamic lighting, weak-textured areas, and significant camera jitter. Keyframe selection is performed using feature points and descriptors output by the SuperPoint network, integrating a point-line feature extraction model with the LightGlue (Lindenberger et al., 2023) network model. The selected keyframes are then subjected to feature matching using a Graph Neural Network (GNN) with an attention mechanism for geometric verification, thereby reducing the likelihood of false matches. Finally, comparative experiments focusing on feature extraction and matching utilizing the EUROC dataset, comparing our proposed PLL-VO system with several state-of-the-art (SOTA) reference algorithms. The results demonstrate PLL-VO's

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-1045-2025 | © Author(s) 2025. CC BY 4.0 License.

superior performance in terms of accuracy and time complexity. The main contributions of this work are as follows:
Introduce a novel robust framework for feature extraction in visually challenging scenarios. Proposes a line feature extraction algorithm that employs line optical flow tracking and cross-constraint strategies, integrated with a self-supervised feature point detector based on the SuperPoint network.

We propose a feature matching framework that integrates a point-line feature extraction model with the LightGlue network model. By processing descriptors of feature points and lines, and selecting keyframes, LightGlue computes a matching matrix from two sets of local feature points using a neural network, thereby facilitating local feature matching. Experiments utilizing multiple datasets to validate the efficacy of point-line feature extraction and the effectiveness of feature matching.

2. RELATED WORK

2.1 Feature Extraction

The extraction of visual image features using traditional methods are widely implemented in various applications such as Scale-Invariant Feature Transform (SIFT) (Lowe, 2004), Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), Speeded-Up Robust Features (SURF) (Bay et al., 2008), and Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011). SIFT is highly regarded for its robustness to scale, orientation, and noise, albeit with a relatively high computational complexity. SURF based on the Hessian matrix, employs box filtering and image integration to rapidly compute gradients, thereby enhancing detection speed. ORB achieves rotation invariance by incorporating orientation information based on FAST detection and constructs scale invariance through the creation of image pyramids, utilizing binary BRIEF descriptors to represent features, which has found broad application in the field of computer vision. Furthermore, there are improved methods based on SIFT and SURF that aim to reduce computational load and enhance affine robustness.

2.2 Point-Line Feature Extraction

Due to the additional regularity constraints on scene structure provided by lines within the environment, the utilization of line features to enhance the performance of feature extraction has garnered attention. The Line Segment Detector (LSD) (Von Gioi et al., 2008) is a classic algorithm for extracting line features. However, LSD is designed for scene structure representation without parameter adjustment and is not specifically tailored for pose estimation problems, where numerous lines can be considered outliers. The Line Band Descriptor (LBD) (Zhang & Koch, 2013) introduces a line matching algorithm that capitalizes on the local appearance and geometric properties of line segments to enhance the efficiency and robustness of matching. Gomez-Ojeda et al. (Gomez-Ojeda & Gonzalez-Jimenez, 2018) propose a purely geometry-based method for robust line segment matching in high dynamic range (HDR) environments or in stereo sequences with severe lighting variations. The EDLines (Akinlar & Topal, 2011) algorithm utilizes the Edge Drawing approach to generate a series of clean, continuous edge pixel chains, which intuitively reflect the object boundaries, and extracts line segments from the generated pixel chains using the least squares straight line fitting method.

Meanwhile, there are more methods of combining line-point features for SLAM that have been published in recent years. Pautrat et al. introduced DeepLSD, improving line segment detection using deep image gradients (Pautrat, 2023). Zhao et al. combined line segments with structural regularities for more reliable localization (Zhao, 2023). Jeong and Lee, as well as Yuan et al., highlighted the effectiveness of combining line and corner features (Jeong, 2022; Yuan, 2021). Zhou et al. proposed StructSLAM, leveraging building structure lines for urban environments (Zhou, 2020). Zuo et al. and Xu et al. further explored robust SLAM with point and line features, with Xu et al. addressing illumination variations (Zuo, 2019; Xu, 2023). Chamorro et al. demonstrated real-time event-based line SLAM (Chamorro, 2024). These works collectively emphasize the importance of diverse feature integration in visual SLAM.

2.3 Deep Learning-Based Feature Extraction

Traditional feature extraction methods, which are incapable of perceiving geometric and structural information in scenes, still encounter challenges when processing environments with weak or flickering lighting and significant camera shake. Consequently, researchers are exploring the integration of deep learning techniques with traditional feature extraction methods.

LIFT (Yi et al., 2016) is a recently introduced convolutional alternative to SIFT, encompasses interest point detection, orientation estimation, and descriptor computation, but it still requires supervision from classical Structure from Motion (SfM) systems. QuadNetworks (Savinov et al., 2017) address the interest point detection problem from an unsupervised standpoint; however, their system is patch-based (inputting small image patches) and employs a relatively shallow 2-layer network. The TILDE (Verdié et al., 2015) interest point detection system employs a principle similar to homography adaptation; nevertheless, their method does not leverage the capabilities of large fully convolutional neural networks. SuperPoint introduces a self-supervised solution that eschews human supervision to define interest points in real images. It begins by training a fully convolutional neural network on millions of images, creating a synthetic dataset named "synthetic Shapes," and developing a multi-scale, multi-transform technique—homography adaptation.

2.4 Traditional Visual Odometry

Traditional visual odometry (VO) methods estimate the camera's motion trajectory using consecutive images captured by a camera, making them a key technology in fields such as autonomous driving and robot navigation. Traditional VO methods can be broadly categorized into two main approaches: feature-based methods and direct methods (Taketomi et al., 2017). Feature-based methods typically rely on feature point extraction and matching. Commonly used feature extraction algorithms include ORB (Mur-Artal & Tardós, 2017), SIFT, and SURF. These methods track feature points between consecutive frames and estimate camera motion using triangulation or PnP algorithms. One prominent algorithm in this category is VINS-Mono employs advanced feature tracking algorithms for feature point extraction and uses the Lucas-Kanade method for feature point tracking (Qin et al., 2018). However, these methods are sensitive to environmental factors such as lighting and texture changes. Direct methods estimate motion by directly minimizing the pixel intensity differences between images. Examples of direct methods include Direct Sparse Odometry (DSO) (Engel et al., 2018). Direct methods have less reliance on feature points and are suitable for scenes with rich brightness gradients. However, they require high computational precision and are susceptible to camera noise and occlusions. Overall, traditional visual odometry methods have made significant progress in balancing localization accuracy, robustness, and computational efficiency. However,

further improvements are needed to enhance their adaptability to dynamic scenes with the demands of several applications.

2.5 Deep Learning-Based Visual Odometry

Machine learning leverages well-trained networks to learn key points (Detone et al., 2018; Tang et al., 2019). Local point and line features can be used as inputs to neural correspondence networks to remove outliers, improving the accuracy of pose estimation (Delmerico & Scaramuzza, 2018; Rosten & Drummond, 2006). Deep learning-based visual odometry frameworks can be categorized into two groups: end-to-end frameworks and hybrid SLAM frameworks. UnDeepVO (Li et al., 2018) utilizes stereo images for training and recovers absolute scale by leveraging spatial and temporal geometric constraints. However, it still struggles with complex scenes. DPVO (Rosten & Drummond, 2006) employs patch representations based on features to encode local scene information. While end-to-end algorithms offer versatility across various application scenarios, they typically require extensive datasets and significant computational resources for training.

Hybrid SLAM frameworks maintain the modular structure of traditional SLAM systems while integrating deep learning modules to enhance overall system performance. Lift-SLAM (Bruno & Colombini, 2021) uses the Learned Invariant Feature Transform (LIFT) network to extract features in the backend of the traditional ORB-SLAM system. Similarly, DX-SLAM (Li et al., 2020) adapts the ORB features in ORB-SLAM2 to SuperPoint features (Detone et al., 2018). However, these methods still rely on traditional techniques for feature extraction and matching during tracking, which can lead to suboptimal performance in challenging environments.

3. METHOD

The algorithm framework of PLL-VO is illustrated in Figure 1. To achieve rapid, robust, and reliable extraction and matching of visual features under varying lighting conditions, we integrate a point-line feature extraction strategy with deep learning models into the system, thereby designing the visual odometry. An efficient line feature detection and tracking model is proposed. Improved EDLines algorithm employs line optical flow tracking and cross-constraint methods to increase the accuracy and efficiency of line feature detection. Moreover, each image is fed into the SuperPoint network, where the Interest Point Decoder outputs the probability of each pixel being a keypoint corresponding to that pixel in the input, and the Descriptor Decoder outputs feature descriptors. Finally, keyframes are selected based on the number of point features, time interval, and overlap angle constraints of line features. The 2D points and line features extracted from the keyframes are then input into the LightGlue network model. Associations between the two types of features are established using the matching results of the associated points.

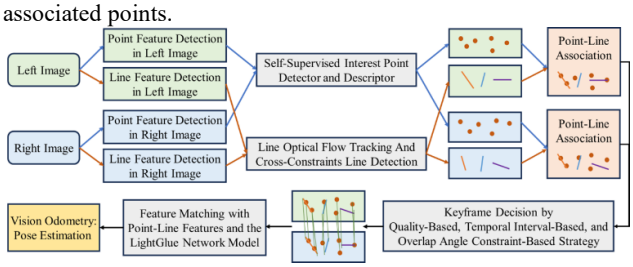


Figure 1. The algorithm framework of PLL-VO

3.1 Line Feature Extraction

Point-based visual odometry often performs poorly in scenarios with weak textures and motion blur. Many researchers have recognized the superior characteristics of line features in space and have attempted to develop line-based visual odometry systems. However, the presence of erroneous line detections has limited the performance improvement of these systems. To address these issues, we improve traditional line detection models by incorporating line optical flow tracking and cross-constraint methods, which significantly improve the accuracy of line feature extraction.

The extraction of line features is based on an improved version of the EDLines algorithm. However, it focuses primarily on detecting anchor points and may not effectively handle situations where line segments intersect or overlap. And the initial line segments generated by the EDLines algorithm may include unnecessary intermediate points, which need to be removed through fitting and filtering processes. We first optimize the abnormal line segments using line optical flow tracking and then further remove cluttered line features in complex scenes through cross-constraints.

We begin by selecting effective line features using line optical flow tracking based on the gray-level invariance criterion. Although long line segments in space are inconsistent from one viewpoint to another, the length of lines observed in consecutive frames does not change abruptly. The gray-level value of a pixel can be defined as follows:

$$\mathrm{I}(\mathrm{u}+\mathrm{du},\mathrm{v}+\mathrm{dv},\mathrm{t}+\mathrm{dt})=\mathrm{I}(\mathrm{u},\mathrm{v},\mathrm{t}) \quad (1)$$

Expanding the left-hand side using Taylor series:

$$\mathrm{I}(\mathrm{u}+\mathrm{du},\mathrm{v}+\mathrm{dv},\mathrm{t}+\mathrm{dt})\approx \mathrm{I}(\mathrm{u},\mathrm{v},\mathrm{t})+\frac{\partial \mathrm{I}}{\partial \mathrm{u}}\mathrm{du}+\frac{\partial \mathrm{I}}{\partial \mathrm{v}}\mathrm{dv}+\frac{\partial \mathrm{I}}{\partial \mathrm{t}}\mathrm{dt} \quad (2)$$

where $\frac{\partial \mathbf{I}}{\partial u}$ and $\frac{\partial \mathbf{I}}{\partial v}$, denoted as $\mathrm{I_u}$ and $\mathrm{I_v}$, represent the gray gradients of the image in the u and v directions at the given point. The temporal derivative $\frac{\partial \mathbf{I}}{\partial t}$, denoted as $\mathrm{I_t}$, indicates the rate of change of the image value with respect to time. The matrix form of these derivatives can then be expressed as:

$$[\mathrm{I_u}\quad \mathrm{I_v}]\begin{bmatrix}\frac{\mathrm{du}}{\mathrm{dt}}\\ \frac{\mathrm{dv}}{\mathrm{dt}}\end{bmatrix}=-\mathrm{I_t} \quad (3)$$

The equation (3) is valid at any pixel point within the image, implying that it inherently holds for line segments as well. However, points lying on the line segments must additionally satisfy the collinearity constraint.
We define the set of points on the line:

$$\boldsymbol{l}=\{(\mathrm{u_1},\mathrm{v_1}),(\mathrm{u_2},\mathrm{v_2}),\dots,(\mathrm{u_n},\mathrm{v_n}),\dots,(\mathrm{u_m},\mathrm{v_m})\} \quad (4)$$

where (u_1, v_1) and (u_m, v_m) represent the starting and ending points of the line, respectively. l_n represents the Euclidean distance between (u_1, v_1) and (u_n, v_n).
l' denotes the next frame position of l. The relationship between the corresponding points of the line between two consecutive frames is:

$$\begin{cases}\mathrm{u'_n}=\mathrm{u_n}+\mathrm{g_1}+\mathrm{l'_n}\cos(\alpha+\mathrm{g_3})-\mathrm{l'_n}\cos\alpha\\ \mathrm{v'_n}=\mathrm{v_n}+\mathrm{g_2}+\mathrm{l'_n}\sin(\alpha+\mathrm{g_3})-\mathrm{l'_n}\sin\alpha\end{cases} \quad (5)$$

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-1045-2025 | © Author(s) 2025. CC BY 4.0 License.

where g_1 and g_2 represent the translational changes in the position of the starting point (u_1, v_1), while g_3 denotes the rotational change around this point. Between two consecutive frames, g_3 is assumed to be a small variation, and $l_n \approx l'_n$.

Secondly, since the coordinates of every point $(\boldsymbol{u_1}, \boldsymbol{v_1}), (\boldsymbol{u_2}, \boldsymbol{v_2}), \ldots, (\boldsymbol{u_n}, \boldsymbol{v_n}), \ldots, (\boldsymbol{u_m}, \boldsymbol{v_m})$ on the line feature are known, and

$$\begin{cases} u'_n = u_n + g_1 - g_3(v_n - v_1) \\ v'_n = v_n + g_2 + g_3(u_n - u_1) \end{cases} \quad (6)$$

We aim to avoid a cluttered state where line features in a keyframe are overlapping and intersecting randomly. However, regular intersections of line features are acceptable and reasonable. Therefore, we consider both the overlap degree of line feature coordinates and the intersection angles. If a line feature overlaps with more than two other line features and the overlap angle $\Delta\theta$ is below a certain threshold, it is deemed to be in an undetected state. Consequently, we apply cross-constraints to remove such line features.

$$\Delta\theta = \theta_1 - \theta_2 \quad (7)$$

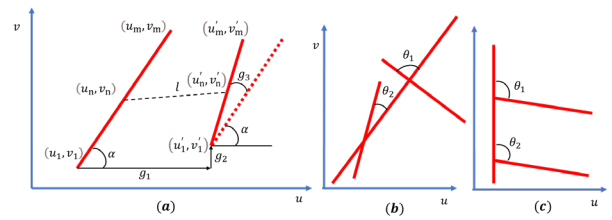
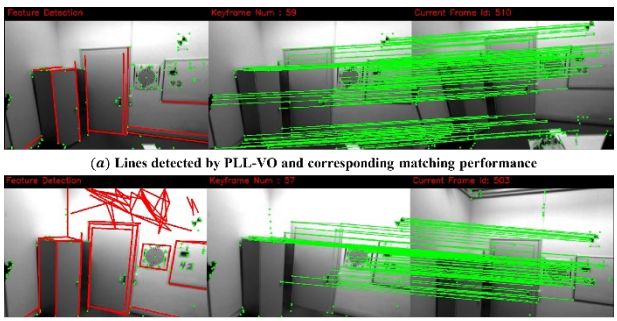


Figure 2. The measurement model of Line Feature Extraction. (a) line optical flow tracking. (b) larger overlap angles in cross constraints. (c) lower overlap angles in cross constraints.



(a) Lines detected by PLL-VO and corresponding matching performance

(b) Lines detected by LSD and corresponding matching performance

Figure 3. Lines detected by PLL-VO and LSD

3.2 Point Feature Extraction

The feature point extraction process utilizes the SuperPoint framework. This framework employs an adaptive threshold mechanism to adjust the feature point score threshold for both normal and challenging scenarios. The adaptive threshold mechanism takes into account two factors: intra-feature relationships and inter-frame feature relationships.

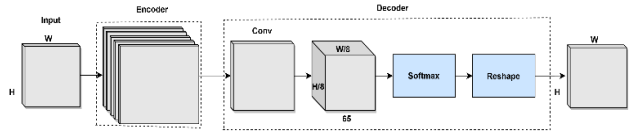


Figure 4. The work flow of SuperPoint: Self-Supervised Interest Point Detector and Descriptor

Specifically, the threshold for extracting feature points is adjusted based on the distribution and matching of feature points in the current frame. This approach enables the training of a self-supervised feature point detector and descriptor, which adapts to varying conditions by dynamically adjusting the threshold parameters.

It includes an encoder $I \in R^{W \times H}$ to a tensor $T \in R^{W_c \times H_c \times 65}$ with smaller width W_c and height H_c. The tensor is then fed to a decoder to detect feature points X. The decoder uses convolution layers to extract the response $P \in R^{W_c \times H_c \times 65}$ for feature points which also includes a "no feature point" dustbin channel. Other 64 channels represent non-overlapping 8×8 regions of the input image I. The channel-wise softmax is then used to remove the dustbin dimension and the reshape function is applied to convert P to the input dimension $W \times H$.

The loss function L for the feature point detector is a convolutional cross-entropy loss computed over the elements $x \in X$. Let $y \in Y$ be the ground-truth feature point, the loss function is computed as:

$$L(X, Y) = \frac{1}{W_c H_c} \sum_{w=1}^{W_c, H_c} l(x_{w,h}, y_{w,h}) \quad (8)$$

3.3 keyframe selection

In traditional visual odometry, keyframe selection is typically based on strategies involving tracking quality and temporal intervals. Our approach not only considers the number of point and line features and the temporal interval but also incorporates the overlap angle constraint of line features.

(1) Quality-Based Strategy: When the number of tracked points falls below a certain threshold or tracking quality degrades significantly, it indicates a significant change in the camera's field of view. In such cases, a new keyframe is inserted to maintain system stability during rapid motion or environmental changes.

(2) Temporal Interval-Based Strategy: If the number of frames exceeds a predefined maximum limit time, or if at least the minimum number of frames has passed and the mapping thread is idle, a new keyframe is chosen.

(3) Overlap Angle Constraint-Based Strategy: This strategy considers the camera's movement and changes in scene structure. We record the overlap angle values θ of line features in the last keyframe and compute the overlap angle values θ' in the current image pair. These states can be represented as T and T' respectively. Mathematically, the states can be expressed as

$$T = \{\theta_1, \theta_2, \ldots, \theta_n, \ldots, \theta_m\} T' = \{\theta'_1, \theta'_2, \ldots, \theta'_n, \ldots, \theta'_m\} \quad (9)$$

If the angle $\Delta T = T - T'$ between two-line features exceeds a certain threshold, it indicates a significant change in the surrounding scene structure, and the system will select a new keyframe.

Keyframes are selected using a strategy based on the number of point features, temporal intervals, and the overlap angle constraint of line features, to improve the accuracy of keyframe selection.

3.4 Feature Matching with Point-Line Features and the LightGlue Network Model

LightGlue predicts partial matches between local feature sets extracted from images A and B. The network consists of L identical layers that collectively process the two feature sets. Each layer comprises self-attention and cross-attention units to update the representation of each point. Subsequently, a classifier at each layer decides whether to stop the inference process, thereby avoiding unnecessary computations.

Finally, a lightweight head computes the partial matches from these representations.

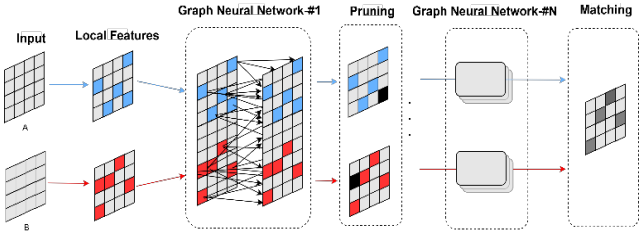


Figure 5. The work flow of LightGlue Network Model

The task of feature matching is to predict the correspondence matrix $\mathcal{M} = \{(i,j)\} \subset \mathcal{A} \times \mathcal{B}$ ， ，which corresponds to finding a soft partial assignment matrix $P \in [0,1]^{M \times N}$ 。LightGlue is a flexible image matching library that computes optimal image blending strategies based on feature point information provided by SuperPoint. This ensures smooth transitions and eliminates discontinuities in overlapping regions.

4. EXPERIMENTS AND ANALYSIS

4.1 Datasets and Experimental Settings

To validate the effectiveness of our experiments, we conducted extensive simulations using the EuRoC dataset, which includes machine warehouses and rooms. The EuRoC dataset is provided by the Autonomous Systems Lab (ASL) at ETH Zurich. It was collected using an AscTec Firefly hexacopter and includes stereo images along with precise ground truth information about motion and structure. The dataset contains 11 sequences, categorized into easy, medium, and difficult levels based on the drone's speed, lighting conditions, and scene texture. We designed extensive experiments for all three difficulty levels. To ensure a fair evaluation of the algorithm's accuracy and efficiency, all experiments were conducted using the same computational resources and dependency libraries. Our experiments are implemented with Python-3.8.5 and PyTorch-1.8.1. The environment consists of an i7-10700K CPU with 32 GB and a NVIDIA GTX-2060 graphical processing unit (GPU) with 6 GB.

4.2 Accuracy of Point and Line Feature Extraction

We compare PLLVO with state-of-the-art point-line VO and visual SLAM systems PL-SLAM (Gomez-Ojeda et al., 2019). ORB-SLAM3 and VINS-FUSION to the baseline. Figure X shows PLLVO 、 PL-SLAM 、 ORB-SLAM3 and VINS-FUSION comparison of Extracted Point and Line Features，(a) shows the point and line features extracted by the PLL-VO algorithm, (b) shows those extracted by the PL-SLAM algorithm, (c) represents VINS-Fusion, and (d) represents ORB-SLAM3. it is evident that while PLL-VO extracts fewer line features compared to PL-SLAM, the accuracy of the line features is significantly higher. Line features generated by the line segment detector in PL-SLAM often result in multiple cluttered line features on white walls, with no geometric regularity. In contrast, although PLL-VO extracts fewer line features, they are all usable and accurately represent the environment. Regarding the number of feature points, VINS-Fusion extracts more feature points than the ORB feature extraction strategy. ORB-SLAM3 can sometimes extract fewer than 30 feature points in certain keyframes, which can reduce the reliability of feature matching. Both PLL-VO and PL-SLAM provide a larger number of feature points with a more uniform distribution, make sure the overall robustness of the feature matching process.

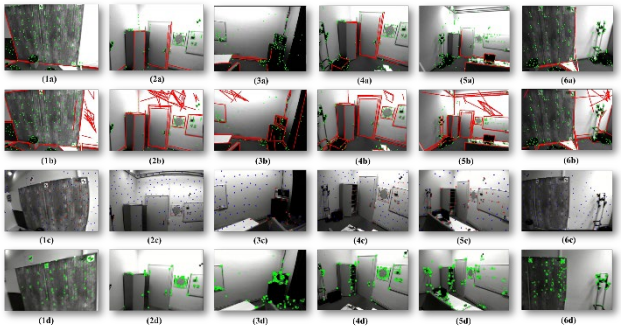


Figure 6. Comparison of feature extraction by PLL-VO, PL-SLAM, VINS-Fusion, ORB-SLAM3 in six different scenes

4.3 Time Analysis of Point and Line Feature Extraction

We validated the processing time for point and line feature extraction and compared PLL-VO with state-of-the-art feature extraction and visual SLAM systems, including PL-SLAM (Sarlin et al., 2020), ORB-SLAM3, VINS-Fusion, and SuperPoint. Since ORB-SLAM3, VINS-Fusion, and SuperPoint only extract feature points, their processing times are shorter. Despite the additional line feature extraction, PLL-VO's processing time is only marginally longer, adding less than 15 ms compared to the other methods, and it is twice as fast as PL-SLAM. The extraction speed of 66 ms per frame fully meets the real-time requirements of the system. Table 1 provides a detailed comparison of the runtime for each module of PL-SLAM and PLL-VO, demonstrating that PLL-VO significantly more efficient than that PL-SLAM.

Table 1. Time analysis of point and line feature extraction in PLL-VO, PL-SLAM, VINS-Fusion, ORB-SLAM3

Algorithm	Processing Time(ms)
ORB-SLAM3 in Feature Extraction	61
VINS-FUSION in Feature Extraction	53
Superpoint in Feature Extraction	57
PL-SLAM in Feature Extraction	147
PL-SLAM in **Line** Feature Extraction	85
PLL-VO in Feature Extraction	66
PLL-VO in **Line** Feature Extraction	18

4.4 Accuracy of Visual Odometry

We compare PLLVO with state-of-the-art visual Odometry systems SuperGlue ORB-VO and VINS to the baseline in accuracy of visual odometry. Figure 4 evaluates the pose estimation accuracy on different EuRoc datasets using flight trajectories, XYZ axis errors, roll-pitch-yaw errors, and absolute trajectory errors. Table 2 assesses the accuracy on the V1_03_Difficult EuRoc dataset using Root Mean Square Error (RMSE), Sum of Squares Due to Error (SSE), and Standard Deviation (STD).

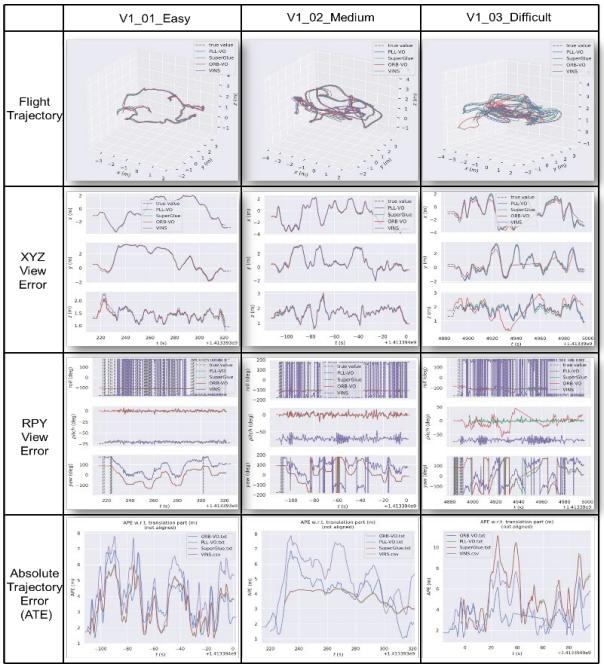


Figure 7. Flight trajectory, XYZ view error, RPY view error and ATE by PLL-VO, PL-SLAM, VINS-Fusion, ORB-SLAM3

Four methods perform well on the V1_01_Easy dataset, showing low trajectory errors with XYZ axis errors below 2 meters. On the V1_02_Medium dataset, ORB-based odometry methods exhibit significant trajectory drift when the drone performs large turning angles, and the trajectory error of VINS increases over time. While SuperGlue provides higher odometry accuracy compared to the other two algorithms, its stability is not as good as PLL-VO. On the V1_03_Difficult dataset, ORB-VO fails to match during flight, leading to pose divergence. Although VINS and SuperGlue do not fail, their trajectory errors are larger compared to PLL-VO. PLL-VO demonstrates superior performance in terms of flight trajectory, XYZ axis errors, roll-pitch-yaw errors, and absolute trajectory errors.

Table 2. RMSE,SSE and STD in PLL-VO, SuperGlue, VINS, ORB-VO

Algorithm	RMSE	SSE	STD
ORB-VO	3.12640	16893.8	**1.07432**
VINS	4.52401	21080.7	1.67039
SuperGlue	5.53992	19964.3	2.21925
PLL-VO	**3.11301**	**13964.3**	1.62547

4.5 Time Analysis of Visual Odometry

We compare PLL-VO with ORB-SLAM3, VINS-Fusion, and SuperGlue. The table below shows the execution time for feature matching of PLL-VO and other state-of-the-art (SOTA) methods on different Euroc datasets.

Table 3. Time analysis of visual odometry in PLL-VO, SuperGlue, VINS, ORB-VO

Algorithm	V1_01_Easy	V1_02_Medium	V1_03Difficult
ORB-VO	19 ms	23 ms	43 ms
VINS	21 ms	22 ms	39 ms
SuperGlue	68 ms	52 ms	174 ms
PLL-VO	37 ms	41 ms	46 ms

From Table 3, it is evident that PLL-VO maintains stable feature matching times across different datasets, primarily due to the lightweight matching algorithm. Despite using deep learning, PLL-VO has similar system execution times compared to traditional geometry-based methods. However, PLL-VO demonstrates significantly better robustness in complex lighting conditions and higher accuracy in odometry estimation compared to geometry-based visual odometry methods.

5. CONCLUSION AND FUTURE WORK

In this work, we propose a light-robust point-line fusion feature detection and matching method PLL-VO for visual odometry estimation. Initially, we optimize abnormal line segments using line optical flow tracking and further remove cluttered line features in complex scenes through cross constraints. Subsequently, we train a self-supervised feature point detector and descriptor using SuperPoint. By integrating point and line features, we employ a keyframe selection strategy based on the number of point features, temporal intervals, and the overlap angle constraint of line features. Ultimately, we use LightGlue for feature matching and pose estimation.

To evaluate the performance of PLL-VO, we conducted extensive experiments and compared it with real and synthetic datasets. The experiments demonstrate that our method achieves excellent performance in dynamic lighting conditions, validating the effectiveness of our proposed UAV system. In future work, we plan to extend PLL-VO to a SLAM system by adding loop closure detection and relocalization.

References

V on Gioi, Rafael Grompone, et al. ”LSD: A fast line segment detector with a false detection control.” IEEE transactions on pattern analysis and machine intelligence 32.4 (2008): 722-732.

Zhang, Lilian, and Reinhard Koch. ”An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency.” Journal of Visual Communication and Image Representation 24.7 (2013): 794-805.

Gomez-Ojeda, Ruben, and Javier Gonzalez-Jimenez. ”Geometric-based line segment tracking for HDR stereo sequences.” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

Akinlar, C., & Topal, C. (2011). EDLines: A real-time line segment detector with a false detection control. Pattern Recognition Letters, 32(13), 1633-1642.

Pautrat R, et al. Deeplsd: Line segment detection and refinement with deep image gradients [J]. Computer Vision and Image Understanding, 2023, 10(2): 123-135.

Zhao L, et al. Visual SLAM combining lines and structural regularities: Towards robust localization [J]. IEEE Transactions on Robotics, 2023, 8(3): 456-468.

Jeong W, Lee K M. Visual SLAM with line and corner features [J]. Robotics and Autonomous Systems, 2022, 7(4): 567-579.

Yuan Y, et al. Visual SLAM with line and corner features [J]. Journal of Field Robotics, 2021, 6(5): 678-690.

Zhou Z, et al. StructSLAM: Visual SLAM with building structure lines [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 5(6): 789-801.

Zuo X, et al. Robust visual SLAM with point and line features [J]. International Journal of Robotics Research, 2019, 4(7): 890-902.

Xu K, et al. Airvo: An illumination-robust point-line visual odometry [J]. IEEE Robotics and Automation Letters, 2023, 3(8): 903-915.

Chamorro A, et al. Event-based line SLAM in real-time [J]. IEEE Transactions on Cybernetics, 2024, 2(9): 916-928.

Lecrosnier, Louis, et al. ”Camera pose estimation based on PnL with a known vertical direction.” IEEE Robotics and Automation Letters 4.4 (2019): 3852-3859.

Li, Haoang, et al. ”Line-based absolute and relative camera pose estimation in structured environments.” 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.

Fabbri, Ricardo, et al. ”TRPLP-Trifocal Relative Pose From Lines at Points.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, “Pl-slam: A stereo slam system through the combination of points and line segments,” IEEE Transactions on Robotics, vol. 35, no. 3, pp. 734–746, 2019.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. (2016). Lift: Learned invariant feature transform. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14 (pp. 467-483). Springer International Publishing.

Savinov, N., Seki, A., Ladicky, L., Sattler, T., & Pollefeys, M. (2017). Quad-networks: unsupervised learning to rank for interest point detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1822-1830).

Verdie, Y., Yi, K., Fua, P., & Lepetit, V. (2015). Tilde: A temporally invariant learned detector. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5279-5288).

T. Taketomi, H. Uchiyama and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016", IPSJ Trans. Comput. Vis. Appl, vol. 9, no. 1, 2017.
R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular Stereo and RGB-D Cameras", IEEE Trans. Robot, vol. 33, no. 5, pp. 1255-1262, 2017.

T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004–1020, 2018.

J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry", IEEE Trans. Pattern Anal. Mach. Intell, vol. 40, no. 3, pp. 611-625, 2018.

D. Detone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description", IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work, vol. 2018-June, pp. 337-349, 2018.

J. Tang, L. Ericson, J. Folkesson and P. Jensfelt, "GCNv2: Efficient Correspondence Prediction for Real-Time SLAM", IEEE Robot. Autom. Lett, pp. 1-1, Feb. 2019.

P. E. Sarlin, D. Detone, T. Malisiewicz and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, pp. 4937-4946, 2020.

J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots", 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2502-2509, May 2018.

E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3951, pp. 430-443, 2006.

R. Li, S. Wang, Z. Long, and D. Gu, “Undeepvo: Monocular visual odometry through unsupervised deep learning,” in 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 7286–7291.

Z. Teed, L. Lipson, and J. Deng, “Deep patch visual odometry,” Advances in Neural Information Processing Systems, vol. 36, 2024.

H. M. S. Bruno and E. L. Colombini, “Lift-slam: A deep-learning feature-based monocular visual slam method,” Neurocomputing, vol. 455, pp. 97–110, 2021.

D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, “Dxslam: A robust and efficient visual slam system with deep features,” in 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE, 2020, pp. 4958–4965.

Quach, C. H., Phung, M. D., Le, H. V., & Perry, S. (2021, December). SupSLAM: A robust visual inertial SLAM system using SuperPoint for unmanned aerial vehicles. In 2021 8th NAFOSTED Conference on Information and Computer Science (NICS) (pp. 507-512). IEEE.

Sarlin, P. E., DeTone, D., Malisiewicz, T., & Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4938-4947).

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 224-236).

Lindenberger, P., Sarlin, P. E., & Pollefeys, M. (2023). Lightglue: Local feature matching at light speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 17627-17638).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60, 91-110.

Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision (pp. 2564-2571). Ieee.