applied sciences

MDPI

Article

Balancing Efficiency and Accuracy: Enhanced Visual Simultaneous Localization and Mapping Incorporating Principal Direction Features

Yuelin Yuan [1,2], Fei Li [1,2], Xiaohui Liu [3,4,*] and Jialiang Chen [2,5]

[1] School of Space and Environment, Beihang University, Beijing 102206, China; y1966026059@163.com (Y.Y.); lifei0406@whut.edu.cn (F.L.)
[2] College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China; chenjl806@mail.dlut.edu.cn
[3] College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China
[4] Engineering Research Center for Position, Navigation and Time, National University of Defense Technology, Changsha 410073, China
[5] Department of Transportation and Logistics, Dalian University of Technology, Dalian 116024, China
* Correspondence: liuxh@nudt.edu.cn

Abstract: In visual Simultaneous Localization and Mapping (SLAM), operational efficiency and localization accuracy are equally crucial evaluation metrics. We propose an enhanced visual SLAM method to ensure stable localization accuracy while improving system efficiency. It can maintain localization accuracy even after reducing the number of feature pyramid levels by 50%. Firstly, we innovatively incorporate the principal direction error, which represents the global geometric features of feature points, into the error function for pose estimation, utilizing Pareto optimal solutions to improve the localization accuracy. Secondly, for loop-closure detection, we construct a feature matrix by integrating the grayscale and gradient direction of an image. This matrix is then dimensionally reduced through aggregation, and a multi-layer detection approach is employed to ensure both efficiency and accuracy. Finally, we optimize the feature extraction levels and integrate our method into the visual system to speed up the extraction process and mitigate the impact of the reduced levels. We comprehensively evaluate the proposed method on local and public datasets. Experiments show that the SLAM method maintained high localization accuracy after reducing the tracking time by 24% compared with ORB SLAM3. Additionally, the proposed loop-closure-detection method demonstrated superior computational efficiency and detection accuracy compared to the existing methods.

Keywords: visual SLAM; principal direction projection; loop-closure detection; features pyramid

check for updates

Citation: Yuan, Y.; Li, F.; Liu, X.; Chen, J. Balancing Efficiency and Accuracy: Enhanced Visual Simultaneous Localization and Mapping Incorporating Principal Direction Features. *Appl. Sci.* **2024**, *14*, 9124. https://doi.org/10.3390/app14199124

Academic Editor: Andrea Prati

Received: 19 August 2024
Revised: 12 September 2024
Accepted: 4 October 2024
Published: 9 October 2024

1. Introduction

Simultaneous Localization and Mapping (SLAM) enables unmanned devices, like autonomous vehicles and drones, to perceive their surrounding environment. In practical applications, the key performance indicators for SLAM include the mapping consistency, localization accuracy, and operational efficiency, which are also the focus of ongoing research.

Depending on the primary sensor, SLAM can be broadly divided into Visual SLAM (V-SLAM) [1] and LiDAR SLAM [2]. Visual SLAM is typically favored in environments rich in visual features, while LiDAR SLAM excels in structured environments and performs well even under low-light conditions. Among the cutting-edge visual SLAM methods, ORB-SLAM3 [3] and DVI-SLAM [4] stand out. In LiDAR SLAM, widely used methods include graph optimization-based LOCUS 2.0 [5] and filter-based Point-LIO [6]. As mentioned, SLAM is widely applied across various devices. Autonomous vehicles rely on SLAM for localization and mapping in dynamic environments to ensure safe navigation, particularly in low-speed or enclosed areas, such as mining sites [7], where SLAM plays a crucial role. Single drones or multi-drone collaborations can utilize SLAM to autonomously explore and map unknown terrains [8], which is vital for search and rescue missions in environments where satellite navigation is unavailable. In robotics, such as indoor service robots [9] and construction robots [10], SLAM enables navigation in both indoor and outdoor environments. With advanced decision-making and control technologies [11], SLAM allows these robots to perform complex and diverse tasks. Visual SLAM, compared to LiDAR SLAM, is more cost-effective and has higher practical value, making it a consistent area of research focus.

Visual SLAM has continuously developed, with major SLAM systems differing primarily in pose estimation and image processing approaches [12]. Pose estimation methods include filter-based and keyframe-based approaches. Regarding image processing, there are two main methods: direct and feature-based. Mono-SLAM [13] is one of the earliest SLAM systems, employing Shi–Tomasi points and an Extended Kalman Filter (EKF) for pose estimation. Subsequent method improvements have largely retained the fundamental principles of Mono-SLAM. Filter-based methods use only current frame information, which, while computationally efficient, results in limited accuracy. Keyframe-based Bundle Adjustment (BA) is widely used in visual systems, with PTAM [14] and SVO [15] employing BA for estimating the pose between successive frames, offering higher accuracy compared to filter-based methods. Direct methods do not require feature extraction; they utilize photometric information. In Reference [16], Zubizarreta et al. introduced the concept of reusing existing map information in direct V-SLAM, achieving improved localization accuracy. Widely used features in feature-based SLAM include oriented FAST and rotated BRIEF (ORB) [17,18], scale-invariant feature transform (SIFT) [19], speeded-up robust features (SURF) [20], and Self-supervised interest point (SURPOINT) [21]. ORB SLAM3 [3] is an advanced and comprehensive visual SLAM system based on ORB features. It supports monocular, stereo, and RGB-D cameras and can be tightly coupled with an Inertial Measurement Unit (IMU) to enhance environmental adaptability. Additionally, ORB SLAM3 offers map reuse, loop closure, and re-localization capabilities to ensure mapping consistency. Compared to ORB SLAM2 [22], ORB SLAM3 features multiple improvements, such as utilizing an atlas to retain all maps, enabling map merging after loop closure, and enhancing loop-closure detection by verifying the local consistency of co-visible frames, thus improving recall. In Reference [3], the authors compared mainstream V-SLAM and Visual Inertial Odometry (VIO) methods, such as DSM [16] and VINS-Fusion [23], demonstrating that ORB SLAM3 is one of the most accurate and robust methods.

The emergence of event cameras has introduced new approaches to the front-end of SLAM systems. Chen et al. [24], unlike pure stereo event-based visual localization [25], designed a tightly coupled system based on stereo event cameras and IMU, enabling stable operation in challenging environments. Event-based visual SLAM demonstrates impressive performance in high-speed motion scenarios but also induces issues, such as sparse data. Additionally, neural networks are playing an increasingly important role in V-SLAM. In Reference [26], the authors proposed a semantic SLAM method for dynamic scenes, where they designed a static semantic keyframe selection strategy based on segmentation results to mitigate the impact of dynamic objects. Wang et al. [27] integrated visual, inertial, and semantic information for final localization. Peng et al. [4] proposed a deep SLAM network that directly integrates visual and IMU data. Incorporating neural networks provides more robust feature-extraction capabilities than traditional SLAM methods, making them more adaptable to complex environments, such as low-texture scenes. However, neural-network-based methods rely heavily on training data, have high computational requirements, and involve a more complicated system. The methods above are primarily based on point features. Alamanos et al. [28] innovatively combined point and line features in ORB-LINE-SLAM, improving the localization accuracy in challenging environments. In Reference [29], the authors used vertical line features and point features to achieve promising results in underground parking lots. Combining line features enhances geometric constraints but also introduces challenges, such as the instability of line features and increased complexity in matching.

During the operation of visual systems, the proportion of the target's appearance in the image is unknown. A more effective method for extracting target features is to generate an image pyramid composed of images at different scales, scanning the target layer-by-layer [30]. The image pyramid is extensively used in feature extraction to adaptively extract features from objects at different scales within the image [31]. Higher-level features exhibit lower repeatability, offering limited benefits in localization accuracy, but they are still advantageous for feature matching and orientation estimation [32]. In practical applications, the number of pyramid levels and the scale factor are related to feature selection and the image size. The most time-consuming part of the front-end process in ORB SLAM3 is the multi-level feature extraction based on the image pyramid. Taranco et al. [33] designed a high-performance hardware accelerator for ORB feature extraction. In engineering applications, ORB feature extraction is often processed in parallel using multithreading to improve the runtime efficiency.

These methods mentioned above for accelerating feature extraction impose high hardware requirements, hindering their applicability and scalability. While ORB features exhibit desirable performance and the effectiveness of the image pyramid has been validated, due to performance constraints, low-performance intelligent devices, like drones, primarily use FAST features at the front end [34]. For instance, Geneva et al. [35] developed a computationally efficient visual–inertial odometry system by integrating FAST with optical flow techniques. This approach is particularly suitable for visual localization in drones due to its low computational requirements. Neural-network-based methods require the design of complex systems and the collection of large amounts of data, making their implementation challenging in most applications. We hope to propose an efficient method that enhances the runtime efficiency of visual SLAM systems while maintaining stable localization accuracy.

Loop-closure detection can provide spatial constraints between previous and current frames, effectively improving mapping consistency. Recent research primarily focuses on detecting loop closure in visual SLAM by constructing descriptors (hand-engineered features). These descriptors can be classified into local and global feature descriptors [36]. Local feature descriptors, such as SIFT, effectively represent image details and are robust to changes in viewpoint and scale. Global descriptors, such as Histograms of Oriented Gradients (HOGs) [37], primarily capture global image information and are highly adaptable to lighting conditions. The Bag-of-Words (BOW) model, which represents feature descriptors using visual words, is one of the effective methods for image feature representation [38]. Loop-closure detection based on local features generally combines BOW. After calculating the corresponding word for the local features, the word frequency is counted and weighted, followed by similarity computation. BOW-based loop-closure detection requires offline vocabulary training, making it inconvenient, and its performance is heavily influenced by the training data. If the training and application data differ significantly, detection accuracy decreases. Furthermore, dictionaries typically have large sizes to enhance adaptability to different scenes, which poses efficiency challenges. Unlike BOW, Wang et al. [39] proposed a novel approach for loop-closure detection based on salient regions, SRLCD, matching directly in the frequency domain. Salient regions can also be considered a type of local feature. Among global descriptors, the Gradient Orientation Histogram [37] and Grayscale Histogram [40] are widely used. Dalal et al. [37] employed the Gradient Histogram as an image feature, combined with SVM, achieving significant success in human detection, demonstrating the potential of global features in image detection. Li et al. [36] also used the Gradient Orientation Histogram as a descriptor for image representation and introduced image block division and clustering to enhance the local description. The proposed method also incorporated an online incremental vocabulary method based on BOW, offering better environmental adaptability. However, until the vocabulary accumulates to a certain extent, the method cannot detect accurately, and the complexity of computing feature descriptors introduces additional time consumption and performance uncertainty. Tao et al. [41] directly employed a combination of the Grayscale Histogram and key region covariance for loop-closure detection, achieving higher computational efficiency.

Moreover, neural networks can directly extract image features, replacing hand-engineered features in loop-closure detection. Neural-network-based loop-closure detection methods have garnered significant attention. Chen et al. [42] used Convolutional Neural Networks (CNNs) to extract image features, combined with spatial and sequential detection for loop-closure detection, achieving good adaptability to viewpoint changes and high detection accuracy. Gao et al. [43] selected multiple image blocks in the image using key points, such as SIFT, and employed an unsupervised deep learning network, a stacked denoising auto-encoder (SDA), to extract feature representations for similarity computation to achieve loop-closure detection. Samadzadeh et al. [44] incorporated deep neural networks (DNNs) for feature extraction and matching in the loop-closure-detection stage, which improved the accuracy of loop-closure detection. In Reference [45], the authors considered the low-latency characteristics of Spiking Neural Networks (SNNs) and proposed a lightweight and fast place-recognition method called VPRTempo. Neural networks can also extract high-level image features, such as semantic features, which can be applied in loop-closure detection. Cheng et al. [46] replaced traditional image features with semantic vectors and calculated vector similarity to filter candidate keyframes. Li et al. [47] computed the similarities of GIST features, semantic features, and appearance features separately and then performed loop-closure detection using a weighted fusion approach. High-level semantic features and hand-engineered features provide complementary information. Arshad et al. [48] combined feature similarity with semantic similarity to make final loop-closure judgments, enhancing the adaptability to dynamic object interference and viewpoint changes. In Reference [49], the authors designed a Semantic-Visual Word to improve the robustness of the loop-closure-detection algorithm. Combining semantic and BOW models or removing dynamic objects based on semantic information [50] can lead to higher accuracy.

For loop-closure detection based on local features, extracting thousands of feature descriptors is expected to ensure detection performance, which is computationally expensive. Moreover, during use, BOW is often combined, requiring pre-training of the vocabulary, which poses challenges for practical applications. The performance of neural-network-based loop-closure detection methods is strongly related to the quality of training data, and there is no significant advantage in computational efficiency. Furthermore, integrating high-level features, such as semantics, increases the complexity of SLAM systems. Therefore, we focus on proposing a precise and efficient loop-closure-detection method that integrates global features to enhance the applicability and operational efficiency of visual SLAM systems.

Based on the current issues and research purposes, we propose an innovative method to enhance the applicability and computational efficiency of SLAM systems. The multi-level extraction of ORB features is time-consuming, affecting the real-time performance of the system. The proposed method optimizes the number of feature extraction levels while maintaining the approaching image extraction depth. In order to address the issue of the decreased positioning accuracy caused by adjusting the number of levels, we have added the global structural information of the feature points to maintain the positioning accuracy. Specifically, the principal direction error of feature points is combined with reprojection error for pose estimation. This approach ensures fast computation and maintains the localization accuracy of the SLAM system. The loop-closure-detection method based on the BOW model requires prior training and is highly dependent on training data, which could not be conducive to practical applications. Therefore, we directly utilize the grayscale and gradient information of images to construct two-dimensional descriptors for selecting loop-closure keyframes. Directly using two-dimensional descriptors for the similarity calculation incurs high computational and storage costs. We first aggregate the two-dimensional feature descriptors into two one-dimensional feature vectors for subsequent similarity calculations to improve the operational efficiency. The aggregation process is computationally fast and reduces the data volume by sacrificing some detail resolution.

For images that are either excessively dark or bright, we apply a weighted function to the feature vectors, aiming to ensure the accuracy of the similarity calculations. The final loop-closure keyframes are obtained through a multi-layer detection consisting of a coarse and a fine selection, achieving high accuracy and fast runtime.

The main contributions of this work are summarized as follows:

1. We integrate the principal direction error of feature points with reprojection error for pose estimation, using Pareto optimal solutions to ensure result quality. This approach maintains localization accuracy while reducing the number of ORB feature extraction levels, thereby improving system efficiency.
2. We combine grayscale and gradient information to construct feature descriptors for loop-closure detection, utilizing dimensionality reduction and a multi-layer detection approach to ensure high speed and accuracy. It does not require the pre-training of vocabularies or loading pre-trained vocabularies.
3. Experiments on public and local datasets demonstrate that the proposed method outperforms the comparison methods in performance and real-time operation.

The rest of the paper is organized as follows: Section 2 presents our proposed method, including the new method framework, improved pose estimation method, descriptor construction, and multi-layer search strategy. Section 3 evaluates the proposed method based on public and local datasets. Section 4 provides further analysis and discussion of the experimental results, highlighting key conclusions. Finally, Section 5 concludes the paper with a summary of the main contributions.

2. Materials and Methods

2.1. System Overview

As one of the most advanced visual SLAM methods available, ORB SLAM3 delivers satisfactory performance across various applications. Our method adopts ORB SLAM3 as the core framework, with several essential modifications. The overall flowchart of the method is shown in Figure 1. The modifications focus on two main aspects: (1) incorporating the principal direction information of feature points, which reflects their overall geometric distribution, into the error calculation function during pose estimation, thereby maintaining high localization accuracy even with a reduced number of ORB feature-extraction levels. As shown in the yellow dashed box labeled a; and (2) replacing the BOW-based loop-closure-detection method in ORB SLAM3 with our proposed loop-closure-detection method based on aggregation descriptors, which eliminates the need for a training process, improves application efficiency, and ensures both detection speed and accuracy, as shown in the yellow dashed boxes labeled b and c.

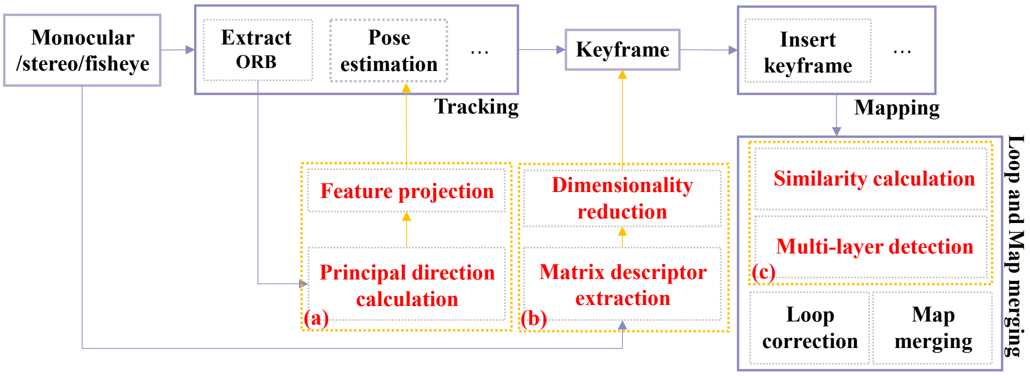


Figure 1. The visual SLAM framework. The main contributions of our work are highlighted within the yellow dashed box. (**a**) Pose estimation incorporating principal direction information; (**b**) descriptor extraction; (**c**) similarity calculation and loop-closure detection.

Specifically, we provided a detailed description of the modifications to illustrate our work. The process in Figures 2 and 3 serve as supplementary explanations of the detailed components from the visual framework shown in Figure 1. In Figure 2, the principal direction of the feature points in the current frame is calculated, and the feature points are projected onto this orientation. The points in the world coordinate system are similarly transformed into the pixel coordinate system and projected onto the principal direction. During the camera pose estimation, the projection error along the principal direction is minimized with the conventional reprojection error, resulting in improved estimation accuracy. ORB features exhibit a certain degree of robustness to image noise [3]. By ignoring variations in secondary directions, the primary direction information reduces the impact of local noise, thereby enhancing noise resistance. Consequently, this method is designed with scene generalization capabilities in mind, considering the operational efficiency of the system.

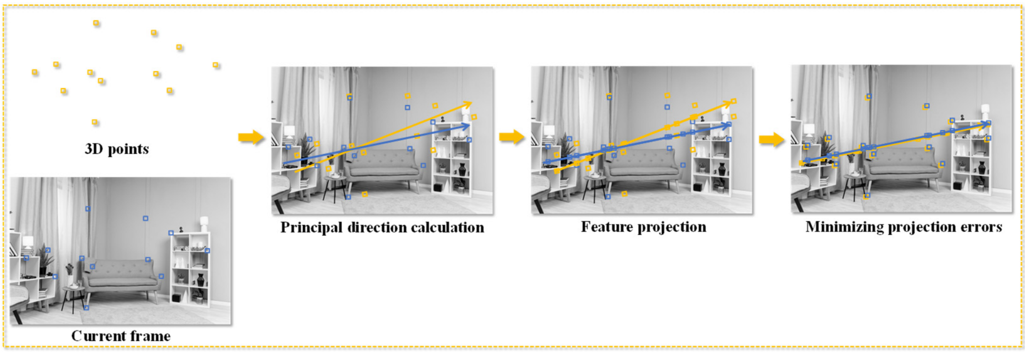


Figure 2. Pose estimation incorporating principal direction information. The small rectangles represent feature points, and the straight lines with arrows represent the principal directions of the feature points.

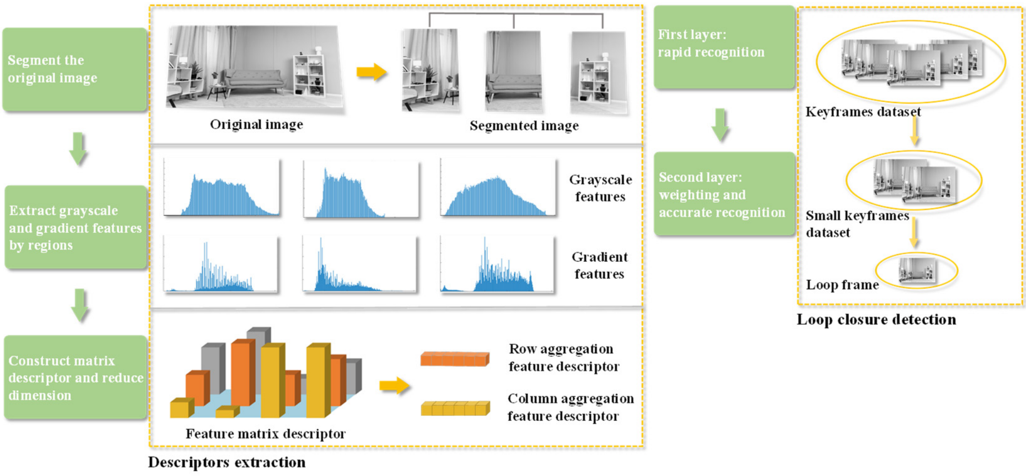

Segment the original image
Extract grayscale and gradient features by regions
Construct matrix descriptor and reduce dimension
Original image
Segmented image
Grayscale features
Gradient features
Row aggregation feature descriptor
Column aggregation feature descriptor
Feature matrix descriptor
Descriptors extraction
First layer: rapid recognition
Second layer: weighting and accurate recognition
Keyframes dataset
Small keyframes dataset
Loop frame
Loop closure detection

Figure 3. Loop-closure detection based on aggregation descriptors. The left side shows the aggregation feature descriptor extraction process, while the right side shows the loop-closure-detection process.

Figure 3 shows the workflow of our proposed loop-closure-detection method. Firstly, the image is segmented, and grayscale and gradient features are extracted from each region to construct a feature matrix descriptor. Then, the matrix is aggregated and dimensionally reduced along both columns and rows (by summing) to obtain two types of aggregation feature descriptors, thereby enhancing the efficiency of subsequent filtering and storage. Loop-closure detection is then performed in two stages. The first stage aims to identify k candidate frames quickly, using only one aggregation feature descriptor and calculating the Manhattan distance, which is computationally simple. The second stage focuses on accurately identifying the final loop-closure frame by combining two aggregation feature descriptors and employing a more complex correlation coefficient calculation to measure similarity. This method also considers sensor generalization in its design. Based on global image information, the method is insensitive to input image types, such as fisheye and panoramic images, and does not require additional processing steps to accommodate these variations.

2.2. Pose Estimation

In visual localization, camera pose estimation is performed by minimizing the reprojection error. The 3D points are projected onto the current frame based on the pose and camera model of the camera, resulting in image points. The camera pose is iteratively adjusted to minimize the error between the observed and projected image points. It is assumed that these points are perfectly matched. The error $\mathbf{e}_{ij}$ for the 3D point $\mathbf{p}_j$ in the frame i is defined as follows [3]:

$$\mathbf{e}_{ij} = \mathbf{u}_{ij} - \Pi\Big(\mathbf{T}_{cw} \bigoplus \mathbf{p}_j\Big) \tag{1}$$

where $\mathbf{u}_{ij}$ is the coordinate of the observed point in the pixel coordinate system and $\Pi : \mathbb{R}^3 \to \mathbb{R}^n$ represents the projection function of the camera. $\mathbf{T}_{cw}$ stands for the transformation from the world to camera coordinate. $\bigoplus$ is the transformation operation of $SE(3)$ group over $\mathbb{R}^3$ elements.

We adjusted the number of levels in the multi-level feature-extraction process to enhance the processing speed of the system. This adjustment introduces feature-extraction and matching errors, which can ultimately lead to deviations in pose estimation. To maintain the accuracy of pose estimation, it is essential to incorporate additional information constraints and reduce the influence of outliers.

The global distribution characteristics of the position points can be leveraged. The principal direction, representing the direction with the most significant data variation and maximum variance, reflects the distribution characteristics of the data. Incorporating the projection error along the principal direction into the error calculation can improve pose estimation accuracy. Specifically, we first centralize the matched image feature point $\mathbf{x}_{ij}$ and compute the covariance matrix $\mathbf{C}$. By decomposing the covariance matrix, we obtain the principal direction. The position points $\mathbf{p}_j$ in the world coordinate system are then transformed into the pixel coordinate system and projected into the principal direction, where we compare the projection error.

$$\mathbf{x}_{ij}^c = \mathbf{x}_{ij} - \mathbf{u} \tag{2}$$

$$\mathbf{C} = \frac{\sum_j^n \mathbf{x}_{ij}^c \mathbf{x}_{ij}^{c\,\mathrm{T}}}{n-1} \tag{3}$$

$$\mathbf{C} = \mathbf{V}^{\mathrm{T}} \mathbf{\Lambda} \mathbf{V} \tag{4}$$

$$\mathbf{x}_{ij}' = \mathbf{v}_m^{\mathrm{T}} * \Pi\Big(\mathbf{T}_{cw} \bigoplus \mathbf{p}_i\Big) * \mathbf{v}_m + \mathbf{u} \tag{5}$$

where $\mathbf{u}$ is the center of image feature points, n is the number of feature points, $\mathbf{V}^{\mathrm{T}}$ is the matrix of eigenvectors, $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, and $\mathbf{v}_m$ is the principal direction, the eigenvector corresponding to the largest eigenvalue. After obtaining the projected point $\mathbf{x}_{ij}'$, the principal direction projection error $\mathbf{e}_{ij}'$ is

$$\mathbf{e}_{ij}' = \mathbf{u}_{ij}' - \mathbf{x}_{ij}' \tag{6}$$

Given that multiple error functions are optimized simultaneously, an error upper bound constraint ε is introduced to ensure the quality of the solution, constraining the projection error along the principal direction.

$$\mathbf{e}_{ij}'^{\mathrm{T}} * \mathbf{e}_{ij}' < \varepsilon \tag{7}$$

Thus, the improved error formula is

$$\begin{cases} argmin\left(\sum_i^n \left\|\mathbf{e}_{ij}\right\|_{\Sigma} + \sum_i^n \left\|\mathbf{e}_{ij}'\right\|_{\acute{\Sigma}}\right) \\ \mathbf{e}_{ij}'^{\mathrm{T}} * \mathbf{e}_{ij}' < \varepsilon \end{cases} \tag{8}$$

In Equation (8), the reprojection error $\sum_i^n \left\|\mathbf{e}_{ij}\right\|_{\Sigma}$ is consistent with the formulation in Reference [3], where n represents the number of feature points, and Σ denotes the covariance matrix. The difference in our approach is that we additionally incorporate the principal direction error $\sum_i^n \left\|\mathbf{e}_{ij}'\right\|_{\acute{\Sigma}}$ into the error equation. We also apply constraints to the error boundary to mitigate the influence of outliers. The final pose estimation is obtained by minimizing the reprojection error and the principal direction error.

After incorporating the principal direction projection error, although both error components share the same unit of pixel distance, it is inevitable that optimizing one objective function may deteriorate the other. In practice, it is necessary to identify the Pareto optimal solution. The specific description is as follows. Assume

$$f_1(\xi) = \sum_i^n \left\|\mathbf{e}_{ij}\right\|_{\Sigma} \tag{9}$$

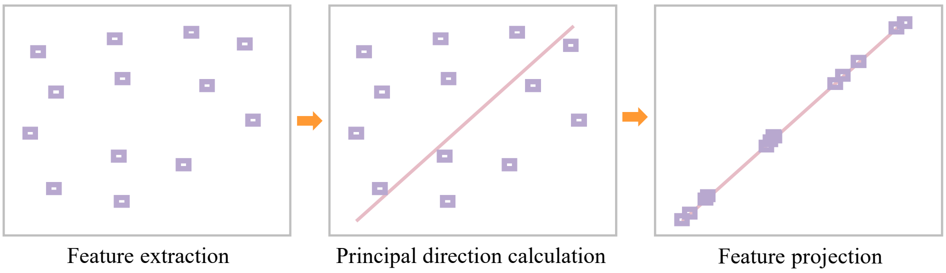
$$f_2(\xi) = \sum_i^n \left\|\mathbf{e}_{ij}'\right\|_{\acute{\Sigma}} \tag{10}$$

If there exists a solution $\boldsymbol{\xi_1}$ that satisfies the following formula:

$$\begin{cases} f_1(\boldsymbol{\xi_1}) \le f_1(\boldsymbol{\xi_2}) \\ f_2(\boldsymbol{\xi_1}) \le f_2(\boldsymbol{\xi_2}) \\ (f_1(\boldsymbol{\xi_1}) < f_1(\boldsymbol{\xi_2})) \vee (f_2(\boldsymbol{\xi_1}) < f_2(\boldsymbol{\xi_2})) \end{cases} \tag{11}$$

where $\vee$ denotes a logical OR. Then, $\boldsymbol{\xi_1}$ is considered to dominate the solution $\boldsymbol{\xi_2}$. Solutions not dominated by others are referred to as solutions on the Pareto Frontier. The pose-estimation process is executed multiple times, and after each execution, we obtain an estimation solution and remove suspicious outlier feature points. Indeed, some of these points might be accurate but classified as outliers due to estimation errors. This is one of the reasons for employing the Pareto approach. All solutions on the front form a solution set, from which the Pareto optimal solution is selected. We choose the solution on the Frontier that minimizes $f_1(\xi)$ as the Pareto optimal solution.

The principal direction projection is shown in Figure 4. We first calculate the principal direction of the feature points, then project the feature points onto the principal direction, and finally use them in pose estimation along with the reprojection error.



Feature extraction

Principal direction calculation

Feature projection

Figure 4. The illustration of principal direction feature projection. The purple rectangles represent feature points, while the pink lines indicate the principal directions of the feature points.

2.3. Descriptor Extraction

Unlike local feature descriptors, global features have a simpler computation process with lower time complexity. Additionally, since they do not require storing many feature points, the required storage space is minimal. Grayscale and gradient features are typical examples of global features. Grayscale features effectively represent the overall brightness of an image and are insensitive to global changes, such as illumination. Gradient features, on the other hand, capture the texture information of an image, making them sensitive to shape and texture while being robust to changes in viewpoint. This allows them to complement grayscale features in describing local characteristics. We construct a feature matrix descriptor by combining grayscale and gradient information. This method combines grayscale and gradient information to construct a feature matrix descriptor. We reduce feature dimensionality by aggregating the feature matrix to obtain two aggregation feature descriptors in the column and row directions.

During the feature-extraction process, the image is vertically segmented into multiple small regions R_i to ensure that the extracted features capture more detailed information. Grayscale features are firstly extracted from each region, with the grayscale range redivided into intervals—we used 64 intervals. Assuming the grayscale value of the image is $I(x, y)$, the grayscale feature can be represented as follows:

$$\mathbf{h}_g(g) = \sum\nolimits_{(x,y)} \delta(I(x, y) - g) \tag{12}$$

where δ is the Dirac function and g represents the grayscale index. The subscript g represents that the $\mathbf{h}$ vector is a grayscale feature vector.

Next, convolution kernels are applied to compute the gradient information of the image region R_i in both the X and Y directions. The gradient, including magnitude and direction, is then calculated based on the results from these two directions. The gradient direction intervals are redefined, and the gradient magnitude is used for weighting. Assuming the gradient direction of the image is $D(x, y)$ and the gradient magnitude is $G(x, y)$, the gradient feature can be expressed as

$$\mathbf{h}_s(\theta) = \sum\nolimits_{(x,y)} \delta(D(x, y) - \theta)G(x, y) \tag{13}$$

where θ represents the directional index and the subscript s represents that the $\mathbf{h}$ vector is a gradient feature vector.

The grayscale and gradient features are combined to obtain a 2D feature matrix $\mathbf{H}_{gs}(\theta, g)$.

$$\mathbf{H}_{gs}(\theta, g) = \sum\nolimits_{(x,y)} \delta(D(x, y) - \theta)\delta(I(x, y) - g)G(x, y) \tag{14}$$

Directly using the feature descriptor matrix for similarity computation can result in a high computational cost, especially as the number of features accumulates, making the cost prohibitive. We employ aggregation-based dimensionality reduction to reduce the feature matrix into feature vectors along the column and row directions to address this. Aggregative dimensionality reduction methods are essentially a form of information quantization. They offer fast computation speeds and achieve data volume reduction by sacrificing some detail resolution. After dimensionality reduction, the aggregation feature descriptor is expressed as

$$\mathbf{v}_r = \sum\nolimits_g \mathbf{H}_{gs}(\theta, g) \tag{15}$$

$$\mathbf{v}_c = \sum\nolimits_\theta \mathbf{H}_{gs}(\theta, g) \tag{16}$$

Figure 5 displays the process of generating the aggregation feature descriptor, which primarily includes image segmentation, feature matrix construction, and aggregation-based dimensionality reduction. The length of the feature vectors is related to the quantization indexes of the grayscale and gradient direction, with longer vectors providing higher-detail resolution. It is important to note that the two vectors have different units.

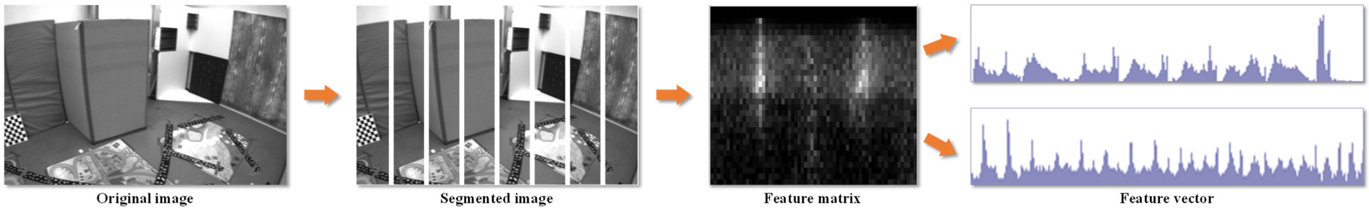


Figure 5. Aggregation feature descriptor generation process. Each image generates two feature vectors, aggregated from the feature matrix along the row and column directions.

2.4. Loop-Closure Detection

Loop-closure detection aims to determine whether the robot has returned to a previously visited localization. Without absolute localization, loop closure cannot be simply judged based on position, as odometry errors accumulate over time. In visual SLAM, loop-closure detection primarily involves three key components: (1) similarity calculation, where the similarity between the current and previous keyframes is computed; (2) candidate frame selection, where keyframes that are likely to have loop closures are selected based on similarity results; and (3) loop verification, which is critical as incorrect loop closures can lead to severe localization errors, typically involving inliers and co-visible frames. We focus on the first two aspects.

We employ a multi-layer detection approach to ensure both detection speed and accuracy.

2.4.1. Similarity Calculation

The primary requirement for the first detection layer is high computational efficiency with a reasonable level of accuracy to filter out a coarse set of candidate frames. The feature distance will be calculated using the Manhattan distance, characterized by its low computational complexity, fast speed, and robustness against outliers, making it suitable for this purpose. The calculation formula is as follows:

$$m = \sum_{i=1}^{n} |\mathbf{v}_1[i] - \mathbf{v}_2[i]| \tag{17}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ represent the aggregation feature descriptors of the current frame and the candidate frame, respectively, and n is the size of the descriptor.

The second layer of detection involves more refined processing. We first assess the energy distribution of the current grayscale features to accommodate overly dark or bright image data. We uniformly divide the vector into three segments and calculate the energy proportion of the first and third segments. When this energy proportion p exceeds 95%, we have sufficient reason to consider the current image excessively dark or bright. We apply a weighting function to the features for such images, amplifying local differences. Inspired by Contrast Limited Adaptive Histogram Equalization (CLAHE), we perform energy proportion and weighting function calculations in each well-segmented small area to better preserve local information. We then use the correlation coefficient to calculate similarity. Although this computation is relatively complex, it offers superior discriminative performance.

$$p = \frac{\sum_{i}^{n/3} \mathbf{v}[i]}{\sum_{i}^{n} \mathbf{v}[i]} \tag{18}$$

The weighting function K is defined as follows and is represented in vector form as $\mathbf{w}$:

$$K(i) = \frac{k}{\sqrt{1 + (\frac{i}{i_0})^2}} = \mathbf{w}[i] \tag{19}$$

The correlation coefficient for calculating similarity is as follows:

$$r = \frac{\sum_{i=1}^{n} (\mathbf{v}_1[i] - \overline{\mathbf{v}_1})(\mathbf{v}_2[i] - \overline{\mathbf{v}_2})}{\sqrt{\sum_{i=1}^{n}(\mathbf{v}_1[i] - \overline{\mathbf{v}_1})^2 \sum_{i=1}^{n}(\mathbf{v}_2[i] - \overline{\mathbf{v}_2})^2}} \quad (20)$$

where, $\overline{\mathbf{v}}$ denotes the mean of the descriptors, and the range of r is $[-1, 1]$. The correlation coefficient reflects the linear relationship between features, making it suitable for feature selection [51]. In our work, we adopt the discrete form of the correlation coefficient. Unlike the Manhattan distance used in the first layer of detection, the correlation coefficient is dimensionless and unaffected by the units of feature vectors. This makes it particularly advantageous for calculating the similarity between two aggregated feature vectors with different units. Additionally, the correlation coefficient captures the directional differences between features rather than just absolute differences, offering higher-detail resolution and better adaptability to different scenes. Choosing the correlation coefficient for the similarity calculation in the second layer of detection is a more rational and robust design compared to conventional distance-based methods.

Applying the weight vector to the similarity calculation, $\mathbf{v} = \mathbf{w}^{\mathrm{T}} * \mathbf{v}$, enhances the adaptability of the method.

The luminance of an image inherently represents global information. These features must be handled carefully during the detection to preserve this information. Therefore, the weighted features should only be used to calculate similarity when the current frame and the frame under consideration are either excessively dark or bright.

2.4.2. Candidate Frame Detection

This section will discuss the specific process of finding candidate frames. We adopt a multi-layer detection method to achieve efficient and accurate detection. In the first layer of detection, although there are two feature datasets, S_c and S_r, composed of column-aggregation and row-aggregation features, the primary goal is preliminary filtering to quickly narrow the detection range. It does not require precise results, so selecting one feature dataset is sufficient. We use Manhattan distance to select k feature vectors with the smallest distance from the feature set S_r, obtaining a candidate feature set C_r ($\#C_r = k$). The second layer of detection will perform refined screening among the k-selected frames. This includes feature distribution judgment and more accurate similarity calculations. The screening process will comprehensively use aggregation features $\mathbf{v}_r$ and $\mathbf{v}_c$ from each segmented region to calculate similarity.

$$score = \frac{\sum_{i}^{n} r_{\mathbf{v}_r}^{i} * r_{\mathbf{v}_c}^{i}}{n} \quad (21)$$

where n is the total number of image segmentation regions and $r_{\mathbf{v}_r}^{i}$ and $r_{\mathbf{v}_c}^{i}$ are the similarities of the aggregation features $\mathbf{v}_r$ and $\mathbf{v}_c$, respectively.

The detection process is a progressively refined filtering procedure that comprehensively considers feature types and precise similarity calculations, ensuring that the selected candidate frames exhibit high relevance and consistency. The specific implementation steps of the multi-layer detection method can be found in Appendix A. This appendix provides detailed descriptions of implementing the primary function of the technique.

3. Results

This section evaluates the effectiveness of the proposed method through extensive experiments. The experimental results are thoroughly analyzed to quantify the contributions of the proposed method in detection performance, computational efficiency, and localization accuracy, as well as to obtain accurate conclusions.

3.1. Experimental Setup

The datasets used in this study include local datasets, EuRoC [52], KITTI [53], TUM [54], and UMA [55]. The local data were collected using a vehicle equipped with four fisheye cameras mounted on the front and rear bumpers and the left and right mirrors, capturing around-view images in an underground parking lot of an industrial park. The images have a size of 416 × 416 pixels and were equipped at a frequency of 10 Hz. Except for the local dataset, all other datasets used in our work are publicly available. The experimental scenarios encompass indoor and outdoor environments, with images collected from around-view monitors, standard cameras, and fisheye cameras using vehicles, drones, and handheld devices. These scenarios cover conditions of normal luminance, low luminance, and high luminance. A detailed description is provided in Table 1. The diversity of experimental scenarios and the richness of the image data allow for a comprehensive evaluation of the effectiveness of the proposed method. Figure 6 shows some complex scenes in the dataset, such as stitched and rotated images. All experiments have been run on an AMD R5 CPU with 16 GB memory. For each input image, it is vertically divided into eight regions. When constructing the feature descriptor matrix, the grayscale values are quantized into 32 indexes, and the gradient direction are quantized into 64 indexes. To balance accuracy and speed in the first layer of detection, the number of candidate frames is set to 10 ($k = 10$), meaning that 10 candidate frames are selected for further processing in the second layer of detection.

Table 1. Explanation of the datasets used in the experiment.

Dataset	Explanation
Local dataset	Indoor, underground, around-view images, collect by vehicle, normal luminance
EuRoC	Indoor, regular images, collect by drone, low/normal luminance
KITTI	Outdoor, regular images, collect by vehicle, normal luminance
TUM	Indoor, fisheye image, collect by handheld, low/normal luminance
UMA	Outdoor, regular images, collect by handheld, high/low/normal luminance

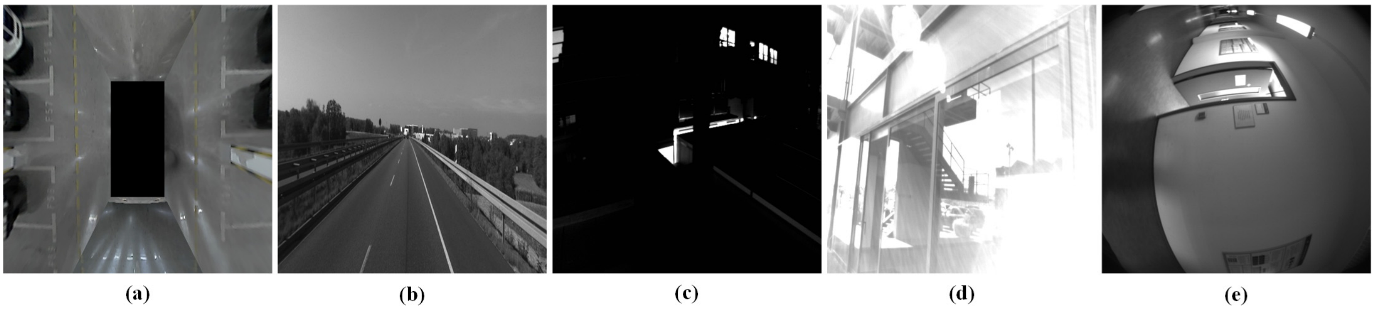


Figure 6. Examples of scenes in dataset. (**a**) Around-view image; (**b**) fast motion and uneven feature distribution; (**c**) low luminance; (**d**) high luminance; (**e**) image rotation.

3.2. Detection Efficiency

First, we evaluate the computational efficiency of the proposed detection method. ORB features combined with BOW2, already used and validated in ORB SLAM3, are currently one of the most widely used loop-closure-detection methods. We will refer to this method as BOW2 in the following text for brevity. We will compare and analyze computational efficiency with this method. We only compare the detection time for more accurate results without geometric verification. Since the proposed loop-closure-detection method has special processing for overly dark and bright images and the UMA dataset contains such images, we include the UMA dataset in the experiment. We select sequences with loop closures from the KITTI and UMA datasets as experimental sequences. The experimental results are shown in Tables 2 and 3 below, using mean and maximum values as evaluation metrics. The reduction is defined as the ratio between the decreased value in the proposed method and the corresponding result from the comparison method. The experimental results show that our proposed method outperforms the comparison method based on both datasets.

Table 2. Comparison of running time (max and mean in *ms*) between the proposed method and BOW2. The sequences are from the KITTI dataset.

	Sequences	**00**	**05**	**06**	**07**
Max	BOW2	13.70	9.35	3.70	6.51
	Proposed method	8.69	5.32	2.38	2.16
	Reduction	36.6%	43.1%	35.7%	66.8%
Mean	BOW2	5.51	4.13	2.95	3.04
	proposed method	4.35	2.52	1.05	1.04
	Reduction	21.0%	43.8%	64.4%	65.8%

Table 3. Comparison of running time (max and mean in *ms*) between the proposed method and BOW2. The sequences are from the UMA dataset.

	Sequences	**Gattaca-1**	**Gattaca-2**	**Parking-1**	**Parking-2**
Max	BOW2	6.47	17.03	6.03	7.16
	Proposed method	3.64	11.36	6.01	6.07
	Reduction	43.7%	33.3%	0.2%	15.2%
Mean	BOW2	3.13	5.72	3.26	3.50
	Proposed method	1.85	5.36	1.71	2.01
	Reduction	40.9%	6.3%	47.5%	42.6%

3.3. Detection Precision

When evaluating classification results, it is common practice to use the True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) to describe the relationship between the predicted and actual results, as shown in Table 4.

Table 4. Relationship between classification results and fact.

Results/Fact	**True**	**False**
True	True positive (TP)	False positive (FP)
False	False negative (FN)	True negative (TN)

Loop detection is a classification task, and commonly used evaluation metrics include precision and recall. Precision represents the proportion of correctly predicted positive samples out of all samples predicted as positive, while recall indicates the proportion of correctly predicted positive samples out of all actual positive samples. By using these two metrics, we can obtain a more comprehensive understanding of the different aspects of the performance of the loop detection method. The formulas for these metrics are provided in (22) and (23).

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

Since the publicly available datasets do not provide ground truth for loop-closure frames, we used an automated tool to obtain the ground truth for this experiment. Figure 7 below shows a schematic of the KITTI experimental trajectory, with the red trajectory indicating the acquired loop-closure ground truth. To enhance the diversity of the experiment, we also included a local dataset collected in an underground parking lot. The data comprise around-view images primarily used for automated parking functions. The experimental trajectory and loop-closure ground truth are shown in Figure 8, with the red trajectory representing the loop-closure ground truth.

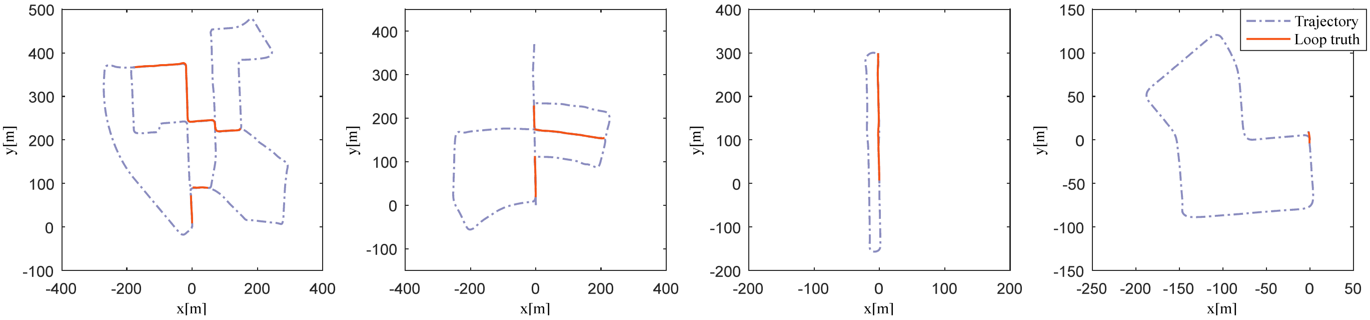


Figure 7. Experimental trajectory and loop-closure ground truth. From left to right: sequences 00, 05, 06, 07 from KITTI.

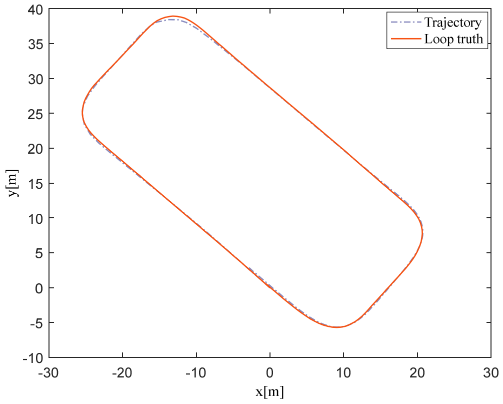


Figure 8. Experimental trajectory and loop-closure ground truth in local datasets.

Precision and recall are mutually exclusive metrics, with the ideal scenario being high recall and high precision. An important metric for evaluating detection performance is the precision–recall curve. The methods selected for comparison are BOW2 and SRLCD [39]. Wang et al. [39] proposed an intriguing loop-detection method, SRLCD, which differs from feature-based methods, like BOW2. Their research suggests that salient regions in images serve as excellent descriptors, aligning with human recognition processes. Moreover, their method matches salient regions in the frequency domain, providing rotation and scale invariance. We tested the proposed method and the comparison methods based on the 00 sequence from the KITTI dataset, which is lengthy and rich in loop-closure scenes, and plotted the precision–recall curves. Figure 9 below shows the experimental results, with the X-axis representing recall and the Y-axis representing precision. The dark purple curve represents the proposed method, while the light purple and pink curves correspond to the results of BOW2 and SRLCD, respectively.

The core idea of SRLCD is salient region recognition. Still, the local dataset consists of around-view images that have been cropped and stitched, lacking distinct salient regions, making SRLCD unsuitable for this scenario. Therefore, we compare only the proposed method and BOW2 on the local dataset.

In addition to the precision–recall curve, another critical metric for evaluating detection performance is the maximum recall at 100% precision, which reflects the boundary performance of the method. A higher recall at this boundary point indicates better detection performance. The experimental results are shown in Figure 10 below, where the X-axis represents the test sequences from KITTI and the Y-axis represents recall. The results demonstrate that the proposed method achieves higher recall at the precision boundary than the comparison methods. Notably, in sequence 07, while the proposed method achieved higher recall than SRLCD, it did not outperform BOW2.

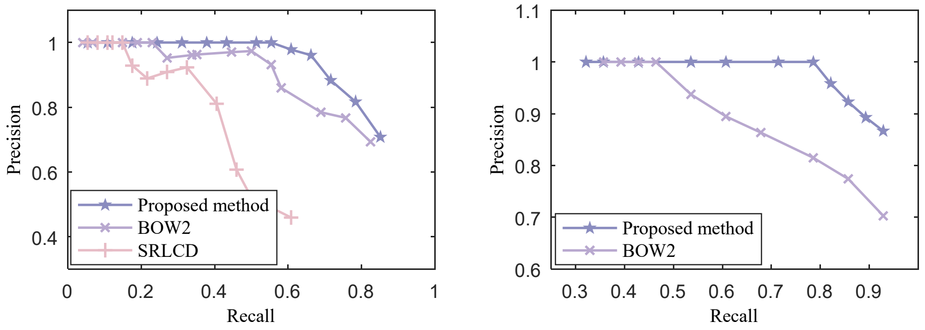


Figure 9. Precision–recall curves based on 00 sequence (**left**) and local dataset (**right**).

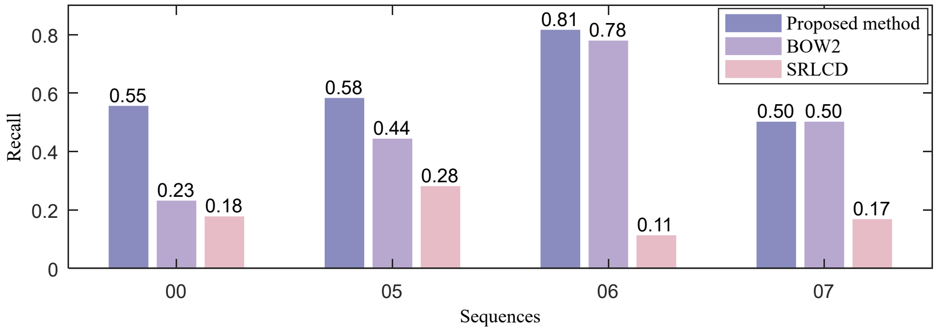


Figure 10. Maximum recall at 100% precision.

To evaluate the detection performance of the proposed method in low-luminance scenarios, we reduced the overall luminance of all image data to 15% of the original. Figure 11 shows the image effects before and after luminance reduction. Since SRLCD performs loop-closure detection based on salient regions and is sensitive to image contrast, it is unsuitable for low-luminance scenarios. Therefore, this experiment only compares the proposed method with BOW2. The experimental results are shown in Figure 12. Overall, the detection performance of the proposed method shows a slight decrease but still outperforms the comparison method.

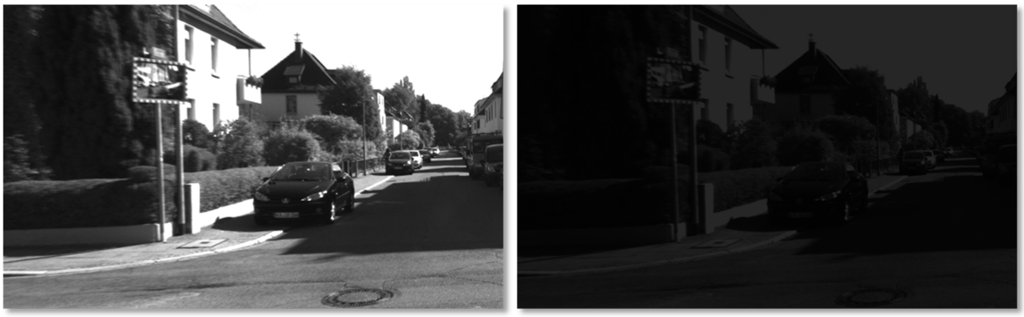


Figure 11. Schematic of image processing. The left image is the original, and the right image has luminance reduced to 15%.

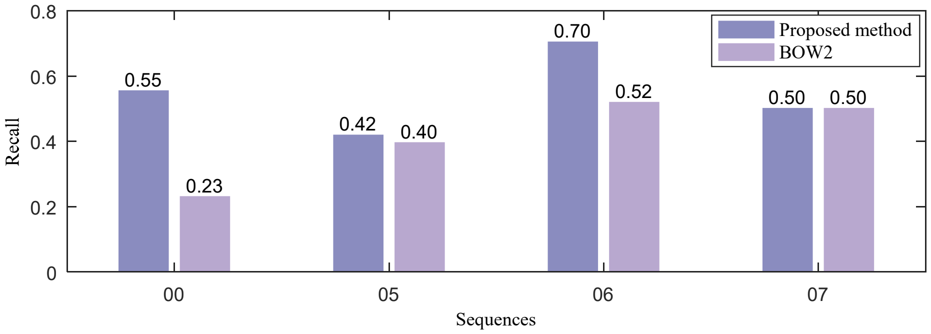


Figure 12. Maximum recall at 100% precision in low-luminance scenarios.

3.4. Trajectory Accuracy

This section primarily evaluates the localization performance and computational efficiency of the proposed method. The metrics for evaluating localization performance include the maximum value and root mean square error (RMSE), while computational efficiency is compared using the average processing time. The first comparison method is ORB-SLAM3, one of the most advanced visual SLAM methods. We also compare the benefits with the method, which integrates line features, to evaluate the proposed method further. As previously described, the principal direction information can represent the geometric properties of feature points, while line features directly reflect the geometric edges of the environment. For the second comparison, we selected ORB-LINE-SLAM [28], one of the advanced SLAMs based on point and line features, where the line feature descriptor used is the Line Band Descriptor (LBD). A comparison with it will further explore the potential of principal direction information in enhancing localization accuracy. We first assess the accuracy and effect of incorporating the principal direction information, then evaluate computational efficiency, and finally, perform a comprehensive comparison of the localization performance of the visual SLAM systems. The datasets used for testing are KITTI, EuRoC, and TUM.

First, we evaluate the localization performance gain after incorporating principal direction information with the same number of ORB extraction levels. The number of levels is 8, with a factor of 1.2. The comparison method and proposed method all use the same number of levels. To clearly display the results, we turned off loop-closure detection and only compared odometry errors. We selected KITTI sequences 03 to 07. Each group of results was run five times, and then, we averaged the Max and RMSE results, as shown in Table 5. The experimental results show that the proposed method outperforms the comparison method ORB SLAM3 based on all test sequences, with particularly significant effects on sequences 05 and 06. The RMSE decreased by 12.62% and 23.06%, and the maximum values decreased by 22.55% and 24.37%, respectively. ORB-LINE-SLAM integrates line features, which improve overall localization performance compared to ORB-SLAM3. In sequence 03, ORB-LINE-SLAM outperforms our proposed method. In sequence 07, the RMSE of our method is reduced by 16.15% at the maximum. In comparison, except for sequence 03, the proposed method demonstrates superior localization performance. Figure 13 shows a 2D trajectory comparison result, selecting the significant sequences 05 and 06. The solid trajectory represents the proposed method, while the dashed and dotted trajectories represent the ground truth and ORB SLAM3. The dash–dot trajectory represents ORB-LINE-SLAM. The results show that the proposed method is closer to the ground truth.

Table 5. Average localization error (RMSE and Max in *m*) over five runs of the executions without loop correction. Experimental sequences are selected from 03 to 07, from the KITTI dataset. The ratios in the reduction represent ORB SLAM and ORB-LINE-SLAM.

	Sequences	03	04	05	06	07
RMSE	ORB SLAM3	1.3514	0.2406	2.1088	2.1038	1.2341
	ORB-LINE-SLAM	1.2060	0.2367	1.9920	1.9270	1.3885
	Proposed method	1.3210	0.2172	1.8426	1.6187	1.1642
	Reduction	2.25%/−9.54%	9.76%/8.24%	12.62%/7.50%	23.06%/16.00%	5.67%/16.15%
Max	ORB SLAM3	2.4148	0.4221	5.1520	4.0300	2.8769
	ORB-LINE-SLAM	2.0625	0.4563	5.1518	3.9385	3.2673
	Proposed method	2.3555	0.3876	3.9905	3.0479	2.6627
	Reduction	2.45%/−14.21	8.16%/15.06%	22.55%/22.54%	24.37%/22.61%	7.45%/18.51%

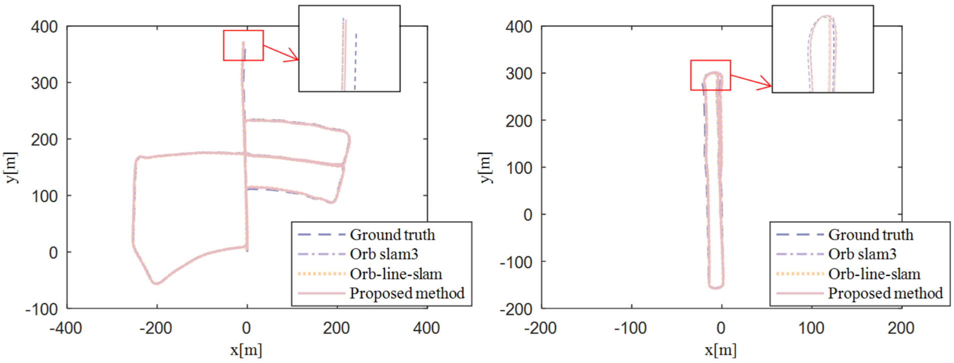


Figure 13. Comparison of trajectory results. The red box indicates a locally magnified trajectory. The left and right images correspond to sequences 05 and 06, both from the KITTI dataset.

There is no definitive method for selecting the number of levels and factors of the feature pyramid. Generally, it relates to the size of the image to be extracted and the scale of the objects to be detected. A larger factor leads to faster image size decay, so the number of pyramid levels is usually not very high. In ORB SLAM3, the number of levels is 8 with a factor of 1.2, which incurs a significant computational cost. In Reference [16], the authors compared the effects of feature extraction with different numbers of levels, from 1 to 5, using the Matier dataset, with feature point accuracy and repeatability as evaluation metrics. The results show that at levels 4 and 5, feature point accuracy and repeatability decrease rapidly. The down-sampling rate chosen is 2 in Reference [16], and at the third level (with level 1 as the starting level), where the effect is relatively good, the image size becomes 1/16 of the original. The benefits brought by higher-level features are limited. When the number of levels and factor are 8 and 1.2, respectively, the image at the highest level will be less than 1/12 of the original (with level 0 as the starting level), within a reasonable range. Considering the above discussion, we choose four levels as the final experimental number of levels to balance efficiency and accuracy, with a factor of 1.54, reducing the area of the top image to about 1/12 of the original.

The primary improvement proposed in this method is in the tracking component, so the focus is on comparing the computational efficiency gains during tracking, including the time taken for pose prediction and the overall tracking process. The test sequences selected are sequences 03, 04, 05, 06, and 07 from the KITTI dataset, with loop closures in sequences 03 and 04. The experimental results are shown in Table 6. The results indicate that reducing the number of levels in the feature pyramid can yield a 24.23% improvement in efficiency compared to ORB SLAM3. The feature-extraction process operates at the pixel level and involves grid partitioning, with the extraction time related to the image size. Level 0 has the most prominent image size and, thus, the highest time cost. Adjusting the number of levels aligns with theoretical expectations, bringing the expected benefits. ORB-LINE-SLAM does not demonstrate an advantage in terms of temporal efficiency. The maximum tracking time reaches 359 ms, indicating that integrating line features imposes a significant computational burden on the system. Additionally, incorporating principal direction information into pose estimation increased the computation time by approximately 1 *ms*.

Table 6. Running time of pose prediction and total tracking (in *ms*). Experimental sequences are selected from 03 to 07, from the KITTI dataset. The ratios in the reduction represent ORB SLAM and ORB-LINE-SLAM.

	Sequences	**03**	**04**	**05**	**06**	**07**	**Mean**
Pose pred	ORB SLAM3	1.924	1.653	1.868	1.770	1.833	1.8096
	ORB-LINE-SLAM	20.311	15.015	18.865	16.424	20.987	18.320
	Proposed method	2.925	2.247	2.950	2.655	2.754	2.7062
Total tracking	ORB SLAM3	92.991	107.006	109.367	105.990	106.371	104.345
	ORB-LINE-SLAM	345.445	347.838	359.972	347.946	358.183	351.877
	Proposed method	73.294	81.968	81.852	79.478	78.274	78.9732
	Reduction	21.18%/78.78%	23.40%/76.44%	25.16%/77.26%	26.41%/77.16%	24.23%/78.15%	24.23%/77.56%

Comprehensive tests are conducted to verify the positioning accuracy of the improved visual SLAM. The datasets chosen are KITTI, EuRoC, and TUM, with detailed descriptions of the dataset scenarios in Table 1. Pose estimation uses the improved pose estimation method. The loop-closure-detection method uses the one proposed in our work, with geometric verification also incorporated after loop-closure detection to prevent false detections. The comparison method ORB SLAM3 uses BOW2 for loop-closure detection with default parameters. The integration of line features significantly reduces the real-time performance of ORB-LINE-SLAM, which is unacceptable for practical use. Therefore, in the comprehensive experiments, we only use ORB-SLAM3 as the comparison method. The experimental results are shown in Table 7. The results demonstrate that the improved system does not decrease the positioning accuracy and even brings slight gains in most sequences.

Table 7. Average localization error (Max/RMSE in *m*) over five runs of the executions.

Dataset	**Seq.**	**Length**	**LC**	**ORB SLAM3**	**Proposed Method**
KITTI	00	3724	√	4.112/1.211	3.299/1.286
	01	2453	–	24.986/15.027	23.617/14.541
	04	394	–	0.422/0.241	0.388/0.217
	05	2206	√	1.688/0.957	1.680/0.918
EuRoC	MH01	81	–	0.161/0.045	0.137/0.044
	MH02	74	–	0.103/0.044	0.103/0.045
	MH03	131	–	0.122/0.056	0.134/0.048
	MH04	92	–	0.209/0.057	0.217/0.069
TUM	room1	146	√	0.128/0.078	0.127/0.076
	room2	142	√	0.130/0.050	0.130/0.051
	corridor1	305	√	0.257/0.107	0.215/0.092
	corridor2	322	√	0.295/0.119	0.252/0.104

Seq.: the experimental sequences from dataset. Length: the length of trajectory in *m*. LC: loop closure. √: provide loop closure.

In summary, to balance the computational efficiency and localization accuracy, we made a trade-off in the number of levels in the feature pyramid based on previous research and innovatively introduced principal direction information and Pareto optimal solutions to ensure that localization accuracy was not decreased. The experimental results align with expectations.

4. Discussion

In this study, we propose an enhanced visual SLAM method based on the principal direction projection of feature points, improving the applicability and computational efficiency. We also integrate global image information, grayscale, and gradient direction to construct feature descriptors for position recognition. Experimental results show that introducing principal direction projection error for position estimation accuracy can achieve satisfactory gains with the same number of levels. Using this method while reducing the number of feature extraction levels improves computational efficiency while maintaining positioning accuracy. The proposed loop-closure-detection method improves detection efficiency and accuracy compared to the comparison method. In this discussion, we will analyze the benefits of the method in terms of detection efficiency, detection accuracy, and localization accuracy.

The second section of the experimental study evaluates the computational efficiency of the proposed loop-detection method by comparing it with BOW2 [38]. To improve computational efficiency, the proposed method optimizes two aspects. First, it reduces the dimensionality of the data by aggregating the 2D feature matrix into a 1D feature vector, reducing storage space and decreasing the complexity of distance calculations, leading to higher efficiency. Second, a multi-layer detection strategy is employed, where the first-layer-detection phase prioritizes speed by using a simple distance calculation to identify candidate frames, and the second-layer-detection phase focuses on accuracy, thereby improving the overall detection efficiency. Regarding the comparison method, Reference [20] introduces a hierarchical vocabulary tree to enhance search efficiency, and in practical use, an inverted index is often used to associate each word with the images in which it appears. Typically, the number of extracted feature points ranges from 1000 to 2000, with each node in the vocabulary tree having ten child nodes and a tree depth of 6. This setup requires many computations during the word calculation phase. Despite the accelerated processing techniques used, the computational load remains significant. Experimental results show that the proposed method outperforms the comparison method regarding computational efficiency.

In the third section, we evaluated the gains in the detection performance of the method. The detection performance of the proposed method is superior to SRLCD [39] and BOW2 [38]. SRLCD uses salient regions for the loop-closure judgment, but unlike semantic information, the detailed features of salient regions are not very prominent, limiting their distinctiveness. Reference [39] also mentioned adding necessary verification modules, but to make the results more intuitive during our experiments, we only compared the detection methods and disabling verification modules, which is also why the precision of SRLCD is limited. Regarding BOW2, in Reference [38], the authors constructed a hierarchical dictionary structure through clustering to avoid the enormous computational cost of directly comparing features with dictionary words one-to-one. While this provides higher search efficiency, it also reduces discrimination. Moreover, for generality, we did not conduct targeted vocabulary training for specific usage scenarios but used a general vocabulary table trained on a large number of various types of images. These two points are the main reasons affecting the detection performance of BOW2. Of course, SRLCD and BOW2 mainly focus on local information, ignoring global information, which also impacts performance. The method proposed in our work demonstrates superior detection performance compared to the comparison methods, which we attribute to the following principal reasons. Firstly, dividing the image into regions to capture more image features indeed brings better effects, consistent with the conclusions in Reference [37]. Secondly, the constructed feature descriptor uses a feature matrix that fuses image grayscale and gradient features, considering both global and detailed information in the image. The features can reflect the current usage environment without targeted training. Lastly, the multi-layer detection approach ensures detection accuracy using a more distinctive distance calculation method in the precise detection stage. In sequence 07 of Figure 10, the proposed method does not demonstrate superior performance compared to BOW2. This is related to image occlusion, as dynamic objects consistently appear in the loop-closure segment. When the proportion of dynamic objects in the images exceeds a certain threshold, the original grayscale and gradient features of the images are altered significantly, negatively impacting the detection accuracy of the proposed method. This also highlights a limitation of the proposed method, as dynamic objects can interfere with the detection precision.

In the comprehensive experiments presented in the fourth section, we primarily compare localization accuracy and computational efficiency. First, by incorporating the principal direction projection error with the same number of pyramid levels, the RMSE of localization results shows an improvement of up to 23.06% compared to ORB SLAMS. Line features can effectively capture geometric information, especially in structured environments with abundant geometric edges [29]. When point features cannot effectively represent the scene characteristics, integrating line features can provide better spatial constraints. The experimental results demonstrate that point-line-based SLAM shows improved performance compared to ORB-SLAM3. However, line features have instability and complexity in matching, such as incomplete line segments and repeated parallel lines, leading to less robust performance in certain scenarios. Additionally, the results indicate that the proposed method, which incorporates principal direction information, outperforms the method integrating line features. We attribute our improvement to the following critical reasons: (1) the principal direction can characterize the global geometric features of feature points and focuses on parts with significant data changes, effectively adding additional structural information constraints. Moreover, diverse constraints can help avoid local optima; (2) the Pareto optimal solution ensures optimal overall performance, enhancing the quality of the results. Next, in the computational efficiency tests, compared to ORB SLAM3, the proposed method achieves an average improvement of 24.23%, which is dependent on the input image size. As detailed in Reference [18], the ORB feature-extraction process is closely related to image size. Our method reduces the number of feature extraction levels, decreasing the computational load associated with feature extraction. Since the first level (level 0) has the largest image size and accounts for the majority of the computation time, reducing the number of levels by half does not correspond to a proportional reduction in the computational time. The integration of line features results in significantly higher computational costs. It necessitates the simultaneous extraction and matching of both point and line features, increasing the complexity of the entire process. The preprocessing of line features, such as line segment merging and removal, contributes non-negligibly to the computational burden. The increase in the computational cost raises the probability of unreliable estimations, which limits the overall performance improvement of the system. Finally, extensive multi-scenario testing on various datasets demonstrates that our method improves the computational efficiency without compromising the localization accuracy. Although reducing the number of pyramid levels might introduce feature-extraction and matching errors, which could lead to pose estimation deviations, including principal direction information compensates for these potential inaccuracies. The experimental results align with our expectations.

5. Conclusions

In visual SLAM, both operational efficiency and localization accuracy are crucial. The research purpose of our work is to improve operational efficiency while maintaining high positioning accuracy. We address two specific issues: (1) the multi-level feature extraction in visual SLAM significantly contributes to the computational cost of the front-end. (2) The BOW-based loop closure detection method requires pre-trained vocabularies, limiting its practical use. Therefore, we propose an enhanced visual SLAM method that incorporates principal direction information during pose estimation to ensure accuracy and utilizes grayscale and gradient information to construct feature descriptors for loop-closure detection.

Based on our research and experimental evaluation, we can obtain the following conclusions:

1. Incorporating principal direction information during pose estimation can improve the estimation accuracy. When maintaining the same number of feature-extraction layers, the RMSE of the position estimation can be reduced by up to 23% in the best case

compared to ORB SLAM3 with only an additional approximate 1 ms of computation time, providing satisfactory results.

2. Based on aggregated feature descriptors, the proposed loop-closure-detection method outperforms comparative methods in both computational efficiency and detection accuracy. It is adaptable to various image types, including around-view and fisheye images.
3. The experimental data were collected from various devices with different image types and application scenarios, demonstrating the strong generalization capability of the proposed enhanced method across different sensors and environments. This aligns well with the design purposes.

Certainly, the proposed detection method does have its limitations, with occlusion being the most significant challenge. Large areas of occlusion can alter the global features of an image, impacting the detection precision of the method. In future work, we plan to mitigate the effects of dynamic occlusion by incorporating stereo-based joint detection or image sequence-based methods. Integrating contextual information could also further enhance the detection performance. We will further optimize the storage structure of the aggregated descriptors to accelerate detection, particularly for long-term detection tasks. Additionally, we will continue to focus on improving the operational efficiency of SLAM systems.

Author Contributions: Conceptualization, Y.Y. and F.L.; methodology, Y.Y., F.L., and X.L.; software, Y.Y., F.L., and J.C.; validation, Y.Y. and J.C.; formal analysis, Y.Y. and F.L.; investigation, Y.Y. and F.L.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y. and X.L.; visualization, Y.Y.; supervision, X.L.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number U20A20193.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

To provide a clearer presentation of the multi-layer detection algorithm, we have detailed its implementation steps in Algorithm A1. The function firstLayerFilter primarily narrows the range of candidate frames, using only one aggregated feature vector and a fast similarity calculation method. The function secondLayerFilter utilizes all feature vectors and a more precise similarity calculation to obtain the final reliable candidate frames.

Algorithm A1: Multi-layer detection algorithm

```
Input:
kf_db: keyframes database
cur_kf: current keyframe
Output:
n: the number of candidate frames
v: the vector of candidate frames
function firstLayerFilter (kf_db, cur_kf):
    for each frame in kf_db:
        dis ← manhattanDistance (frame.vr, cur_kf.vr)–vr is row aggregation feature vector
        if dis < distanceHeap.top( ):–Max-Heap. The maximum value is at the top.
            distanceHeap.pop ()
            distanceHeap.addDis (dis)
            end if
        end for
        return distanceHeap[1: k]
function secondLayerFilter (distanceHeap[1: k], &n, &v):
    n ← 0
    for each frame in distanceHeap:
        if the contrast of frame and cur_kf is too low:
            frame.vr ← weightFunction (frame.vr)
            score_vr ← similarityFunction(frame. vr, cur_kf. vr)
            score_vc ← similarityFunction (frame.vc, cur_kf.vc)–vc is column aggregation feature vector
            if score_vr * score_vc > threshold:
                n++
                v.addFrame (frame)
            end if
        else
            score_vr ← similarityFunction (frame.vr, cur_kf.vr)
            score_vc ← similarityFunction (frame.vc, cur_kf.vc)
            if score_vr * score_vc > threshold:
                n++
                v.addFrame (frame)
            end if
        end if
    end for
function mian (kf_db, cur_kf):
    distanceHeap ← firstLayerFilter (kf_db, cur_kf)
    secondLayerFilter (distanceHeap[1: k], n, v)
    return n, v
```

References

1. Al-Tawil, B.; Hempel, T.; Abdelrahman, A.; Al-Hamadi, A. A review of visual SLAM for robotics: Evolution, properties, and future applications. *Front. Robot. AI* **2024**, *11*, 1347985. [CrossRef] [PubMed]
2. Yue, X.; Zhang, Y.; Chen, J.; Chen, J.; Zhou, X.; He, M. LiDAR-based SLAM for robotic mapping: State of the art and new frontiers. *Ind. Robot. Int. J. Robot. Res. Appl.* **2024**, *51*, 196–205. [CrossRef]
3. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
4. Peng, X.; Liu, Z.; Li, W.; Tan, P.; Cho, S.Y.; Wang, Q. Dvi-slam: A dual visual inertial slam network. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024.
5. Reinke, A.; Palieri, M.; Morrell, B.; Chang, Y.; Ebadi, K.; Carlone, L.; Agha-Mohammadi, A.A. Locus 2.0: Robust and computationally efficient lidar odometry for real-time 3d mapping. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9043–9050. [CrossRef]
6. He, D.; Xu, W.; Chen, N.; Kong, F.; Yuan, C.; Zhang, F. Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry. *Adv. Intell. Syst.* **2023**, *5*, 2200459. [CrossRef]
7. Chen, L.; Li, Y.; Li, L.; Qi, S.; Zhou, J.; Tang, Y.; Yang, J.; Xin, J. High-precision positioning, perception and safe navigation for automated heavy-duty mining trucks. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4644–4656. [CrossRef]
8. Xu, H.; Liu, P.; Chen, X.; Shen, S. D2SLAM: Decentralized and Distributed Collaborative Visual-Inertial SLAM System for Aerial Swarm. *IEEE Trans. Robot.* **2024**, *40*, 3445–3464. [CrossRef]
9. Ouyang, M.; Shi, X.; Wang, Y.; Tian, Y.; Shen, Y.; Wang, D.; Wang, P.; Cao, Z. A collaborative visual SLAM framework for service robots. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021.
10. Yarovoi, A.; Cho, Y.K. Review of simultaneous localization and mapping (SLAM) for construction robotics applications. *Autom. Constr.* **2024**, *162*, 105344. [CrossRef]
11. Moreno–Valenzuela, J.; Torres–Torres, C. Adaptive chaotification of robot manipulators via neural networks with experimental evaluations. *Neurocomputing* **2016**, *182*, 56–65. [CrossRef]
12. Zhu, B.; Yu, A.; Hou, B.; Li, G.; Zhang, Y. A Novel Visual SLAM Based on Multiple Deep Neural Networks. *Appl. Sci.* **2023**, *13*, 9630. [CrossRef]
13. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef] [PubMed]
14. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13 November 2007.

15. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May 2014.
16. Zubizarreta, J.; Aguinaga, I.; Montiel, J.M.M. Direct sparse mapping. *IEEE Trans. Robot.* **2020**, *36*, 1363–1370. [CrossRef]
17. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International conference on computer vision (ICCV), Barcelona, Spain, 6 November 2011.
18. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
19. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
20. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
21. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
22. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
23. Qin, T.; Cao, S.; Pan, J.; Shen, S. A general optimization-based framework for global pose estimation with multiple sensors. *arXiv* **2019**, arXiv:1901.03642.
24. Chen, P.; Guan, W.; Lu, P. Esvio: Event-based stereo visual inertial odometry. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3661–3668. [CrossRef]
25. Zhou, Y.; Gallego, G.; Shen, S. Event-based stereo visual odometry. *IEEE Trans. Robot.* **2021**, *37*, 1433–1450. [CrossRef]
26. Chen, L.; Ling, Z.; Gao, Y.; Sun, R.; Jin, S. A real-time semantic visual SLAM for dynamic environment based on deep learning and dynamic probabilistic propagation. *Complex Intell. Syst.* **2023**, *9*, 5653–5677. [CrossRef]
27. Wang, Y.; Liu, X.; Zhao, M.; Xu, X. VIS-SLAM: A Real-Time Dynamic SLAM Algorithm Based on the Fusion of Visual, Inertial, and Semantic Information. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 163. [CrossRef]
28. Alamanos, I.; Tzafestas, C. ORB-LINE-SLAM: An Open-Source Stereo Visual SLAM System with Point and Line Features. *TechRxiv* **2023**. [CrossRef]
29. Chen, Q.; Cao, Y.; Hou, J.; Li, G.; Qiu, S.; Chen, B.; Lu, H.; Pu, J. VPL-SLAM: A Vertical Line Supported Point Line Monocular SLAM System. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9749–9761. [CrossRef]
30. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
31. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
32. Brown, M.; Szeliski, R.; Winder, S. Multi-image matching using multi-scale oriented patches. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2005.
33. Taranco, R.; Arnau, J.M.; González, A. LOCATOR: Low-power ORB accelerator for autonomous cars. *J. Parallel Distrib. Comput.* **2023**, *174*, 32–45. [CrossRef]
34. Usenko, V.; Demmel, N.; Schubert, D.; Stückler, J.; Cremers, D. Visual-inertial mapping with non-linear factor recovery. *IEEE Robot. Autom. Lett.* **2019**, *5*, 422–429. [CrossRef]
35. Geneva, P.; Eckenhoff, K.; Lee, W.; Yang, Y.; Huang, G. Openvins: A research platform for visual-inertial estimation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May 2020.
36. Li, Y.; Wei, W.; Zhu, H. Incremental Bag of Words with Gradient Orientation Histogram for Appearance-Based Loop Closure Detection. *Appl. Sci.* **2023**, *13*, 6481. [CrossRef]
37. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2005.
38. Zhang, J.; Ai, D.; Xiang, Y.; Wang, Y.; Chen, X.; Chang, X. Bag-of-words based loop-closure detection in visual SLAM. In Proceedings of the 2018 Advanced Optical Imaging Technologies, Bellingham, DC, USA, 17–20 September 2018.
39. Wang, H.; Wang, C.; Xie, L. Online visual place recognition via saliency re-identification. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, LA, USA, 25–29 October 2020.
40. Guclu, O.; Can, A.B. Fast and effective loop closure detection to improve SLAM performance. *J. Intell. Robot. Syst.* **2019**, *93*, 495–517. [CrossRef]
41. Ying, T.; Yan, H.; Li, Z.; Shi, K.; Feng, X. Loop closure detection based on image covariance matrix matching for visual SLAM. *Int. J. Control. Autom. Syst.* **2021**, *19*, 3708–3719. [CrossRef]
42. Chen, Z.; Lam, O.; Jacobson, A.; Milford, M. Convolutional neural network-based place recognition. *arXiv* **2014**, arXiv:1411.1509.
43. Gao, X.; Zhang, T. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Auton. Robot.* **2017**, *41*, 1–18. [CrossRef]
44. Samadzadeh, A.; Nickabadi, A. Srvio: Super robust visual inertial odometry for dynamic environments and challenging loop-closure conditions. *IEEE Trans. Robot.* **2023**, *39*, 2878–2891. [CrossRef]
45. Hines, A.D.; Stratton, P.G.; Milford, M.; Fischer, T. VPRTempo: A fast temporally encoded spiking neural network for visual place recognition. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024.

46. Cheng, X.; Zhang, Y.; Kang, M.; Wang, J.; Jiao, J.; Dong, L.; Jiao, L. A Semantic Spatial Structure-Based Loop Detection Algorithm for Visual Environmental Sensing. *Remote Sens.* **2024**, *16*, 1720. [CrossRef]
47. Li, J.; Wang, P.; Ni, C.; Zhang, D.; Hao, W. Loop Closure Detection for Mobile Robot based on Multidimensional Image Feature Fusion. *Machines* **2022**, *11*, 16. [CrossRef]
48. Arshad, S.; Kim, G.W. Semantics aware loop closure detection in visual SLAM. In Proceedings of the 2021 21st International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 12–15 October 2021.
49. Yuan, Z.; Xu, K.; Zhou, X.; Deng, B.; Ma, Y. SVG-Loop: Semantic–visual–geometric information-based loop closure detection. *Remote Sens.* **2021**, *13*, 3520. [CrossRef]
50. Chen, H.; Zhang, G.; Ye, Y. Semantic loop closure detection with instance-level inconsistency removal in dynamic industrial scenes. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2030–2040. [CrossRef]
51. Zhou, H.; Wang, X.; Zhu, R. Feature selection based on mutual information with correlation coefficient. *Appl. Intell.* **2020**, *52*, 5457–5474. [CrossRef]
52. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [CrossRef]
53. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
54. Klenk, S.; Chui, J.; Demmel, N.; Cremers, D. Tum-vie: The tum stereo visual-inertial event dataset. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021.
55. Zuniga-Noël, D.; Jaenal, A.; Gomez-Ojeda, R.; Gonzalez-Jimenez, J. The UMA-VI dataset: Visual–inertial odometry in low-textured and dynamic illumination environments. *Int. J. Robot. Res.* **2020**, *39*, 1052–1060. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.