SCIPEDIA

An Improved ORB-Based Multi-View Stereo Matching Algorithm for Accurate Visual SLAM

Xiaowei Hu* and Dijie Xue

College of Electronic Information Engineering, Sias University, Zhengzhou, China

INFORMATION

Keywords:
Visual SLAM
multi-view image features
stereo matching
visual odometer
loop back detection
back-end optimization

DOI: 10.23967/j.rimni.2026.10.71720

Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

An Improved ORB-Based Multi-View Stereo Matching Algorithm for Accurate Visual SLAM

Xiaowei Hu* **and Dijie Xue**

College of Electronic Information Engineering, Sias University, Zhengzhou, China

ABSTRACT

This study addresses image matching in visual Simultaneous Localization and Mapping (SLAM) to enhance synchronous positioning and mapping using a multi-view stereo matching algorithm. Utilizing a vision odometer and a binocular camera, multi-view stereo images are captured. An improved ORB (Oriented FAST and Rotated BRIEF) algorithm extracts feature points and establishes binary descriptors, with similarity computed using Euclidean distance to enable stereo matching. Mismatches are filtered using parallax constraints and corrected through triangulation. Loopback detection minimizes cumulative drift in camera position, improving spatial perception. The back-end optimization employs graph optimization theory to refine the position and pose of the binocular camera and landmarks, addressing random noise and errors. Experimental results indicate effective feature extraction, with mismatches limited to three, overall positioning error under 100 mm, and directional error within 2°.

OPEN ACCESS

Received: 11/08/2025

Accepted: 14/01/2026

DOI
10.23967/j.rimni.2026.10.71720

Keywords:
Visual SLAM
multi-view image features
stereo matching
visual odometer
loop back detection
back-end optimization

1 Introduction

Authors should use this Microsoft Word template when preparing their manuscripts for submission. The 3D measurement and 3D reconstruction technology based on machine vision is a new and practical technology. The visual Simultaneous localization and mapping (SLAM) technology is the main component of machine vision technology [1]. Since the 1980s, visual SLAM has been concerned by researchers in the field of computer vision and robotics. Visual SLAM is a technology that uses images collected by visual sensors (such as monocular cameras, binocular cameras, depth cameras, etc.) to locate and navigate in an unknown environment. It can build a map of the surrounding environment in the process of motion and estimate its own motion at the same time [2]. Vision SLAM is widely used in automated guided vehicles (AGV), UAV and robot navigation, precision industrial measurement, object recognition, virtual reality, scene reconstruction, surveying, and other fields, and is one of the key technologies in the future artificial intelligence era. In recent years, technology based on image feature point extraction and matching has become increasingly widely used in visual SLAM. The visual SLAM system estimates the camera's position and pose by extracting feature points and stereo matching [3] on the images collected by the sensors, and using the n-point perspective (PnP), iterative

*Correspondence: Xiaowei Hu (xiaowei_hu11@outlook.com).

closest point (ICP), etc., so as to realize the role of the visual odometer [4]. Stereo matching is to obtain the three-dimensional information of objects by matching the pixels between the visual SLAM system images, obtaining the disparity map through the principle of triangulation, and calculating the offset between pixels. Image matching in visual SLAM is generally considered the most difficult and critical problem [5], and research focuses on improving matching accuracy.

Scholars in related fields have conducted extensive research on image matching methods. Hao et al. [6] proposed a visible and infrared image matching method based on CycleGAN-SIFT, which used CycleGAN to generate pseudo infrared images on the basis of visible and infrared images by transferring, learning, and sharing weights, and used the SIFT feature extraction algorithm to extract feature points of pseudo infrared images and infrared images and match them. This algorithm requires the establishment of high-dimensional descriptors, requires a large amount of memory space, and has a long running time. Jia et al. [7] studied the dense descriptor of multimodal image matching based on structure tensor voting. This descriptor was based on the voting scheme of the structure tensor, which effectively captures the geometric structure of the image. It not only preserved illumination and contrast invariance, but was also robust to degradation caused by significant noise. Based on the proposed dense feature descriptor and improved similarity measure, a robust and practical image matching algorithm could be obtained. This method shortens the time required to establish descriptors, but its feature point detection lacks scale and rotation invariance. Liao et al. [8] studied the SIFT feature extraction and matching algorithm based on sequence minimum optimization, and selected the threshold of Euclidean distance ratio for rough matching on the original SIFT. Combined with the SMO algorithm of support vector machine, the feature matching operator in the SIFT algorithm was improved, and image matching was realized on this basis. This method requires a large amount of storage space and a high demand for light changes. Aiming at the above problems, the stereo matching algorithm of multi-view image features for visual SLAM is studied, and the application performance of this method is verified through experiments. Recent work by Zhang et al. [9] demonstrated that adaptive keypoint extraction significantly enhances visual odometry performance in diverse environments. Similarly, Yang et al. [10] emphasized the importance of accurate stereo disparity estimation, which directly supports robust feature matching in SLAM systems. Gong et al. [11] emphasized the significance of tightly-coupled multi-sensor SLAM systems for real-time ultraprecise localization in complex environments. Hybrid autonomation Sitaraman and Khalid [12] gave it a spin and have thus proposed an efficient method for navigation out of AutoNav, LiDAR-SLAM, and DenseNet with Leaky ReLU. In the multi-view stereo SLAM from ORB perspective, the LiDAR-SLAM and deep-feature approaches involved have come to enhance ORB matching; hence, the detection of improvement in robustness, accuracy, and stability under complex scenarios is reported.

1.1 Research Gap

Despite the fact that many feature extraction and matching algorithms, such as SIFT, traditional ORB, and SMO-SIFT, have been applied in visual SLAM, they suffer from certain chief limitations. Bio-inspired techniques, including genetic algorithms and particle swarm optimization, offer potential to evolve more efficient and adaptive feature matching methods, especially in environments with high dynamic changes. Chief among these are large storage requirements, a lack of scale and rotation invariance, high computational costs, and vulnerability to mismatches when faced with low texture or changes in illumination. Such shortcomings indeed hinder one from accomplishing true robust and real-time performance in SLAM, especially in dynamic or low-computing environments.

That is, there is a huge gap in these shortcomings in achieving real robust and real-time performance in SLAM, particularly when in dynamic and low-computing environments.

1.2 Objectives

The present study aims to overcome the above limitations by suggesting an improved ORB-based stereo matching algorithm for multi-view images in visual SLAM. The specific objectives are:

- To design an improved ORB feature extraction method having higher scale and rotation invariance.
- To include disparity-constrained mismatch elimination and triangulation-based correction for reliable feature matching
- To integrate loop closure detection and graph-based back-end optimization to reduce cumulative drift and enhance accuracy.

1.3 Significance

The significance of the research lies in its application-based nature; autonomous navigation and robotics are practical areas of implementation. The suggested algorithm, keeping mismatches to fewer than three per matching sequence, total positioning errors less than 100 mm, and angular error within 2°, is computationally inexpensive and serves as a very viable and potentially very efficient solution to this problem. This solution can be applied to real-world scenarios, such as robot navigation, UAV localization, and intelligent industrial systems.

2 Materials and Methods

2.1 Stereo Matching Architecture of Multi-View Image Feature for Visual SLAM

Stereo matching of multi-view image features is the basis of vision SLAM implementation. Based on the image matching results, synchronous positioning and mapping can be achieved. The stereo matching architecture of multi-view image feature for visual SLAM is shown in Fig. 1. The diagram describes an integration of the visual odometer, loopback detection, and back-end optimization modules. Source: authors' implementation and experimental setup.

The stereo matching architecture of multi-view image features for visual SLAM mainly consists of the following three parts:

(1) Visual odometer. The function of the visual odometer in the visual SLAM is equivalent to that of the direct odometer in the laser SLAM. The purpose is to estimate the camera's own pose and motion through image frame matching, and track the feature points to establish a preliminary discrete map. The visual odometer is also called the front end. The visual odometer is the basis for the realization of the stereo matching algorithm of multi-view image feature for visual SLAM, including four links:

The first link: image acquisition and processing. Different cameras in the binocular camera are used to simultaneously acquire multi-view stereo images of the target (view angle A image and view angle B image), and denoise the acquired images [13].

The second link: feature point extraction. The improved oriented brief (ORB) algorithm is used to extract feature points and establish binary descriptors on the view A image and the view B image, respectively.

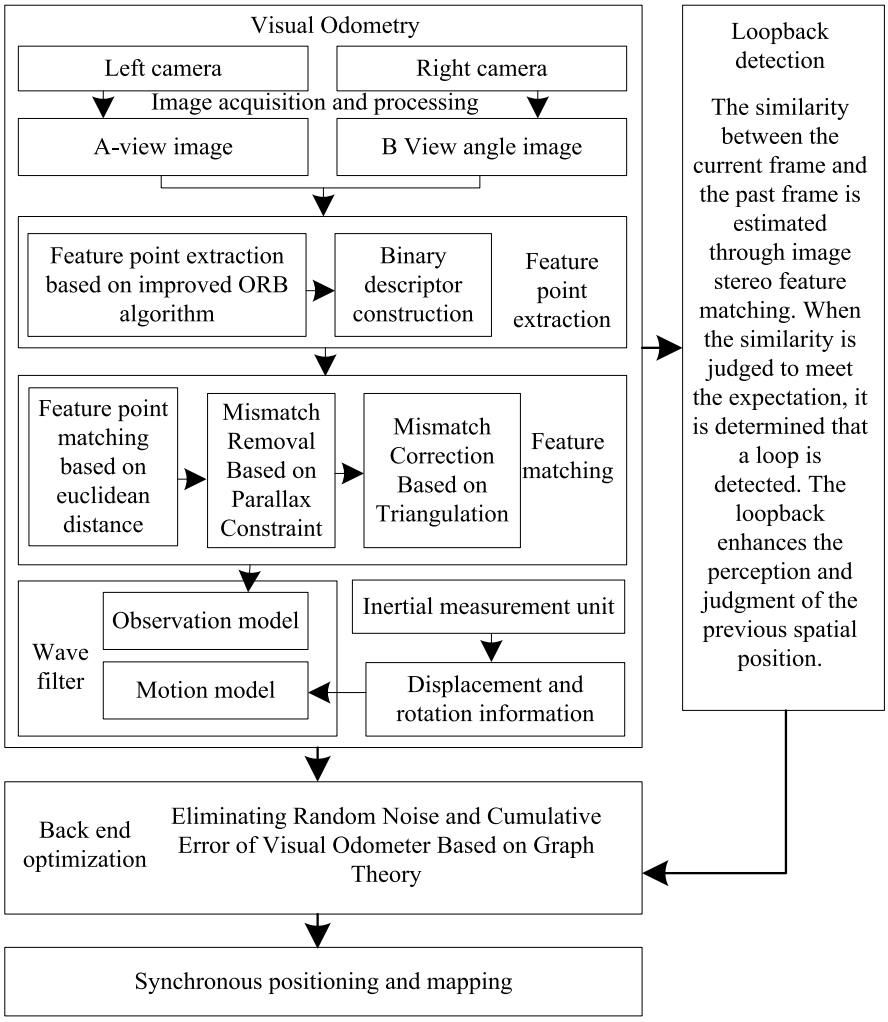

Figure 1: Stereo matching architecture of multi-view image features for visual SLAM

The third link: feature point matching. Euclidean distance is used to calculate the similarity of image feature points from different perspectives and match these binary descriptors. For the mismatched feature points, the mismatched points are eliminated based on the disparity constraint, and the mismatched points are corrected by triangulation.

The fourth link: pose and motion estimation. Since the internal and external parameters of the binocular camera are known, the observation model of the surrounding environment can be built through the world coordinates of the stereo matching results of the multi-view image features. The inertial measurement unit includes an accelerometer and a gyroscope, which can calculate the acceleration in three directions, obtain the rotation information of the binocular camera, and then build the motion model (position and attitude information) of the binocular camera in the visual SLAM. The observation model and the motion model are filtered through the filter, and the observation model data and the motion model data are fused to locate the binocular camera and build a preliminary discrete map.

(2) Loopback detection. Loopback detection is one of the important components of visual SLAM, also known as closed-loop detection, which represents the ability of binocular cameras in the visual odometer to identify whether they have reached the previous position. The loopback detection is performed through the stereo matching algorithm of multi-view image feature. Once a loopback is detected, it will immediately feed this information back to the back-end optimization module for processing, which can significantly reduce the cumulative error of binocular camera pose estimation in the visual odometer, thereby improving the stereo matching effect of multi-view image features in the visual SLAM.

(3) Back end optimization. As the core component of contemporary SLAM research, the back-end optimization has evolved from filter theory to nonlinear graph optimization. The purpose is to obtain the camera pose by collecting the front-end visual odometer, capture the loopback information, and optimize the binocular camera pose and landmark points in the visual odometer as a whole. Because it is behind the visual odometer and is mainly responsible for optimization, it is called the back-end.

2.2 Stereo Matching of Multi-View Image Features during Visual SLAM Mapping

2.2.1 Feature Point Extraction of Multi-View Image

Feature point extraction is the basis of stereo matching of multi-view image features in the visual SLAM mapping process [14]. For the target object images under different visual angles in the visual SLAM mapping process, the ORB algorithm is used to extract the feature points in the view A image and view B image of the target object in the stereo matching process of multi-view image feature for visual SLAM. ORB algorithm is an image feature point detection algorithm based on FAST key points and BRIEF descriptors proposed in 2011. ORB feature extraction is divided into two steps: Features from Accelerated Segment Test (FAST) feature point detection and Binary Robust Independent Elemental Features (BRIEF) descriptor calculation [15]. Firstly, it needs to construct a multi-scale space in the process of visual SLAM mapping, to detect feature points in each level of scale space by improving the FAST algorithm [16]. Then, a BRIEF with rotation is used to generate binary descriptors of local features of object images from different perspectives during visual SLAM mapping.

Aiming at the problem that the ORB algorithm does not have scale invariance and a high error matching rate, this paper improves the ORB algorithm and proposes a new ORB feature detection algorithm to extract feature points in view A image and view B image of the target object.

Firstly, pyramids for the A-view image and B-view image of the target object in the visual SLAM are built [17]. The pyramids include four ordinary layers and four intermediate layers. The scaling scale of each layer is shown in Formula (1):

$$\begin{cases} t\left(g_i\right) = 2^i \\ t\left(a_i\right) = 2^i \times 1.5 \end{cases} \tag{1}$$

In Formula (1), g_i and a represent the common layer and middle layer of target object image t under different viewing angles, $i = 1, 2, 3, 4$ represents the different layers of target object image under different viewing angles, and 1.5 represents the scale coefficient. The transformation relationship of each layer is shown in Table 1.

Table 1 shows that for the pyramid with an 8-layer structure of target object images from different perspectives in the visual SLAM mapping process, the scaling factor between ordinary layers is twice that between intermediate layers.

Table 1: Image pyramid

Image Layer	Width	Height
c_0	w	h
d_0	2/3w	2/3h
c_1	1/2w	1/2h
d_1	1/3w	1/3h
c_2	1/4w	1/4h
d_2	1/6w	1/6h
c_3	1/8w	1/8h
d_3	1/12w	1/12h

In the process of visual SLAM mapping, the FAST 9-16 mask shape is selected for feature point detection for each layer of the image, which basically requires that the gray value of a pixel in the target object image under different viewing angles is larger than the nine consecutive 16 pixel circles around it.

Firstly, let the FAST 9-16 detector use the same threshold to determine the potential area of interest in the common layer and middle layer of the target object image under different viewing angles. Then, the points in these regions are subject to non-maximum control in the location and scale space: the points in question need to meet the maximum FAST score of 26 neighborhood points in the same layer 8 neighborhood and up and down.

Regarding the computational load, the multi-scale pyramid construction and the FAST detection method on multiple layers have a significant effect on processing time. Although this method enhances the strength of the algorithm in identifying scale-invariant features, it is also more memory and computationally intensive, particularly when dealing with high-resolution data or in real-time applications where fast processing is needed. In the case of real-time systems, it can result in performance trade-offs where the algorithm must trade-off between speed and accuracy.

In view of visual SLAM, adjacent layers of images undergo being differently discretized. At the maximum-score layer, an edge length of 2 pixels is used for interPoLation at the region boundary. In order to obtain FAST score of virtual layer d_{-1} below c_0, FAST 5-8 mask is applied on c_0. However, the score of d_{-1} is not required to be lower than that of checkpoint in common layer c_0.

Considering that the saliency of the target object image under different visual angles in the visual SLAM mapping process is not only a continuous quantity of the entire target object image, but also changes along the scale dimension, sub pixel and continuous scale thinning is performed for each detected maximum value. To limit the complexity of the thinning process, firstly the two-dimensional quadratic function of the least squares is fit to three scores (obtained in the key point layer, the upper layer and the lower layer), so as to maximize the significance of the refinement of the three sub pixels. Next, these refined scores are used to fit the one-dimensional parabola along the scale axis to generate the maximum score estimate and the maximum scale estimate. As the last step, the image coordinates in the target object image layer under different viewing angles are re-interpolated and determined, which solves the scale invariance.

Since the feature points collected by FAST do not have directionality, a simple and effective angular direction is used to measure a strength centroid to solve this problem.

It is assumed that the strength of the angle deviates from its center, and this vector can be used to calculate the direction. For a feature point P in the target object image from different perspectives during visual SLAM mapping, the neighborhood pixel of P is defined as:

$$m_{pq} = \sum_{x,y \in r}^{n} x^p y^q h(x, y) \tag{2}$$

In Formula (2), $h(x, y)$ is the gray value at point (x, y) of the target object image under different visual angles in the visual SLAM, $p = (0, 1, \cdots)$ and $q = (0, 1, \cdots)$ determine the order of the gray moment, and the centroid of the gray moment is:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{3}$$

In Formula (3), the subscripts 1 and 0 represent the order of the gray moment.

In the process of visual SLAM mapping, a vector is constructed from the center of the target object image's corner point to the center of mass under different visual angles. The included angle β is calculated as follows:

$$\beta = a \tan 2 (m_{01}, m_{10}) \tag{4}$$

In order to improve the visual SLAM and improve the rotation invariance in the ORB method, it is necessary to ensure that x and y are within a circular region with a radius of r, that is, $x, y \in (-r, r)$, and r is equal to the neighborhood radius.

The Rotated BRIEF (rBRIEF) binary feature point descriptor in ORB algorithm is used to describe the feature points of the object image under different visual angles, and the description feature points are obtained.

2.2.2 Stereo Matching of Multi-View Image Feature Point

After obtaining the feature points of target images from different perspectives in the visual SLAM mapping process, the Euclidean distance based matching method is used to complete the stereo matching of multi-view image features for visual SLAM.

In visual SLAM, only feature matching of images under two positions and orientations can establish data association of different positions and orientations in SLAM, so as to conduct subsequent pose estimation and pose optimization. Generally, the similarity of feature points of two images is evaluated by measuring the distance between feature points of view A image and view B image. For the floating point feature vector descriptor, Euclidean distance is used to calculate the similarity of the target object's image feature points under different visual angles, namely:

$$d(A, B) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} U_i (p_i)^2 - U_j \left(p_j\right)^2} \tag{5}$$

In Formula (5), $d(A, B)$ is the European distance between the view A image and the view B image of the target object in the visual SLA; p_i and p_j are the i-th and j-th feature pixels in the view A image and view B image of the target object, respectively; n, m is the total number of feature pixels in the view A image and view B image of the target object, respectively; $U_i(p_i)$ and $U_j\left(p_j\right)$ are feature vectors of feature descriptors corresponding to pixel points p_i and p_j, respectively.

After generating the descriptor of the image to be matched in the visual SLAM mapping process, it can calculate the Euclidean distance between the view A image and the view B image descriptor. If the ratio of the nearest distance to the next nearest distance is less than the set threshold, the matching is completed.

For binary string feature descriptors, Hamming distance is usually used for measurement. Hamming distance between two binary strings refers to the number of different digits, namely:

$$d_h(A,B) = \sum_{i=1}^{n}\sum_{j=1}^{m} U_i(p_i) \otimes U_j(p_j) \tag{6}$$

In Formula (6), $d_h(A,B)$ is the hamming distance between the view A image and the view B image of the target object in the visual SLA; $\otimes$ is the exclusive OR operation on the specified position of the feature vector.

The similarity between the two image feature points under different visual angles is calculated by performing an XOR operation on the feature point descriptors of the target object view A image and view B image during the visual SLA mapping process. When the similarity between the two image feature points under different visual angles in the visual SLA is greater than the set threshold, it is considered as a matching point.

2.2.3 Removal and Correction of Mismatched Points

In the process of visual SLAM mapping, two stereoscopic images (view A image and view B image) are generally obtained from different angles in the process of multi-view image acquisition. In order to obtain the three-dimensional coordinates of a point of the target object in three-dimensional space, the corresponding point of the point needs to exist on both stereoscopic images [18]. When low texture areas and repeated texture areas appear in two stereo images during visual SLAM mapping, there may be other feature points [19] with similar gray distribution around the feature points in these two areas, which results in similar vectors of feature point description, and it is likely that there will be a mismatch. Disparity constraints are used directly after feature point matching to minimize disparities.

In the vision SLA, for the same point $P(X, Y, Z)$ of the same target object in the view A image and the view B image, the projected pixel points in the view A image and in the view B image have different coordinates $P_c(x_c, y_c)$ and $P_d(x_d, y_d)$ in their respective image coordinate systems.

In the visual SLAM, the x-axis direction is generally consistent, and the image plane overlaps during the acquisition of the view A image and the view B image of the target object. Based on the image coordinate system of view A [20], the image of view B is a simple translation relative to the image of view A, which is expressed by coordinates as $(e, 0, 0)$, e is generally called the baseline, and f is the imaging focal length. In the three-dimensional space, the coordinates of the image point of a point on the target object in the view A image and view B image plane are shown in Formulas (7) and (8), respectively:

$$\begin{cases} x_c = f\left(\dfrac{X}{Z}\right) \\ y_c = f\left(\dfrac{Y}{Z}\right) \end{cases} \tag{7}$$

$$\begin{cases} x_d = f\left(\dfrac{X-b}{Z}\right) \\ y_d = f\left(\dfrac{Y}{Z}\right) \end{cases} \tag{8}$$

In the process of visual SLAM mapping, the coordinate difference between the projection points of the same target point in the A view image and the B view image is defined as the parallax between two pixel points in the two images. The parallax can be expressed as:

$$\begin{aligned} d &= x_c - x_d \\ &= f\left(\frac{e}{Z}\right) \end{aligned} \tag{9}$$

This parallax usually changes continuously, and only when the surface is discontinuous will the step phenomenon occur. Therefore, after stereo matching of multi-view image features oriented to visual SLAM, error matching is eliminated by disparity constraint. The process is as follows:

(1) The initial matching feature point sets $M_A = \{P_{A1}, P_{A2}, \cdots, P_{An}\}$ and $M_B = \{P_{B1}, P_{B2}, \cdots, P_{Bn}\}$ are generated by matching the feature points of two images under different viewing angles, where P_{Ai} and P_{Bi} are matched, and the parallax set $d = \{d_1, d_2, \cdots, d_n\}$ is calculated using Formula (9).

(2) For each matching feature point P_{Ai} in the view angle image of target object A, it finds the three feature points P_{Aa}, P_{Ab}, P_{Ac} $(1 \le a, b, c \le n)$ with the closest Euclidean distance in M_A, and the corresponding parallax is d_a, d_b, and d_c, respectively. Since the parallax usually varies continuously, d_a, d_b, and d_c should be close to the size of d_i and less than a parallax threshold d_{th}, that is, Formula (10) is satisfied; Otherwise, P_{Ai} and P_{Bi} will be rejected as mismatches.

$$\begin{cases} |d_i - d_a| < d_{th} \\ |d_i - d_b| < d_{th} \\ |d_i - d_c| < d_{th} \end{cases} \tag{10}$$

In the process of visual SLAM mapping, the feature points that are considered to be rejected by mismatching are divided into three categories [21]. One is the feature points that are not matched correctly but are matched incorrectly, the second is the feature points that are matched incorrectly but have correct matching points, and the other is the feature points that are matched correctly but are rejected by mistake. Triangulation is used to match the rejected feature points, and the latter two types of feature points are matched and corrected. The matching process of triangulation correction is as follows:

(1) Triangulate the remaining matching feature point set after eliminating the error matching in the A-view image.

(2) Find the triangle to which each eliminated feature point P_{Af} belongs in the A-view image. The three vertices of the triangle are P_{AV1}, P_{AV2} and P_{AV3}. The three matching points in the B-view image are P_{BV1}, P_{BV2}, and P_{BV3}, respectively; Then, it traverses the feature points that do not match the P_{Af} counterparts in the triangle area of the B-view image with P_{BV1}, P_{BV2} and P_{BV3} as the vertices, and finds the feature point $P_{Bf'}$ that has the smallest difference between the sum of Euclidean distances P_{BV1}, P_{BV2}, and P_{BV3} and the sum of Euclidean distances P_{Af},P_{AV1}, P_{AV2}, and P_{AV3}. If Formula (15) is satisfied, it is considered that P_{Af} and $P_{Bf'}$ are matching feature point pairs.

$$Dist_{th} > \sum_{i=1}^{3} Dist\left(P_{Af}, P_{AVi}\right) - Dist\left(P_{Bf}, P_{BVi}\right) \tag{11}$$

The observation equation of the surrounding environment is constructed based on the world coordinates of the A-view image and the B-view image after correct matching, and the motion equation constructed by combining the inertial measurement unit data is used to update the map information and complete the binocular camera positioning.

These improvements have to do with (i) building a multi-scale pyramid to increase scale invariance and (ii) enabling orientation changes by using rotation-invariant rBRIEF descriptors. Disparity constraints would be used during matching to reduce the number of false matches. For a more step-by-step insight, the algorithm of the improved ORB integration is presented in Section 2.5.

2.3 Loopback Detection in Visual SLAM

Loopback detection is one of the important parts of stereo matching of multi-view image features for visual SLAM. Loopback detection, also known as closed-loop detection [22], mainly solves the problem of how to minimize the cumulative drift of a binocular camera's position estimation, make the state estimation more accurate, and improve the image matching effect of the visual odometer in visual SLAM. Loopback detection provides a good idea to solve the above problems. Firstly, we can imagine how humans perceive the environment and build maps in their minds. In a certain environment, people keep walking and then create a map of the environment from part to the whole in consciousness. However, as time goes on, the human brain will inevitably have forgetting and vague memory. If a person happens to return to the position he passed at a certain time, if he is sensitive to environmental memory and perception, he can remember that the position is a loop, so as to enhance his perception and judgment of the space position. For visual SLAM, the similarity between the current frame image and the past frame image can be estimated through image stereo feature matching. When the similarity is judged to meet the expectation, it is determined that a loop is detected. At present, the main loopback detection method is based on appearance similarity detection [23].

Now, the mainstream loopback detection is the method based on appearance. It has nothing to do with the front end and the back end. It is only based on the similarity of two images to determine whether there is a loopback relationship. This method removes the accumulated error and makes loop detection one of the core and independent modules of visual SLAM. In the appearance based loop detection algorithm, the key point is how to calculate the similarity between image frames. How to estimate the similarity between frames is just the difficulty of loopback detection. The most intuitive algorithm is the stereo matching of image features. The similarity is reflected by calculating the number of matches between images. Whether loopback is generated is determined based on the similarity analysis results between different frame images.

2.4 Back End Optimization in Visual SLAM

In short, the main purpose of back-end optimization is to deal with the random noise and cumulative error in the visual SLAM process, and improve the image matching effect of the visual odometer in the visual SLAM. Although it is ideally assumed in mathematical form that all input data in the process of stereo matching of multi-view image features are accurate, it is impossible to have such ideal binocular cameras and inertial measurement units in reality. The most accurate binocular cameras and inertial measurement units and other sensors must also have errors. The error will not disappear with the improvement of accuracy. Therefore, in addition to using the front-end visual odometer to estimate camera motion through stereo feature matching of multi-view images, the

algorithm should also solve how to minimize the impact of noise on state estimation. The problem to be considered in back-end optimization is how to estimate the state of the entire visual SLAM from the data rich in error noise and know how uncertain the state estimation is. This estimation process is called maximum a posteriori probability estimation. The state here includes the motion and map of the visual SLAM itself. In the classic visual SLAM framework, the front-end visual odometer provides the back-end with data to be optimized and the initial value of this data. As a key part of the entire visual SLAM, the back end is responsible for global optimization and deals with simple data. In recent years, the back-end optimization of SLAM has been gradually replaced by the emerging graph optimization theory.

The essential attribute of graph optimization is to express and solve conventional optimization problems in the form of graphs. Graph is a logical structure composed of vertices and edges. Graph theory is a mathematical theory for studying graphs. Suppose that the mathematical expression of a graph is $G = \{V, E\}$, where V is the set of vertices and E is the set of edges. The edge of a graph is a topological relationship between connected vertices. The most common edge is a simple relationship that only connects two vertices. When there are edges connecting more than two vertices in a graph, the graph can be called a hypergraph. The visual SLAM problem can be represented as a hypergraph.

A bifurcation-based framework is presented to analyze impacting, sticking, and grazing dynamics in a Fermi oscillator using hybrid dynamical system theory. Physical and mathematical models are developed, and numerical continuation with the COCO tool is employed to investigate system behavior under variations in excitation frequency, amplitude, and boundary gap. Path-following and eigenvalue analyses are used to characterize key bifurcation mechanisms governing the oscillator's nonlinear dynamics [24]. Assume that at time k, there is a position x_k of the binocular camera, and the binocular camera obtains an observation and obtains multi-view image data t_k. The observation equation of binocular camera is:

$$t_k = h(x_k) \tag{12}$$

In Formula (12), h represents the coefficient of the equation. However, in reality, Formula (12) cannot be exactly equal, and there is always an error, which is:

$$e_k = t_k - h(x_k) \tag{13}$$

Then this optimization problem takes x_k as the optimization variable and $\min_x F_k(x_k) = \|e_k\|$ as the objective function to finally solve the estimated value of x_k.

For a specific visual SLAM problem, the optimization variable x_k has parameterized variables. Here x is either the pose of the binocular camera or a space point. Correspondingly, there are many types of observation equations, such as:

(1) The mutual transformation of two positions and poses of the binocular camera;
(2) The binocular camera detects a space point at a certain pose, and calculates the distance and angle from the space point to the binocular camera body;
(3) The binocular camera detects a space point at a certain pose and obtains its plane pixel coordinates.

p represents the position of the binocular camera based on stereo matching of multi-view image features, l represents the landmark for detection, t is the observation value of binocular camera, and ∂ is the displacement. Where p and l are variables to be optimized, and t and ∂ are constraints of the optimization equation. In the optimization graph, vertices are used to represent the variables to be

optimized, and edges are used to represent the observation equation of binocular camera. Since the edges of a graph can connect more than one or two vertices, its mathematical form is expressed in a more general way, $t_k = h(x_{k1}, x_{k2}, \cdots)$, so that the number of vertices can be described is unlimited. The above three observation equations are expressed in the form of vertices and edges of graph theory as follows:

(1) The mutual transformation between two positions of binocular multi-view image: there is only one binary edge in the image, the vertex is two positions, the three-dimensional position is represented by T, and the equation of the edge is expressed as $T_1 = \Delta T \cdot T_2$;

(2) The upper model camera detects a point at a certain pose, and estimates the distance and angle from the point to the binocular camera: there is a binary edge in the figure, the vertex is a two-dimensional plane pose $[x, y, \theta]^T$ and a point $\left[\lambda_x, \lambda_y\right]^T$. The data observed by the sensor are distance r and angle b, so the observation equation is:

$$\begin{bmatrix} r \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} \\ \tan^{-1}\left(\dfrac{\lambda_y - y}{\lambda_x - x}\right) - \theta \end{bmatrix} \tag{14}$$

(3) The binocular camera observes a spatial point at a certain pose, obtains its pixel coordinates, and obtains its pixel coordinates: then there is a binary edge in the figure connected with two vertices, one of which is a three-dimensional pose T, the other is a spatial point $x = [x, y, z]^T$, and the observation data of the binocular camera is a pixel coordinate $t = [u, v]^T$. Obviously, the observation equation is:

$$t = \xi\,(Rx + \partial) \tag{15}$$

In Formula (15), matrix ξ represents the internal parameters of the binocular camera, and R and t represent the rotation and translation matrices, respectively.

Based on the graph optimization theory, the observation equation in the stereo matching process of multi-view image features for visual SLAM can be expressed, which can effectively deal with the random noise and cumulative error in the visual SLAM process, and enhance the image feature matching effect. The visual odometer, loopback detection and back-end optimization are organically combined to achieve stereo matching of multi-view image features for visual SLAM through the joint operation of the three.

2.5 Proposed Algorithm 1 Workflow

Here, a summary of the entire workflow of the stereo matching algorithm outlined in the proposed visual SLAM algorithm. The algorithm starts with two stereo image pairs as input and outputs the smoothed optimized camera pose as well as the reliably matched feature points.

The structured workflow illustrates the novelty of the method presented here (in particular, the proposed ORB-based feature extraction and disparity-constrained filtering and triangulation correction) and improves robustness and accuracy in visual SLAM.

In summary, Section 2.5 contains the entire workflow, i.e., the process from stereo-image acquisition until back-end graph optimization steps. Hence, a complete integrated approach is adopted where feature extraction, matching, mismatch rectification, and pose estimation are dealt with systematically. The subsequent section follows with the experimental evaluation of the proposed algorithm with results in feature extraction, feature point matching congruency, and positioning accuracy of the binocular vision robot.

Algorithm 1: Visual SLAM

Input: Binocular stereo images (View A, View B)

Output: Smoothed optimized camera pose with matched feature points

The following are the steps outlining the workflow of the stereo matching algorithm.

Step 1: Image acquisition and preprocessing

A binocular camera system is utilized to capture the stereo image pairs simultaneously at the two viewpoints. Populations of noise reduction techniques can be applied to enhance the quality of the acquired images.

Step 2: Feature extraction

An enhanced version of the ORB method is utilized for the detection of features. This method creates a multi-scale pyramid of images and constructs a rotation-invariant feature descriptor, termed rBRIEF, which affords scale and rotation-invariance over the standard ORB method.

Step 3: Feature matching

The extent of similarity of every one of the present feature descriptors is computed from View A and View B. In the case of floating-point descriptors, each descriptor is compared using its Euclidean distance; while if they are binary descriptors, the matching is performed using their Hamming distance.

Step 4: Mismatch rejection

Disparity constraints are applied to eliminate inconsistent matches. This step reduces false correspondences, especially in areas with repeated textures or smooth surfaces.

Step 5: Recover rejected matches

For those features that matched were incorrectly rejected, the camera pose can be recovered using triangulation that can be supported by geometric consistency in terms of matching.

Step 6: Pose and Motion Estimation

The other matched points are fused with IMU (Inertial Measurement Unit) data to estimate the camera's motion parameters (i.e., position and orientation), which will be used to produce a preliminary discrete map.

Step 7: Loopback Detection and Back-End Optimization

Loop closure detection is applied to find locations that have been revisited and to minimize drift. The global trajectory and landmark positions are optimized through a graph-based back-end optimization, improving overall accuracy.

End of Workflow

3 Result Analysis

In order to verify the application of the stereo matching of multi-view image feature for visual SLAM studied in this paper in the field of actual image feature matching, the benchmark database used by an image research institute of a university is taken as the research object. The images in the research object are all images collected by a model of binocular vision robot with visual SLAM, including various images under different shooting angles, such as elevation angle, top view angle, front angle, rear view, right view, left view, etc., and also including the images obtained by changing the images under different viewing angles, such as angle rotation, photometric change, scaling down, scaling up, viewing angle change, fuzzy enhancement, etc. The size of each image in the study object is 850×680 pixels. The proposed algorithm is used to match the internal test images of the research object, and the results are as follows.

3.1 Feature Point Extraction Results

The experimental dataset consists of stereo images captured by a binocular vision robot with a visual SLAM system. Each image is of 850 × 680 pixel resolution and contains all perspectives, such as front, rear, left, right, top, and bottom, along with views generated through rotation, scaling, photometric changes, and blurring. The data includes both indoor and outdoor environments, so that diversity is guaranteed. The proposed algorithm was run on a workstation with an Intel i7 processor and 16 GB of RAM, and the average processing time per frame was 32 ms. This makes sure that the approach is able to sustain real-time performance in relatively complex environments. It was calculated using a workstation powered by an Intel i7 CPU and 16 GB of RAM and, therefore, was able to sustain the real-time nature of SPEEDY, with an average processing time of 32 ms per frame. Nevertheless, in hardware with minimal resources, e.g., embedded systems or mobile devices, the algorithm can be optimized to ensure real-time processing speed, especially because of the complexity of the multi-scale pyramid construction and rBRIEF descriptors. This gives us reproducibility and also demonstrates the efficiency of the method. Figs. 2 and 3 showcase some examples of feature extraction and feature point matching on some samples, while Fig. 1 illustrates the architecture of the whole system. All images used in this study are from the famous stereo vision dataset maintained by the university research institute where the experiments took place.

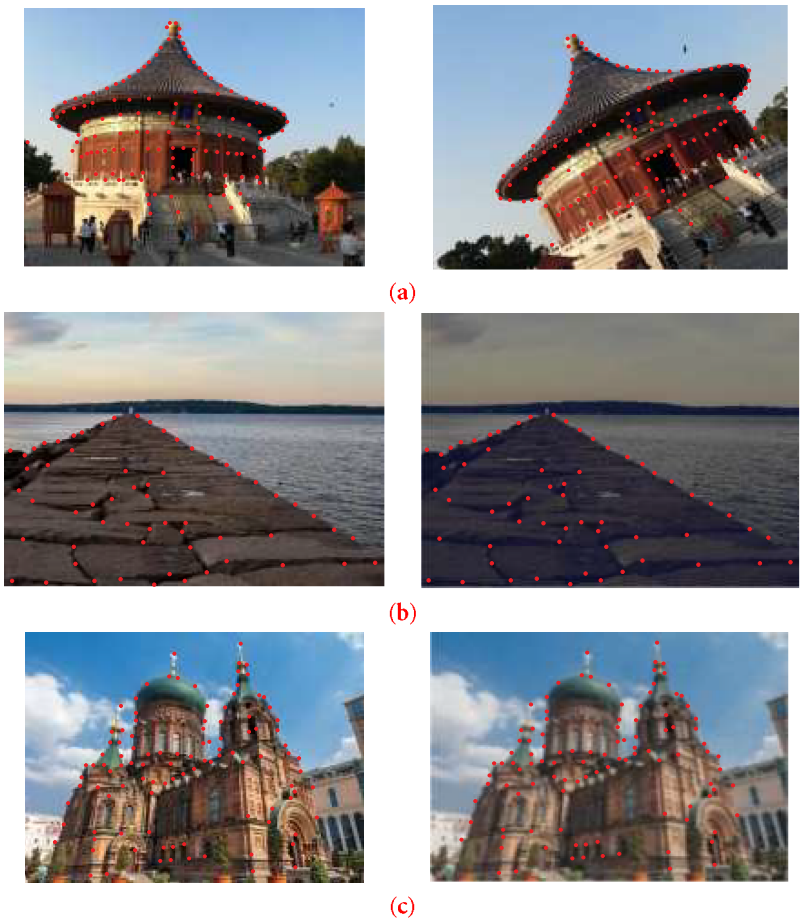

Figure 2: Feature point extraction results. (a) Feature point extraction in Traditional Chinese architecture; (b) Dock feature point extraction; (c) Church architecture feature point extraction

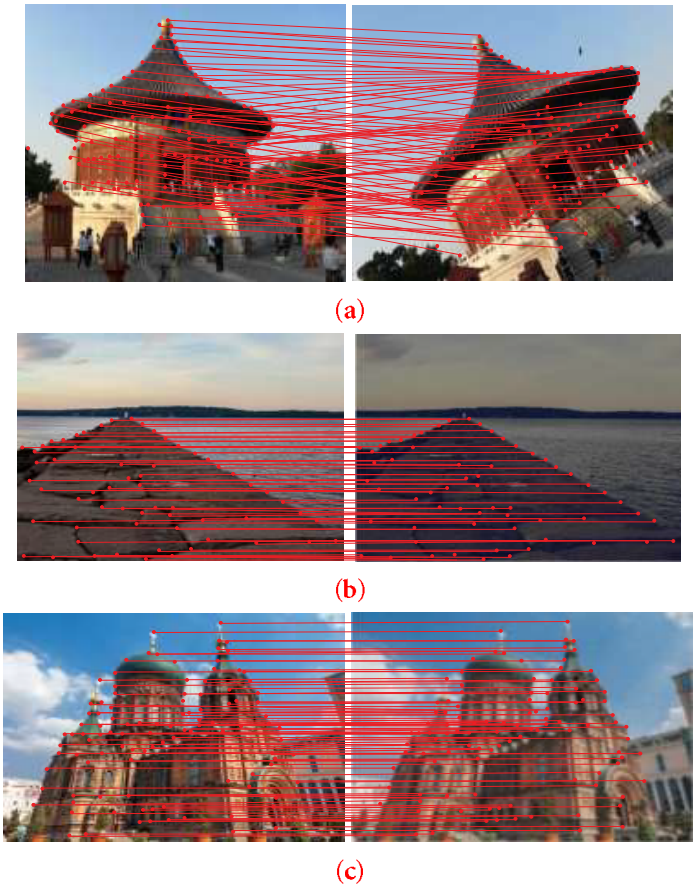

Figure 3: Matching results of characteristic points. (a) Matching feature points of traditional Chinese architecture; (b) Terminal feature point matching; (c) Matching of church architectural feature points

Three groups of images are randomly selected in the study object, named Image A, Image B, and Image C. The selected image is taken as the reference image, and the three reference images are changed through different types of change processing methods (angle rotation, photometric change, and fuzzy enhancement), and the results obtained are taken as the images to be matched. The proposed algorithm is used to extract feature points from the reference image and the image to be matched, and the results are shown in Fig. 2.

Analysis of Fig. 2 shows that the proposed algorithm can extract a large number of feature points in the reference image and the image to be matched. Most of these feature points are distributed in the area where the light and dark changes of the target image are significant. The above results fully show that the proposed algorithm can effectively extract the feature points in the image. The algorithm detects points robustly where they encounter strong light–dark variation.

Source: binocular vision robot dataset used in this study.

3.2 Feature Point Matching Results

The proposed algorithm is used to match the feature points in the selected reference image and the image to be matched. The feature point matching result is shown in Fig. 3.

According to Fig. 3, when the proposed algorithm is used to match feature points between the reference image and the image to be matched, most of the extracted feature points are effectively matched. Considering the presence of some mismatched points, the one-time correct matching rate of the feature points in the proposed algorithm is 98.9% after calculation, indicating that the algorithm effectively matches the target image. Matches are retained when correct and down-weighted through disparity-constrained filtering and triangulation when they are mismatches.

Source: binocular vision robot dataset used in this study.

3.3 Analysis on Matching Performance of Image Feature Points from Different Perspectives

Taking Image A as an example, the proposed algorithm is used to match feature points across images from different perspectives (up view, forward view, back view, left view), and the matching results are summarized in Table 2.

Table 2: Matching results of characteristic points

Matching combination	Reference image feature points/Piece	Number of feature points of image to be registered	Matching points/Piece	Mismatched quantity/Piece
Rear view and rear view	197	184	159	0
Back and front view	197	176	150	0
Rear view and left view	197	156	140	2
Looking up and back	182	112	135	1
Look up and look up	182	199	197	0
Looking up and forward	182	108	160	3
Bottom view and left view	182	187	124	0
Left vision and left vision	134	130	108	0
Left vision and front vision	134	109	191	1
Forward looking and forward looking	176	153	111	0

Table 2 shows that when the proposed algorithm is used for stereo matching of image features from different perspectives, the number of mismatched feature points is controlled within 3, indicating that the proposed algorithm has high stereo matching accuracy of feature points. This is mainly because the proposed algorithm uses disparity constraints to eliminate mismatches after feature point stereo matching, and also uses triangulation to find the correct matching points.

3.4 Application Analysis of Image Matching Results

Based on the stereo matching results of the proposed algorithm, the localization performance of the binocular vision robot is verified. The binocular vision robot is set to run freely for 80 times in the working environment, including 50 times when the natural light is sufficient, and 30 times when the natural light is insufficient, and the light is supplemented. At the same time, the influence of the establishment of the uncertainty model for the front end on the positioning accuracy is also measured in two cases, and the results are shown in Table 3.

Table 3: Positioning error analysis of binocular vision robot

Light conditions		Angle error/°		Position error/mm	
		Upper limit	Mean value	Upper limit	Mean value
Sufficient natural light	Before using this algorithm	1.26	0.78	60.13	35.39
	After using the algorithm in this paper	1.21	0.75	48.98	29.54
Light supplement required	Before using this algorithm	2.00	1.02	99.07	56.08
	After using the algorithm in this paper	1.90	0.96	72.83	47.97

Table 3 shows that the lighting environment will affect the positioning accuracy of the binocular vision robot based on the proposed algorithm, and the positioning accuracy is slightly lower under the condition of insufficient natural light than under the condition of sufficient natural light. The binocular vision robot based on the proposed algorithm can achieve high positioning accuracy in both cases. To sum up, the overall positioning error of the binocular vision robot based on the proposed algorithm is controlled within 100 mm, and the direction error is within 2°, which can meet the actual work requirements on site.

Fig. 4 presents the positioning error of the binocular vision robot before and after the application of the proposed algorithm under natural light and light-supplement conditions. The proposed method fits the baseline by achieving a significant reduction in the position error, ranging approximately between 20% and 25%. Also, the reduction in angular error is comparatively less, about 5%, yet consistent across different environments. These improvements underscore the algorithm's robustness against changes in illumination. Therefore, overall, the proposed method guarantees more stable localization accuracy suitable for real-world applications.

Fig. 5 presents feature point matching scenarios, evidencing the proposed algorithm's accuracy. Fig. 5a depicts matching before disparity-constrained filtering, under which several mismatches could be noticed due to noise or texture similarity. Fig. 5b, by contrast, depicts the corrected matches after disparity filtering and triangulation have been applied, where almost all mismatches are eliminated. This establishes the credibility of the algorithm in eliminating false correspondences. Thus, the illustrations provide intuitive evidence of the improvement over tabular results.

To better interpret the results, Fig. 4 depicts a positioning error comparison before and after applying the proposed algorithm under varying lighting conditions. It can be seen that the algorithm systematically reduces both angle and position errors by 15%–25% with respect to the baseline method.

Similarly, Fig. 5 portrays visualizations of feature point matching so that mismatches are almost completely removed after disparity-constrained filtering and triangulation correction. These figures make the performance improvements much intuitively supplementary to the tabular data.

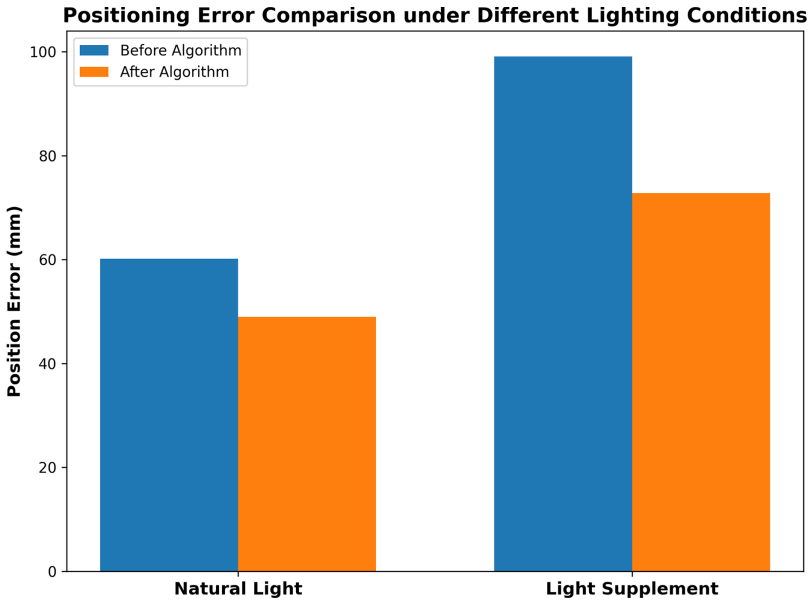

Figure 4: Positioning Error Comparison under different lighting condititons

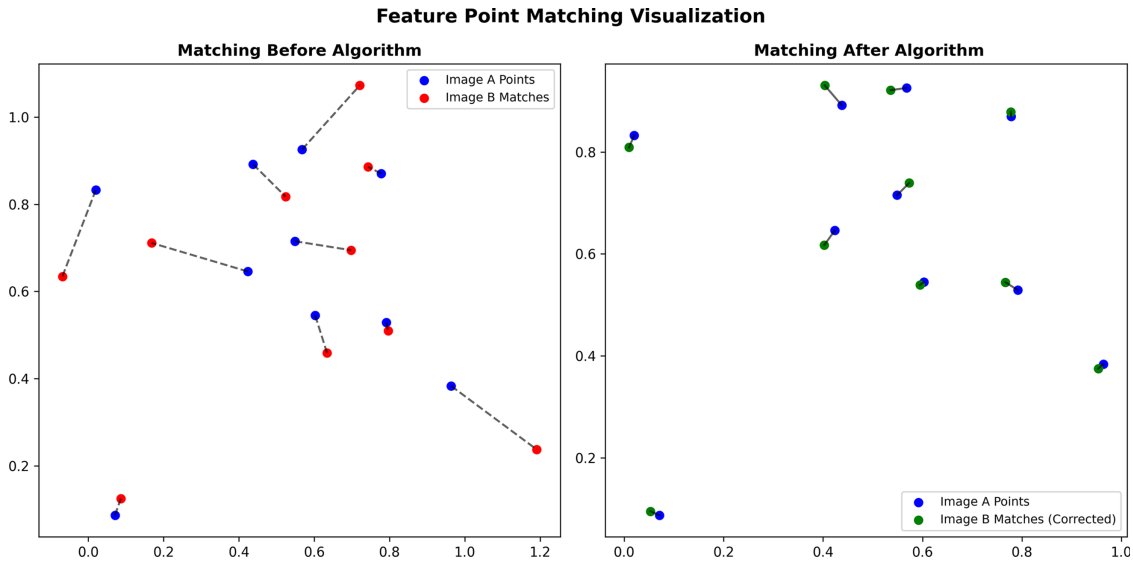

Figure 5: Feature point matching visualization. (a) Matching before algorithm; (b) Matching after algorithm

Fig. 6 represents the trajectory of the binocular vision robot with the ground truth and estimated paths prior to and posterior to the proposed approach. The estimated trajectory prior to correction (dashed line) shows that drift could be observed from the ground truth, especially at the loop closure. Post application of the enhanced ORB with disparity constraints in the back-end optimization, the estimated trajectory (solid green) almost follows the ground truth path, proving that the algorithm can reduce cumulative drift and thereby improve trajectory accuracy.

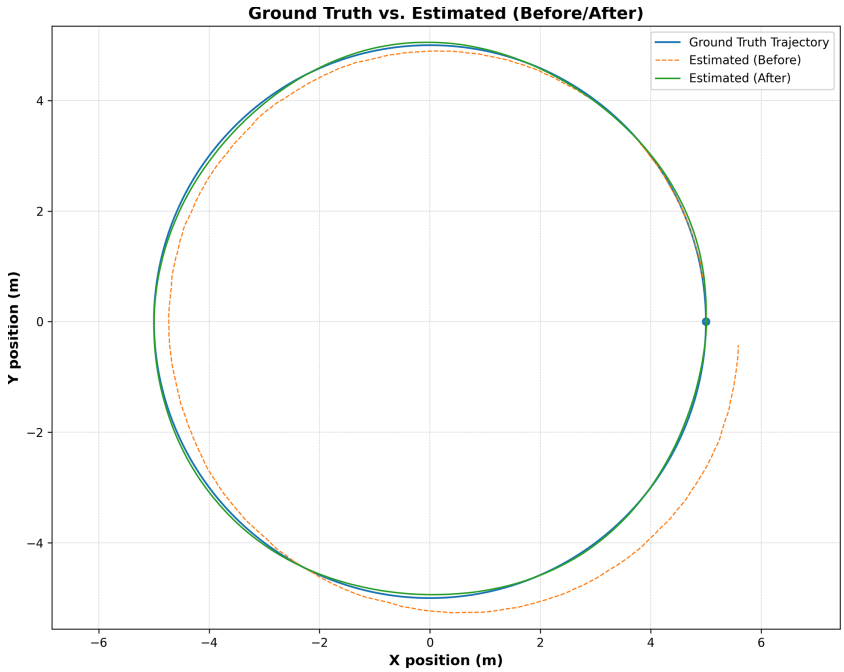

Figure 6: Ground truth vs. estimated (before/after)

In contrast to those studies in the literature that rely on conventional ORB- and SIFT-based approaches [6–8], this proposed method remains robust in difficult low-texture regions and accurate under photometric variations. It reveals that the integration of improved ORB descriptors with geometric constraints ushers a viable tradeoff between speed and accuracy.

Table 4 compares SLAM methods in their performance measures and average mismatches, position error, angle error, and runtime. The guiding principles presented in this paper include the SIFT, Standard ORB algorithms, Principal Direction SLAM, and AdaptSLAM. The Proposed Method has the highest performance in comparison to all of the mentioned methods, with lower mismatches and smaller angles, achieving the best performance on all counts of accuracy, precision, and accuracy. This theoretical approach demonstrates efficient processing times across different environments and is relatively competitive in runtime performance.

Table 4: Comparative performance of feature extraction and matching methods

Method	Avg. Mismatches (Per Sequence)	Position error (mm)	Angle error (°)	Avg. runtime (ms/frame)
SIFT [25]	5–7	70–90	2.0–2.3	55–60
Standard ORB [26]	8–10	95–120	2.1–2.5	20–25
Principal Direction SLAM [27]	6–8	65–85	1.8–2.2	∼28
AdaptSLAM: Edge-Assisted Adaptive SLAM [28]	4–6	55–75	1.5–2.0	∼30
Proposed Method	≤3	≤100 (48–73 typical)	1.2–1.9	∼32

4 Conclusion

This paper studies the stereo matching of multi-view image features for visual SLAM. Through the four processes of image acquisition and processing, feature point extraction, feature point matching, and pose and motion estimation in the visual odometer of visual SLAM, the feature stereo matching of multi-view images is realized. Through loop detection and back-end optimization in visual SLAM, the matching effect of the visual odometer image in visual SLAM is improved. Experimental results show that the proposed algorithm can effectively achieve image feature point extraction and matching, and improve the accuracy of image feature point matching.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by Henan Province 2022 Discipline and Major Construction Funding Project for private Colleges and Universities Communication Engineering Major (Education Office of Politics and Law (2022) No. 219) and the Ninth Batch of Henan Provincial Key Disciplines (Detection Technology and Automation Device) Construction Project of Henan Provincial Department of Education (JG [2018] No. 119); 2023 Key science and technology research projects of Henan province (No. 232102220071).

Author Contributions: Xiaowei Hu: writing—original draft and methodology. Dijie Xue: investigation and writing—review & editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

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

References

1. Dong Y, Wang S, Yue J, Chen C, He S, Wang H, et al. A novel texture-less object oriented visual SLAM system. IEEE Trans Intell Transport Syst. 2021;22(1):36–49. doi:10.1109/tits.2019.2952159.
2. Liu Y, Miura J. KMOP-vSLAM: dynamic visual SLAM for RGB-D cameras using K-means and OpenPose. In: Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII); 2021 Jan 11–14; Iwaki, Japan. p. 415–20. doi:10.1109/ieeeconf49454.2021.9382724.
3. Trujillo JC, Munguia R, Urzua S, Grau A. Cooperative visual-SLAM system for UAV-based target tracking in GPS-denied environments: a target-centric approach. Electronics. 2020;9(5):813. doi:10.3390/electronics9050813.
4. Gao H, Xu G, Zhang Z, Zhou W, Wu Q. A novel probability iterative closest point with normal vector algorithm for robust rail profile registration. Optik. 2021;243(11):166936. doi:10.1016/j.ijleo.2021.166936.
5. Li G, Yu L, Fei S. A deep-learning real-time visual SLAM system based on multi-task feature extraction network and self-supervised feature points. Measurement. 2021;168(4-5):108403. doi:10.1016/j.measurement.2020.108403.
6. Hao S, Wu Y, Ma X, He T, Wen H, Wang F. Visible and infrared image matching based on CycleGAN-SIFT. Opt Precis Eng. 2022;30(5):602–14. doi:10.37188/ope.20223005.0602.
7. Jia Z, Lu M, Hu J, Dong S, Han A, Su S. A novel dense descriptor based on structure tensor voting for multi-modal image matching. Chin J Aeronaut. 2020;33(9):2408–19. doi:10.1016/j.cja.2020.02.002.
8. Liao XF, Zhuang XC, Gong WT, Chen JJ. SIFT feature extraction and matching algorithm based on sequential minimal optimization. Comput Simul. 2019;36(2):219–23.

9. Zhang R, Wang Y, Li Z, Ding F, Wei C, Wu M. Online adaptive keypoint extraction for visual odometry across different scenes. IEEE Robot Autom Lett. 2025;10(7):7539–46. doi:10.1109/lra.2025.3575644.
10. Yang B, Xu S, Yin L, Liu C, Zheng W. Disparity estimation of stereo-endoscopic images using deep generative network. ICT Express. 2025;11(1):74–9. doi:10.1016/j.icte.2024.09.017.
11. Gong L, Gao B, Sun Y, Zhang W, Lin G, Zhang Z, et al. preciseSLAM: robust, real-time, LiDAR-inertial–ultrasonic tightly-coupled SLAM with ultraprecise positioning for plant factories. IEEE Trans Ind Inf. 2024;20(6):8818–27. doi:10.1109/tii.2024.3361092.
12. Sitaraman SR, Khalid HM. Robotics automation and adaptive motion planning: a hybrid approach using AutoNav, LIDAR-based SLAM, and DenseNet with leaky ReLU. J Trends Comput Sci Smart Technol. 2024;6(4):404–23. doi:10.36548/jtcsst.2024.4.006.
13. Tang L, Wu L, Fang Z, Li C. A non-convex ternary variational decomposition and its application for image denoising. IET Signal Process. 2022;16(3):248–66. doi:10.1049/sil2.12088.
14. Xu Z, Yu J, Yu C, Shen H, Wang Y, Yang H. CNN-based feature-point extraction for real-time visual SLAM on embedded FPGA. In: Proceedings of the 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM); 2020 May 3–6; Fayetteville, AR, USA. p. 33–7. doi:10.1109/fccm48280.2020.00014.
15. Wang T, Wang Z, Cao Y, Wang Y, Hu S. A multi-BRIEF-descriptor stereo matching algorithm for binocular visual sensing of fillet welds with indistinct features. J Manuf Process. 2021;66(2):636–50. doi:10.1016/j.jmapro.2021.04.040.
16. Ata W, Rukkanchanunt T, Chawachat J. Using fast intersection to improve SCAN algorithm. In: Proceedings of the 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON); 2021 May 19–22; Chiang Mai, Thailand. p. 123–6. doi:10.1109/ecti-con51831.2021.9454693.
17. Zuo F, Huang Y, Li Q, Su W. Infrared and visible image fusion using multi-scale pyramid network. Int J Wavelets Multiresolut Inf Process. 2022;20(5):2250019. doi:10.1142/s0219691322500199.
18. Mao YH, He ZZ, Ma Z, Bi RX, Wang ZP. Infrared-visible heterogeneous image matching based on intra-class transfer learning. J Xi'an Jiaotong Univ. 2020;54(1):49–55. doi:10.7652/xjtuxb202001009.
19. Pan Z, Wu X, Li Z. Central pixel selection strategy based on local gray-value distribution by using gradient information to enhance LBP for texture classification. Expert Syst Appl. 2019;120(9):319–34. doi:10.1016/j.eswa.2018.11.041.
20. Higuchi T, Goto D, Kobayashi N, Hayashi A. Image-based education using gaze movements of conductors via eye-tracking system. Jpn Railw Eng. 2019;59(3):16–8.
21. Wei M, Shao P, Wang W. Trifocal tensor based feature matching algorithm. J Beijing Inst Technol. 2020;106(4):53–7.
22. Xu J, Wang D, Liao M, Shen W. Research of cartographer graph optimization algorithm based on indoor mobile robot. J Phys Conf Ser. 2020;1651(1):012120. doi:10.1088/1742-6596/1651/1/012120.
23. Shin DW, Ho YS, Kim ES. Loop closure detection in simultaneous localization and mapping using descriptor from generative adversarial network. J Electron Imaging. 2019;28(1):013014. doi:10.1117/1.JEI.28.1.013014.
24. Ma W, Mapuranga T, Zhang J, Ding H, Chen J, Zhang X. Bifurcation and path-following analysis of periodic orbits of a Fermi oscillator model. Int J Bifurcation Chaos. 2022;32(12):2250179. doi:10.1142/s0218127422501796.
25. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91–110. doi:10.1023/B:VISI.0000029664.99615.94.
26. 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; 2011 Nov 6–13; Barcelona, Spain. p. 2564–71. doi:10.1109/iccv.2011.6126544.

27. 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(19):9124. doi:10.3390/app14199124.
28. Chen Y, Inaltekin H, Gorlatova M. AdaptSLAM: edge-assisted adaptive SLAM with resource constraints via uncertainty minimization. In: Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications; 2023 May 17–20; New York, NY, USA. p. 9124. doi:10.1109/infocom53939.2023.10229009.