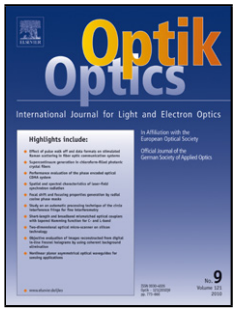
Journal Pre-proof



An Improved Visual SLAM Based on Affine Transformation for ORB Feature Extraction

Lecai Cai, Yuling Ye, Xiang Gao, Zhong Li, Chaoyang Zhang

PII: S0030-4026(20)31257-2

DOI: https://doi.org/10.1016/j.ijleo.2020.165421

Reference: IJLEO 165421

To appear in: *Optik*

Received Date: 15 February 2020

Revised Date: 10 July 2020

Accepted Date: 10 August 2020

Please cite this article as: Cai L, Ye Y, Gao X, Li Z, Zhang C, An Improved Visual SLAM Based on Affine Transformation for ORB Feature Extraction, *Optik* (2020), doi: https://doi.org/10.1016/j.ijleo.2020.165421

An Improved Visual SLAM Based on Affine Transformation for ORB Feature Extraction

Lecai Cai[1], Yuling Ye[2*], Xiang Gao[3], Zhong Li[4] and Chaoyang Zhang[5]
[1]Sanjiang Institute of Artificial Intelligence and Robotics, Yibin University, Yibin 644000, China
[2]School of Automation and Information Engineeringg, Sichuan University of Science & Engineering, Zigong 643000, China
[3]School of Mechanical Engineering, Sichuan University of Science & Engineer, Zigong 643000, China
[4]School of Computer and Information Engineering, Yibin University, Yibin 644000, China
[5]School of Physics and Electronic Engineering, Yibin University, Yibin 644000, China

*Corresponding author: Yuling Ye (e-mail: 821174742@qq.com).

This work is supported by Research on Intelligent Brewing Device and Its Key Technology Based on Solid State Fermentation(2019YFN0104) and Research on Substation Inspection Robot Based on Multi-sensor Fusion (2016ZGY021)

Abstract

Aiming at the problems of the existing robot vision SLAM(Simultaneous Localization and Mapping), such as the small number of feature point extraction and the easy loss of keyframes, which leads to the problem of trajectory deviation, many existing visual SLAM methods based on keyframes only propose a holistic system solution in the scheme, no detailed research is carried out on the feature extraction of the front-end visual odometry. In this paper, an affine transformation based ORB feature extraction method(Affine-ORB) is used and applied to existing robot vision SLAM methods, and an improved visual SLAM method is proposed. In the proposed SLAM, we first use the BRISK method for feature point description; Secondly, the mathematical method of affine transformation is introduced into the ORB feature extraction; Finally, the sample is normalized and the image is restored. By requiring a handheld camera to take the vision SLAM experiment, it is judged that the keyframe loss rate of the proposed algorithm is significantly reduced. Through evaluation experiments with TUM, KITTI and EUROC datasets, the keyframe extraction effect and positioning accuracy of the SLAM algorithm set out in the present paper are compared with PTAM, LSD-SLAM and ORB-SLAM, respectively. The frame loss rate of feature extraction SLAM based on affine transformation has decreased from 0.5% to 0.2%, and the root mean square error (RMSE) of the running trajectory has been drastically reduced. That means the key frame extraction speed is faster, the keyframe loss rate is lower at the same moving speed, and the positioning accuracy is higher

Keywords Visual SLAM, Affine-ORB, Feature extraction, Positioning

Introduction

When a robot device enters a small area, it needs to obtain location information and plan the movement path to solve the minor area positioning problem. The method of Simultaneous localization and mapping (SLAM) [1-4] performs well in solving this problem. In the unknown unstructured or small-scale positioning and mapping, visual SLAM combined with laser information or [5] inertial conduction [6] has become a popular research point. As the cost and the volume of the camera reducing gradually, the advantages of visual SLAM have attracted widespread attention and exploration by researchers. Previous researchers usually

discussed general problems of vision SLAM, and even only described them comprehensively from the system combining laser and vision, but the research lakes the distinctive analysis of visual SLAM. For example, Cadena *et.al.*[7] carried out some comprehensive research and exploration on SLAM, but it did not conduct in depth research about keyframes. A more definitive visual odometry SLAM system was proposed in Literature [8] , but it did not discuss the defects and improvement points of the visual odomety in details.

With the improvement in our real-time requirements of robotic system, in the process of positioning and mapping of mobile robots, if the camera moves too fast, it will cause the feature points that can be matched between adjacent frames to decrease sharply and lose keyframes. It may leads to the problem of the pose tracking failed, which also causes a large trajectory drift directly. Therefore, we will start with the graphical features of the SLAM front end, try to improve the extraction of keyframes and optimize the correlation, and propose an improved visual SLAM, and achieve the goal of improving the accuracy of mapping and positioning. Aiming at the problems of weak performance of real-time, easy losing of keyframes, and huge deviation of trajectory in the existing methods of visual SLAM, this paper proposes a affine method based on the principle of ORB feature extraction (Affine-ORB). The effects of extracting and matching directed FAST [9] and Rotated Brief [10] are more obvious. Subsequently, on this basis, a visual SLAM method based on Affine-ORB is proposed to achieve directional matching, speed up the extraction of keyframes, reducing the frame loss rate, and significantly improving the accuracy of mapping.

The main contributions of this article are as follows: Firstly, we discuss the model of affine transformation, and propose an improved visual SLAM algorithm based on the affine transformation of ORB (Affine-ORB) feature extraction through the datasets and actual tests. Under the condition of the constant moving speed, this algorithm takes the advantage of more keyframes and lower keyframe losing rate. Secondly, we use the evaluation tool to compare the horizontal and vertical coordinates of keyframes, the angle offset, the trend of the motion trajectory and the positioning mapping trajectory. It shows the advantages of our algorithm in the accuracy of location mapping.

In the rest of this article, we will first discuss related work, then we describe the proposed Affine-ORB feature extraction method based on an affine transformation in detail. Based on this, experiments are conducted by handheld devices and definitive datasets, and the results are compared with several exist methods of visual SLAM. Finally, this issue is summarized in the last section.

Related Work

Visual SLAM

As an important application part of machine vision, robot vision SLAM has developed rapidly in the past ten years. Early monocular vision SLAM is widely promoted based on filtering methods, such as FAST-SLAM [11], it's an early version of the nonlinear error model and large calculations limited its practical application. In 2007, the monocular vision mapping method proposed by Davison *et al.*[12], which primarily used the idea of probable deviation to solve the purpose of positioning mapping. Subsequently, semi-direct monocular vision(SVO) SLAM was proposed by Forster *et al.*[13], it used the procedure of tracking optical flow to directly combine image feature points. Vins-Mono[14] utilized the tight coupling method to recover the scale by monocular combined with the Inertial Measurement Unit(IMU) to obtain feature points. Although these methods consume fewer calculations during the tracking and matching process, they are insensitive to keyframes, and they lack the development potential for the increasing requirements of SLAM for real-time and reliability of mobile robots. Especially in the extraction of keyframes, if the algorithm cannot meet the real-time requirements of the system, the feature points will not be defined and the keyframe extraction will fail. The accuracy of the mapping will be lowered, and the boundary adjustment and loop closing will be biased because of the lacking of keyframes.

In recent years, direct visual SLAM based on feature extraction have gradually become the mainstream methods. Compared with the traditional filtering-based method, it has better real-time performance and positioning accuracy. Parallel Tracking and Mapping (PTAM) [15] is an early visual feature SLAM algorithm based on non-linear back-end optimization. After extracting the optimal 3D model and camera parameters from optical reconstruction, combining each feature point and making optimal adjustments to the camera pose and feature point spatial position, it implements Bundle Adjustment (BA) [16] with a limited real-time performance. Since then, most feature-based methods have been enhanced, one of which is ORB-SLAM [17-19]. ORB-SLAM has been optimized and improved based on its predecessors, and its real-time performance has been significantly improved. It establishes a common visibility chart and a minimum spanning tree. These graphics are used to identify key frames in order to track and map the running of tasks while allowing work in complex environments, and pose optimization can be performed when the loop is closed. The above-mentioned methods of visual SLAM, which are simultaneously tracked and mapped, rely on feature points. Therefore, when the feature points are not obvious, it is difficult to obtain them, which may cause tracking failure. In 2016, RK-SLAM [20] uses a method based on RANSAC [21] to extract local homomorphism data between the current frame and another key frame to solve the problem of unstable feature extraction caused by the camera when it experiences fast motion. A

3D reconstruction SLAM proposed by Greene *et al.* based on keyframes[22], which uses a large number of key frames to obtain positioning environmental information. Although the accuracy is high, the device has high requirements on the GPU. The DATMO-SLAM method proposed by Petrovskaya *et al.* [23] is mainly for the detection of dynamic feature points. Despite the fact that the feature points are easy to extract, the calculation is relatively complicated and the hardware cost is high.

In some direct visual SLAM methods, the pose of the camera is directly estimated from the intensity value of the image. For example, SVO-SLAM[24] directly aligns the images of sparse models, but it is not completely direct in terms of pose estimation and back-end adjustment. DSO-SLAM[25] uses a precise method for pose estimation, and uses a displacement transformation method of keyframe to adjust the pose after 4-6 keyframes. However, it lacks the steps of loop closing detection, so it gradually causes larger cumulative errors. LSD-SLAM [26] estimates the pose of the robot device by building a semi-dense point cloud depth image to match the next frame, but the method is sensitive to light changes, so it is easy to lose keyframes in environments where the light intensity changes significantly.

Feature Extraction

Image feature extraction is a key step of visual SLAM front-end visual odometry. Early invariant feature extraction such as SIFT [27] and the accelerated version of SIFT [28] are not easily affected by changes in ambient light, camera rotation, and image size and scale, so they have some robustness. However, if the robot device moves too fast or the field of view changes rapidly at the corners, it is easy to cause keyframes to be lost. In the feature extraction of affine transformation, MSER [29] uses different gray thresholds to minimize the image, and uses the area threshold method to detect gray changes. As the threshold value increases, the area of the connected part changes relatively small. Then different threshold values are selected to obtain the connected components. Finally, the appropriate threshold value is determined to obtain the final stationary region, but this method does not have complete scale invariance. The Harris-Affine [30] algorithm obtains analogous regions of an image under different affine changes. The detected feature points are independent of the image transformation and do not change with the image transformation, but the feature description will change. If severe affine distortion occurs, Harris-Affine's matching effect will be significantly worse. In 2009, the ASIFT method proposed by Morel *et al.* [31], which simulated the camera's viewing angle change and combined the SIFT algorithm for feature matching, to a certain extent, eliminating image distortion caused by changes in the optical axis direction.

The ORB algorithm [32] was proposed by Rubele *et al.* in 2011. It can be used to detect local key points in an image, and has better performance and lower computational load. The ORB feature extraction consists of two parts: a key point and a descriptor. The two parts constitute the fundamental elements of the ORB feature extraction, and the key point is used to detect the change of gray value. Therefore, the distinctive feature of the images extracted by the ORB algorithm is that they have local feature invariance. Besides, the key points and descriptors of the ORB algorithm constitute the quintessential elements of the ORB feature extraction, and the FAST key point detection and the BRIEF feature descriptor are combined to improve. For the shortcomings of the FAST algorithm, scale and rotation descriptions are added, and the principal directions of the feature points are calculated. A rotation invariance is added to the BRIEF method, and a greedy search [33] is performed to solve the problem of substantial correlation between feature descriptors at the meantime.

Aiming at the problems that the robot moves too fast and the feature extraction is easy to lose, this article will conduct research on the front-end visual odometry [34-36]. Based on ORB features, the feature points are first described using the method of BRISK[37], and then combined with the principle of affine invariance, a visual SLAM method based on affine transformation is proposed. We will use datasets experiments and handheld cameras to simulate robot pose differences, and prove the advantages of the proposed algorithm, such as robust feature extraction and accurate positioning.

Proposed Visual SLAM on Affine-ORB Method

Feature Point Extraction Descriptor of Visual SLAM

First, we use BRISK descriptors to describe feature point extraction. BRISK feature description is a binary-based feature descriptor with constant constancy and direction of rotation by using the domain sampling model. BRISK has been improved by Ji Z *et al.* [38] based the method that the BRIEF descriptor selects two pixels for binary comparison randomly, and its anti-noise capability has become stronger. The BRISK feature description is given in Figure 1. The blue circle in the center of the image is the main point.The surrounding N points and the corresponding surrounding circle, the small blue circle is the sampling point,

and the red circle represents the filtering radius. The filtering radius is proportionate to the Gaussian variance. Finally, after conducting by Gaussian smoothing, N sampling points are generated.

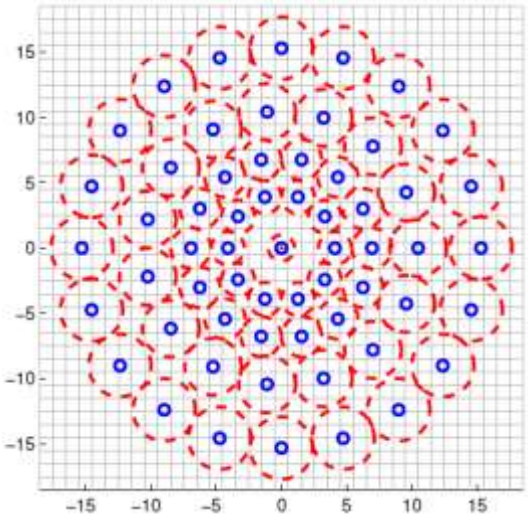


Figure 1. BRISK method

Then select N points in pairs to form corresponding point pairs, which are $N(N-1)/2$ point pairs, expressed as (P_i, P_j), the smooth pixel value of each point is $J(P_i, P_j)$, $J(P_j, P_i)$ respectively. Then calculate the local gradient size (Equation 1):

$$G(p_i, p_j) = (p_j - p_i) \bullet \frac{J(p_i, q_i) - J(p_j, q_j)}{\left\| p_i - p_j \right\|} \quad (1)$$

The total set of sampling points is showed in Equation 2:

$$A = \{(p_i, p_j) \in R^2 \times R^2, i < N^i < i^i, j \in N\} \quad (2)$$

The set of short-distance point pairs and the set of long-distance point pairs are shown in Equation 3 and 4, respectively:

$$D = \{(p_i, p_j) \in A, \left\| p_j - p_i \right\| < q_{\max}\} \subseteq A \quad (3)$$

$$C = \{(p_i, p_j) \in A, \left\| p_j - p_i \right\| < q_{\min}\} \subseteq A \quad (4)$$

Then calculate the main direction of the feature point (Equation 5):

$$t = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \bullet \frac{1}{L} \sum\nolimits_{p_i, p_j \in L} G(p_i, p_j) \quad (5)$$

Because the sampling area needs to be rotated according to the main direction of the feature point, when calculating the feature descriptor, the rotation angle is shown in Equation 6:

$$\theta = \arctan 2(t_y, t_x) \quad (6)$$

Among them, the ratio of short-distance point pairs is the value of each bit of the descriptor (Equation 7):

$$k \begin{cases} 1, I(P_j^\theta, q_j) > I(P_i^\theta, q_i) \\ 0, else \end{cases}, \forall (P_i^\theta, P_j^\theta) \in D \quad (7)$$

Finally, a binary descriptor is generated, and the Hamming distance [39] is selected for matching.

The Basic Model of Affine Transformation

The linear transformation in the plane coordinate system is known as affine transformation. Since the transformed image will not be distorted whether it is a straight line, an arc or other types of curves, the affine transformation effectively resolves the distortion of the two-dimensional plane [40]. Although the coordinate order of the points on the line will remain unchanged, the

angle of each point will change after the transformation. Therefore, further secondary transformations such as translation, zoom, tilt, rotation, and flip are needed.

Due to the distortion of the camera angle of view and the loss of keyframes for loop closing detection, there are drifts and deviations in the construction. Depending on the basic idea of the ASIFT algorithm, a full affine invariant method of the image is proposed. At the same time, the description of the angle change in the vertical direction and the horizontal direction is defined. The change rate of the image generated by the camera is estimated to reduce the construction deviation. This article uses the thinking method of ASIFT to improve on the basis of ORB feature extraction to obtain the keyframes of robot mapping and tracking in SLAM.

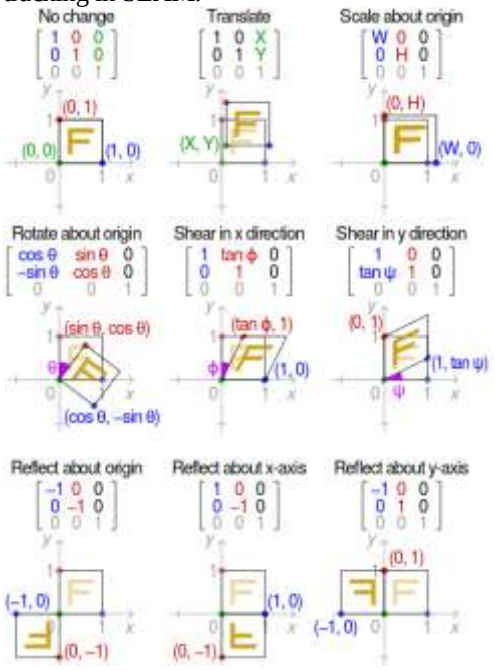


Figure 2. Basic Principles of Affine Transformation

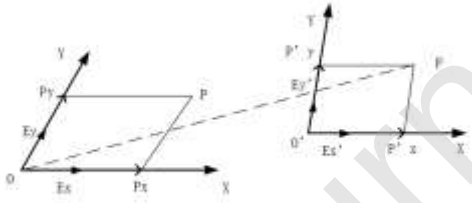


Figure 3.Relationship of points in vector of affine transformation

As shown in Figure 3, the number of points after the affine transformation is shown in Equation 9:

$$\begin{aligned}\overrightarrow{OP'} &= \overrightarrow{OO'} + \overrightarrow{O'P'} = (M_{13}e_1 + M_{23}e_2) + x(M_{11}e_1 + M_{21}e_2) + y(M_{12}e_1 + M_{22}e_2) \\ &= x'e_1 + y'e_2\end{aligned} \tag{9}$$

According to Equation 9, Equation 10 could be further obtained :

$$\begin{cases} x' = M_{11}x + M_{12}y + M_{13} \\ y' = M_{21}x + M_{22}y + M_{23} \end{cases} \tag{10}$$

Among them, $M_{11}, M_{12}, M_{13}, M_{21}, M_{22}, M_{23}, x, y$ are arbitrary integers, and in general, and are not parallel, so Equation 10 can be transformed into equation 11:

$$V(x,y) = v(ax+by+e, cx+dy+f) \qquad (11)$$

Then the affine transformation matrix could be get in Equation 12:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \ (|A| > 0) \qquad (12)$$

Matrix A undergoes singular value decomposition to obtain in Equation 13:

$$A = \mu \begin{bmatrix} \cos\eta & -\sin\eta \\ \sin\eta & \cos\eta \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\xi & -\sin\xi \\ \sin\xi & \cos\xi \end{bmatrix} = \mu R(\eta) T_t R(\xi) \quad (13)$$

Among them, λ is the focal length of the camera, $R(\eta)$ and $R(\xi)$ are the description of the rotation matrix, which is the tilt angle. For ORB-based keyframe extraction, our matrix of the affine camera model is shown in Equation 14:

$$A = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \left|\frac{1}{\cos\beta}\right| & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (14)$$

Among them, α and β correspond to the horizontal and vertical directions of the optical axis that causes the image rotation and vertical change angle, respectively, θ corresponds to the camera's spin angle around the optical axis. The image change set simulated by the Affine-ORB algorithm is obtained by changing α and β. That means simulating different perspectives through the two directions of horizontal and vertical.

In order to solve the horizontal and vertical distortion of the affine transformation, when a keyframe is captured by the camera, the observation angle of the image is measured first, and the sample normalization processing is performed next. Then image sampling is performed, and the slope $t = 1/\cos\beta$ is introduced to indicate that the original image $u(x,y)$ is deformed in the horizontal direction. In order to simulate the affine transformation image set uniformly, the horizontal direction is determined according to the proportional series in Equation 15 (The horizontal sampling time interval ratio of the image is constant $\sqrt{2}$):

$$m = 1, a, a^2, \cdots, a^n (a > 1) \qquad (15)$$

At the same time, the value in the vertical direction is taken as the arithmetic sequence in Equation 16(The optimal value of b is a known constant: b = 72 °):

$$n = 0, b/t, 2b/t, \cdots, nb/t (nb/t < 180^\circ) \qquad (16)$$

The sampling method is shown in Figure 4:

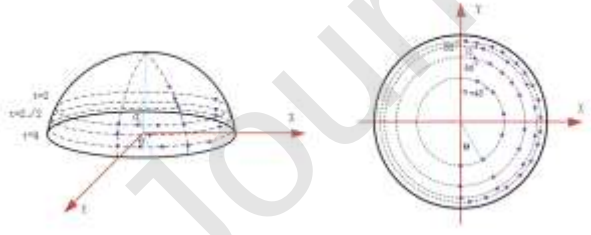


Figure 4. Analog sampling(The purple dots are sample points)

Object Matching and Image Correction

Through the affine transformation, image target matching and correction are also required to meet translation, rotation, and scale invariance. Therefore, after obtaining the sampling information, the camera's initial pose and the more accurate 3D points (Map-point) will be determined, and the feature points in the extracted scene will be tracked in real time, and the camera pose and keyframes will output in real time. Two models are calculated in parallel according to whether the image is a planar scene: the base matrix H and the independent matrix L. When the image is a non-planar scene, choose to calculate the base matrix H, which should) meet the following conditions in Equation 17:

$$x_c^T H x_r = 0 \quad (17)$$

Among them, x_c and x_r are the projection coordinates of any point x in space on the two images, respectively. When the camera is rotated, the polar constraints of the two views do not hold, and the degree of freedom of the basic matrix drops to zero. In this case, an independent matrix L needs to be calculated. L must satisfy the following relationship as Equation 18 shows:

$$L x_r = x_c \quad (18)$$

At the same time, the independent matrix also describes the transformation relationship between the two images in Equation 19:

$$Z' = LZ \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (19)$$

Among them, (x', y') is a point in the reference image, and (x, y) is a corresponding point to (x', y') in the image to be matched. Then calculate the optimal matrix model consists by Equation 20 and 21:

$$S = \sum_i \{ p[d_{cr}^2(x_c^i, x_r^i, K) + p(d_{rc}^2(x_c^i, x_r^i, K)] \} \quad (20)$$

$$P(d^2) = \begin{cases} t - d^2, d^2 > T_m \\ 0, d^2 < T_m \end{cases}, (3.84 < T_m < 5.99) \quad (21)$$

In the optimal matrix model, $d_{cr}^2(x_c^i, x_r^i, K)$ is the projection error of the matching point pair (x_c^i, x_r^i) after the optimal matrix transformation on the current frame, and $d_{cr}^2(x_c^i, x_r^i, K)$ is the symmetric error. T_m can effectively filter invalid values. The more accurate the standard value K is, the smaller the reproduction error of all matching point pairs is, and the larger the value of S is. The maximum values of the basic matrix and the independent matrix are denoted as S_H and S_L respectively. For scenes with small changes in the field of view, select the independent matrix L to restore the image; for scenes with relatively large changes in the field of view, choose to use the base matrix H to restore the image. The judgment method of model selection is as following in Equation 22:

$$R_L = \frac{S_L}{S_L + S_H} \quad (22)$$

If $R_L > 0.45$, the independent matrix is selected, otherwise the base matrix is used.

The Implementation of Affine-ORB

The ORB feature extraction based on the Affine method, combined with the existing BRISK feature descriptors, can quickly respond to the effective acquisition of key frames. The specific steps can be summarized as follows:

Step1: Preprocess the image using BRISK feature descriptors, generate binary descriptors and use Hamming distance for matching.

Step2: According to the thinking method of ASIFT and sampling rules proposed by BRISK, the parameters in the horizontal direction and the vertical direction are recombined and simulated by affine transformation.

Step3: For the two sets of images that have been simulated, normalize the samples to solve the distortion problem.

Step4: After the distortion is eliminated, use the basic matrix or independent matrix to achieve image target matching and image correction.

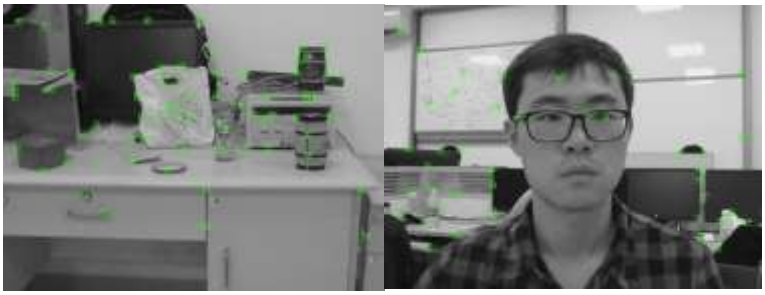
Evolution

In this section, we compare the differences between PTAM, LSD-SLAM, ORB-SLAM and our algorithm(Proposed) in positioning and mapping. Our experiments are operated based on the Linux system(ubantu16.04). The configuration of the laptop is Intel Corei5-3700MQ (quad-core@3.60GHz), and the memory size is 8 GB. In addition to verifying the algorithm of this article by handheld devices, the positioning trajectory and root mean square error of several visual SLAM algorithms are also compared.

The Experiment of Affine-ORB Feature Extraction

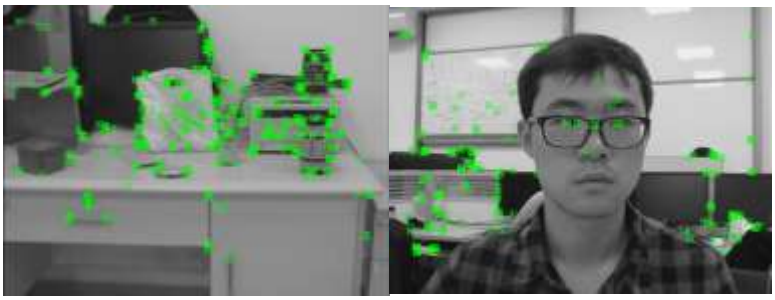
First, an ORB feature extraction experiment is operated. The number of ORB feature points determines the ability to acquire keyframes directly. Compared with the classic ORB-SLAM, the real-time image collected by the RGB-D camera clearly shows

that the number of ORB feature points extracted by the algorithm proposed in this paper is larger, and the feature point acquisition ability is stronger in the same keyframes (Figure 5-6).



(a) **(b)**

Figure 5. Feature point extraction experiment based on ORB-SLAM algorithm.

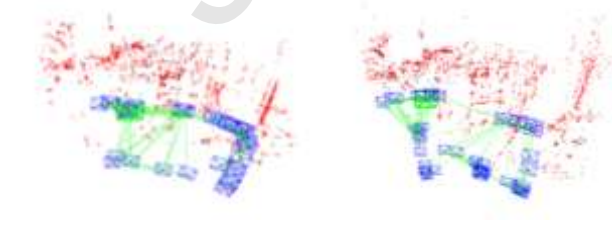


(a) **(b)**

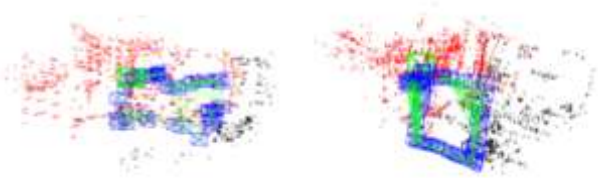
Figure 6. Feature point extraction experiment based on the method proposed in this paper.

Next, in order to reflect the advantages of the improved visual SLAM algorithm proposed in this paper for extracting keyframes visually, we take up an estimation experiment by the handheld RGB-D camera. The optical field range of the feature extraction is shown in Figure7. The camera follows a path shown in the figure below and takes the upper left corner of the visual field as a starting point to move a rectangular loop.

Figure 7. Feature extraction field of view and handheld camera trajectory



(a) **(b)**



(c) (d)

Figure 8. Keyframe-based trajectory map obtains from camera movement.(a)shows the result generated by PTAM, (b)reflects the LSD-SLAM, (c)reflects the ORB-SLAM, and (d)reflects the Improved algorithm proposed in this paper

After the camera moves, according to Figure 8, red and black points are extracted feature points, and the blue is keyframes. Under the same conditions, it takes about 22s to move the camera into a rectangular loop, while keeping the speed as uniform as possible. Compared with three methods of PTAM, LSD-SLAM and ORB-SLAM, it can conclude from the experiments that the improved visual SLAM algorithm based on Affine-ORB loses fewer keyframes, and gets a more complete trajectory. The other three methods of SLAM show obvious keyframe dropping, which is due to the slow speed response of feature extraction. At the same time, after ten experiments, it gives the average number of keyframes captured by the four algorithms within about 22 seconds of the camera movement. Due to the large random error of the experimental method relying on the movement of the handheld device, in order to maintain the accuracy as much as possible, the feature extraction experiments of each algorithm are performed ten times, and the average number of keyframe acquisition and the total number of sequences are adopted. It gives the calculation formula of the frame loss rate and the comparison of the frame loss rate shown in Equation 22 and Table 1, respectively. It can be concluded that the Affinate-ORB algorithm obtains more keyframes in the same time, which means that the loss rate of keyframes extracting is lower.

$$\text{The frame loss rate} = \frac{\text{The number of Track lost frames}}{\text{The total number of sequences}} \times 100\% \quad (22)$$

Table 1 Frame lose ratio comparison (Hand held Camera Experiment)

Algorithm	Total number of sequences(Average)	Number of Key Frames(Average)	Key Frame loss rate(%)
PTAM	522	39	92.53
LSD-SLAM	526	40	92.39
ORB-SLAM	523	58	88.91
Affinate-ORB	521	64	87.72

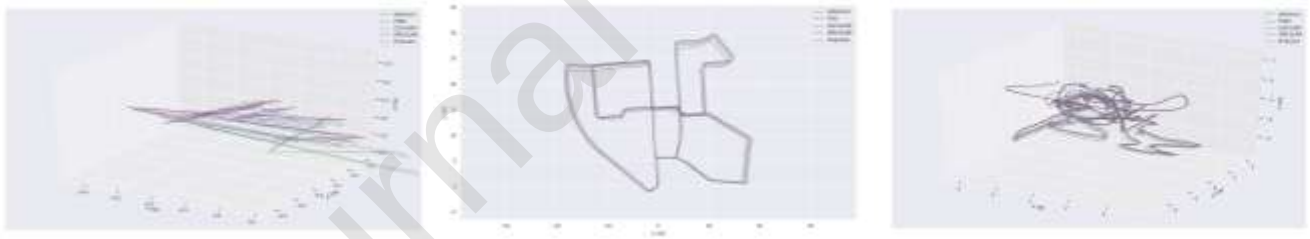
Next, we take up experiments based on the Dataset TUM, which keep the sequences of Handheld SLAM. The Dataset TUM is suitable for evaluating the accuracy of camera positioning to evaluate the accuracy of camera positioning because it provides accurate ground truth of sequences and contains seven sequences recorded by RGB-D camera. We only use handheld sequences (SLDS) and robot sequences (Sock) in seven sequences. The sequences of Handheld SLAM are recorded by handheld cameras generally, so it has complex and unstable trajectories. And the sequences of Robot SLAM are recorded by real robots, so it has stable and simple trajectories of movement. Each experiment is processed five times and then averaged. Table 2 shows the size of the keyframe losing rate of each method. From the data in the table, it's obvious that the improved algorithm proposed in this paper(referred to as "Proposed") based on the dataset (*TUM-Handheld SLAM / rgbd_dataset_freiburg*) experiment than the other three algorithms, except for the sequence *freiburg1_desk*, get a lower rate of the keyframe losing.

Table 2 Frame lose ratio comparison (dataset experiment)

Algorithm		PTAM		LSD-SLAM		ORB-SLAM		Proposed	
Sequence	**Total number of sequence**	**Extracted number of key frames**	**Key Frame loss rate(%)**	**Extracted number of key frames**	**Key Frame loss rate(%)**	**Extracted number of key frames**	**Key Frame loss rate(%)**	**Extracted number of key frames**	**Key Frame loss rate(%)**
freiburg1_360	1517	67	95.58	65	95.71	72	95.25	83	**94.53**
freiburg1_desk	1214	60	95.05	61	94.97	66	94.56	65	94.64
freiburg1_floor	1285	72	94.39	72	94.39	75	94.79	79	**94.47**
freiburg1_room	2728	114	95.82	118	95.67	129	95.27	140	**94.87**
freiburg2_360_hemisphere	5462	286	94.76	273	95.00	297	94.56	332	**93.57**
freiburg2_desk	5935	269	95.47	297	94.00	285	95.12	338	**94.30**
freiburg2_large_no_loop	6721	392	94.17	430	93.60	433	93.59	467	**93.05**
freiburg3_long_office_household	5100	345	93.24	349	93.16	397	92.22	420	**91.76**

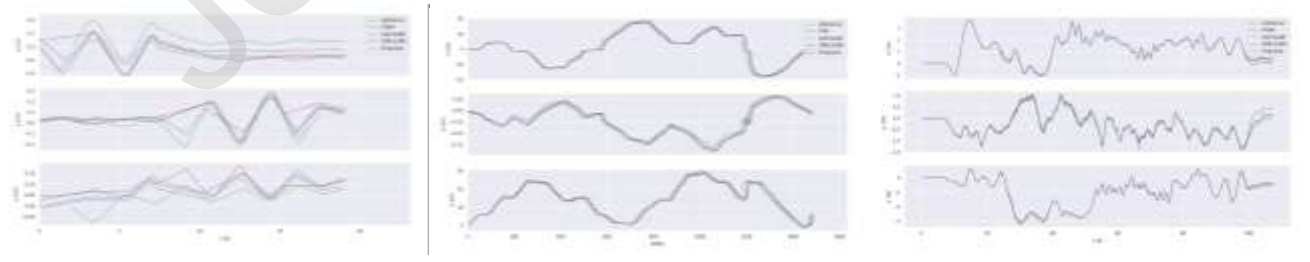
Trajectory Comparison Test

In this section, we run the algorithm proposed in this article (represented by "Proposed" in the figures) by dataset TUM , and plot the trajectory and error. At the same time, we also run PTM, LSD-SLAM, ORB-SLAM for comparison. We use sequences of the *TUM/fre1_xyz* , *KITTI/sq_00* and *EUROC/MH_01_easy* to evaluate the trajectory deviation of the mapping location. The trajectory deviation reflects the overall error of the positioning map of SLAM, the deviation in the X,Y and Z coordinate direction reflects the attitude deviation at each moment, and the pitch, yaw and roll, reflects the posture change trend at each moment, respectively.



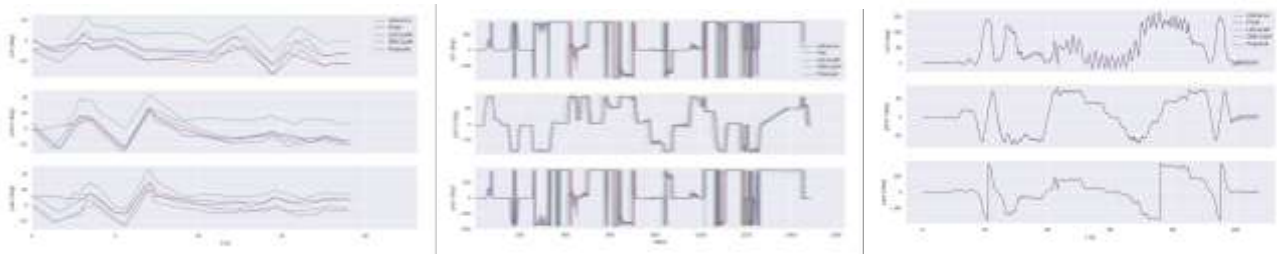
(a) (b) (c)

Figure 9 Trajectory deviation comparison, (a) (b) (c) represent experiments based on TUM / fre1_xyz, KITTI / sq_00 and EUROC / MH_01_easy, respectively. Figure 10 and 11 are the same.



(a) (b) (c)

Figure 10 Coordinate comparison of X,Y and Z



(a) (b) (c)

Figure 11 Comparison of Roll, Pitch and Yaw

In addition, we also use the root mean square error (RMSE) to judge the accuracy of the algorithm in this paper. The reason for the large root mean square error is that the Local Mapping process spends a lot of time in the target detection process, which means that the efficiency of extracting keyframes is reduced. The RMSE calculation formula is as follows:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n}(M_{obs,k} - M_{\mathrm{model},k})^2}{n}} \qquad (23)$$

Among them, n represents the total number of observations, k represents the number of observations, M_{obs} and M_{model} represents the actual pose and the true pose provided by ground truth, respectively. We have operated five sequence tests based on the three datasets of TUM, KITTI and EUROC, and averaged them to give the root mean square error comparison table of the improved visual SLAM based on Affine-ORB and the other three algorithms. From the data in Table 3, we can clearly conclude that RMSE of the algorithm proposed in this paper is significantly smaller than the other three algorithms in addition to the sequences *Fr2/rpy*, *Fr1/desk*, and *KITTI/Sq_08*. Because the method in this paper extracts keyframes more quickly, obtains more keyframes in the same time, it gets more accurate trajectories, and provides a better initial value for the backend.

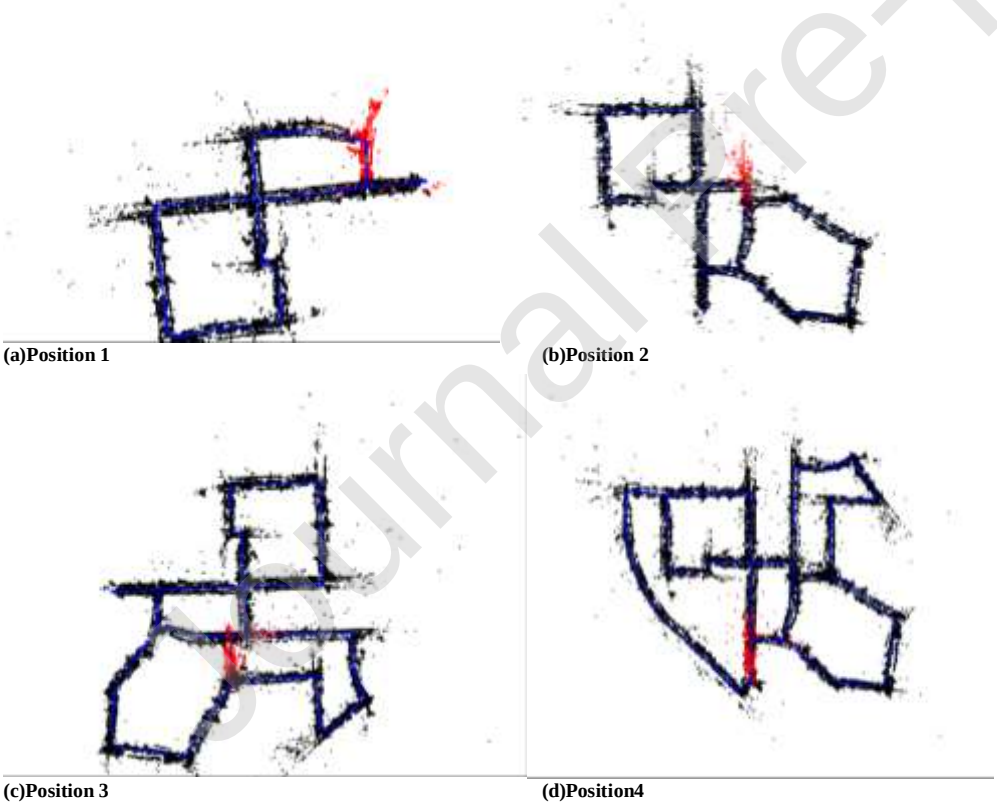
Table 3 Comparison of root mean square error(RMSE)

	Absolute Frame Trajectory RMSE($m\times10^{-2}$)			
datasets	**PTAM**	**LSD-SLAM**	**ORB-SLAM**	**Proposed**
TUM	Fr1/rpy	2.0128	5.1054	1.5327	**1.4439**
	Fr2/xyz	3.0102	4.1135	3.6288	**2.5748**
	Fr2/rpy	4.2153	3.5474	3.2544	3.3556
	Fr1/360	10.6352	12.5479	11.6587	**10.4518**
	Fr1/floor	7.2549	6.5247	6.8597	**5.9874**
	Fr1/desk	12.5748	10.5630	13.4271	12.0857
	Fr2/pioneer_slam	8.5477	7.6487	6.5273	**6.4025**
	Fr2/pioneer_slam2	10.4750	10.2489	9.5324	**9.4563**
	Fr2/pioneer_slam3	14.6879	15.2401	12.0217	**11.3578**
	Sq_01	11.6274	7.8522	4.3217	**4.2108**
	Sq_02	6.8542	8.2479	5.9874	**3.9981**
	Sq_03	14.2314	19.8741	15.2201	**9.0479**

	Sq_04	6.5478	8.0214	7.3549	**6.0610**
KITTI	Sq_05	3.5784	3.2571	3.5210	**3.0478**
	Sq_06	2.8501	2.6322	2.9876	**2.0507**
	Sq_07	11.8749	10.8745	10.0037	**9.7801**
	Sq_08	5.2417	5.0214	4.9876	4.8572
	V2_02_medium	6.2987	6.3578	6.0147	**5.8749**
EUROC	V2_02_difficult	13.6574	13.6932	15.0874	**13.1022**
	MH_03_medium	7.6341	7.5862	8.2415	**6.8461**
	MH_04_difficult	15.0478	14.5647	12.3363	**11.2241**

Loop Closing Detection Experiment

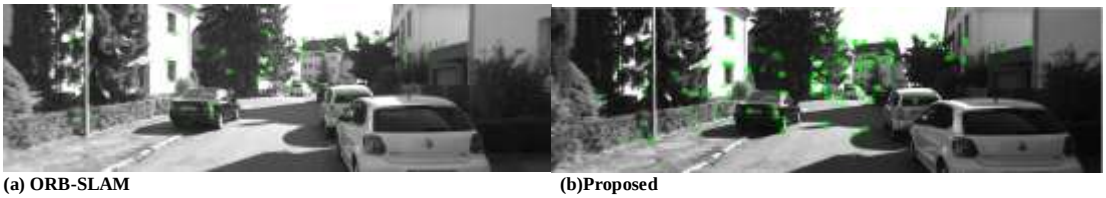
Loop closing detection is an important step to determine whether the robot has returned to the origin. It relates to the trajectory of the robot and the accuracy of the map construction over a long period of time. On the other hand, the robotic device can still use the loop closing detection method to relocate after losing track because the loop closing detection provides the correlation between the current data and all historical data. The improvement of loop closing detection accuracy is quite noticeable to the improvement of the accuracy and robustness of the entire SLAM system.



(a)Position 1 **(b)Position 2**

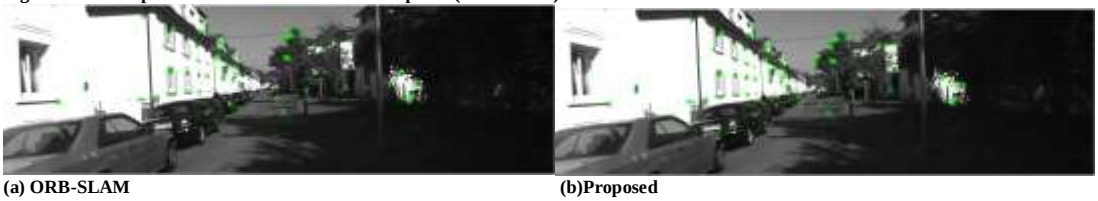
(c)Position 3 **(d)Position4**

Figure 12. Schematic diagram of Loop Closing detection (The red sparse point represents the loop detection position)



(a) ORB-SLAM (b)Proposed

Figure 11. Comparison of the number of feature points(Position Ⅰ)



(a) ORB-SLAM (b)Proposed

Figure 12. Comparison of the number of feature points(Position Ⅱ)

(a) ORB-SLAM (b)Proposed

Figure 13. Comparison of the number of feature points(Position Ⅲ)

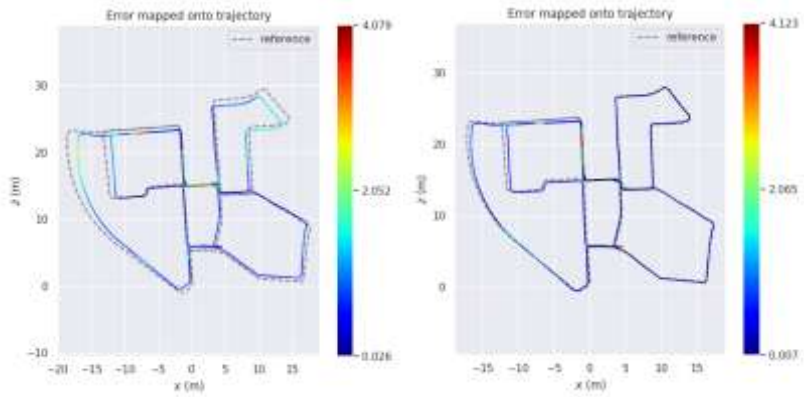
(a) ORB-SLAM (b)Proposed

Figure 14. Comparison of the number of feature points(Position Ⅳ)

Table 4 Comparison table of feature points

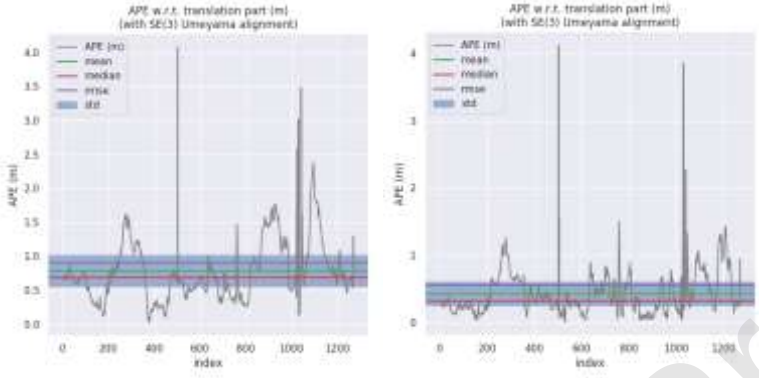
Loop Closing detection point	ORB-SLAM	Propoesd
Ⅰ	39	81
Ⅱ	41	67
Ⅲ	49	92
Ⅳ	51	95

Due to the fact that PTAM and LSD-SLAM are not considered scenarios about loop closing detection, only ORB-SLAM is designed with a loop closing detection step, so this section only conducts comparison experiments with ORB-SLAM. The experiments in this paper are compared with the KITTI dataset 00 sequence, because the images of the KITTI00 sequence collected in a community with street scene, and the collection device passed through several same locations, which is very suitable for loop closing detection experiments of visual SLAM. The experimental diagrams (Figure11-14) and Table 4 show that at four different points of loop closing detection, the feature point extraction ability of the algorithm in this paper is better. Through ten experiments, the comparison of the number of feature points is definitely given after selecting the average date. At the same time, the comparison results of the total trajectory error comparison and the absolute pose error are showed in Figure 15-16. It can obviously see that the overall trajectory overview of this paper is closer to the ground truth, and the absolute trajectory error is smaller than ORB-SLAM algorithm. It indicates that the relocation capability of the algorithm in this paper is better, and a more accurate global consistent map can be obtained eventually.



(a) ORB-SLAM **(b)Proposed**

Figure 15. Comparison of the trajectory



(a) ORB-SLAM **(b)Proposed**

Figure 16. Comparison of the absolute position error(APE)

Table 5 Comparison table of the absolute position error(APE)

	ORB-SLAM	Propoesd	Improvement(%)
Mean(m)	0.774	0.485	37.34
Median(m)	0.689	0.352	48.91
Rmse(m)	0.902	0.526	41.69

Conclusion And Future Direction

This paper proposes an improved visual SLAM method based on affine transformation feature extraction. An innovation suggests in the feature extraction of front-end visual odometry of visual SLAM by the ORB method based on affine transformation. In this method, the number of feature points and keyframe acquiring is increased, the reliability of the front-end visual odometry of the SLAM system is enhanced, and the accuracy of positioning and mapping is also improved by applying the mathematical method of affine transformation to ORB feature extraction, The results show that the method proposed in this paper extracts keyframes faster and the visual odometry works better. Compared with the three classic SLAM methods of PTM, LSD-SLAM and ORB-SLAM, it obtains more continuous and accurate trajectories of location. However, the research in this paper still has limitations: Firstly, this article is an improved simulation experiment based on the ORB-SLAM algorithm. It mainly uses the classic datasets of other researchers, and lacks the image datasets gathered by ourselves. Secondly, the experimental device is a handheld RGB-D camera, not a real robotic device, so the experiment of Handheld SLAM only simulates the motion trajectory of the robotic device. Our future work is to conduct a physical robot verification of the method to make the experimental data more convincing. At the same time, it is necessary to begin further studies, such as the relationship among keyframe detection and global location, back-end optimization. Besides, the combination of laser sensors to continuously optimize is also an aspect to enhance the real-time and accuracy of the visual SLAM.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This paper is supported by Research on Intelligent Brewing Device and Its Key Technology Based on Solid State Fermentation(2019YFN0104) and Research on Substation Inspection Robot Based on Multi-sensor Fusion(2016ZGY021)

References

[1]H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part I,” IEEE Robotics & Automation Magazine. 2006;13: 99-110.
[2]E. Mouragnon, M. Lhuillier, M. Dhome, et al., “Realtime localization and 3D reconstruction,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 2006: 363–370.
[3]M. Adams, B.N. Vo, R. Mahler, J. Mullane, et al. “SLAM Gets a PHD: New Concepts in Map Estimation” IEEE Robotics & Automation Magazine. 2014;21: 26-37.
[4]S. Ortega, H. Fabelo, D. K. Iakovidis, A. Koulaouzidis, and G. M. Callico, “Use of Hyperspectral/Multispectral Imaging in Gastroenterology. Shedding Some–Different–Light into the Dark,” Clinical Medicine., vol. 8, no. 1, pp. 36-46, Jun. 2019.
[5]J. E. Guivant, E. M. Nebot, “Optimization of the simultaneous localization and map-building algorithm for real-time implementation,” IEEE Transactions on Robotics and Automation. 2001;17 :256-267.
[6]A. Markus W., A. Michael, W. Stephan and S. Roland, “Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments,” in IEEE International Conference on Robotics and Automation (ICRA 2011), Shanghai, China. 2011: 3056-3063.
[7]C. Cadena, L. Carlone, H. Carrillo, et al., “Past, present, and future of simultaneous localization and mapping: toward the robust-perception age,” IEEE Transactions on Robotics. 2016; 32: 1309-1332.
[8]J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. Manuel Rendón-Mancha, “Visual simultaneous localization and mapping: a survey,” Artificial Intelligence Review. 2015; 43: 55-81.
[9]M. Wu, “Research on optimization of image fast feature point matching algorithm,” EURASIP Journal on Image and Video Processing., 2018. doi: 10.1186/s13640-018-0354-y.
[10]J. Xu, H. W. Chang, S. Yang, et al., “Fast feature-based video stabilization without accumulative global motion estimation,”. IEEE Transactions on Consumer Electronics. 2015; 7:174-187.
[11]C. C. Hsu, C. K. Yang, Y. H. Chien, et al., “ Computationally efficient algorithm for vision-based simultaneous localization and mapping of mobile robots,” Engineering Computations. 2017; 34: 1217-1239.
[12]A. J. Davison, I. D. Reid, N. D. Molton, et al., “MonoSLAM: real-time single camera SLAM,” IEEE Trans Pattern Anal Mach Intell. 2007; 29: 1052-1067.
[13]C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semidirect monocular visual odometry,”in IEEE International Conference on Robotics and Automation (ICRA), Hong Kong. 2014:15-22.
[14]T. Qin, P. Li and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” IEEE Transactions on Robotics.2018; 34: 1004-1020.
[15]G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” in Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, Nara, Japan, 2007.
[16]P. Sun, N. Lu and M. Dong, “Modelling and calibration of depth-dependent distortion for large depth visual measurement cameras,” Optics Express. 2017; 25: 9834.
[17]R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System,” IEEE Transactions on Robotics. 2015; 31: 1147-1163.
[18]R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras,” IEEE Transactions on Robotics. 2017; 33: 1255-1262.
[19]X. Gao, “Vision SLAM Lecture 14: From Theory to Practice,” China: Electronic Industry Press, 2017: 134–205.
[20]H. Liu, G. Zhang and H. Bao, “Robust keyframe-based monocular SLAM for augmented reality,” in International Symposium on Mixed and Augmented Reality, ISMAR, Merida, Yucatan, Mexico, 2016.
[21]S. Li, D. Yang, G. Tang, et al., “Atomic Norm Minimization for Modal Analysis from Random and Compressed Samples,” IEEE Transactions on Signal Processing. 2017: 99, Mar.
[22]W. N. Greene, K. Ok, P. Lommel, “Multi-Level Mapping:Real-time Dense Monocular SLAM,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016: 833-840.
[23]A. Petrovskaya, M. Perrollaz, L. Oliveira, “Awareness of Road Scene Participants for Autonomous Driving,” Sensing and Actuation. 2012: 1385-1431.

[24]Y. Zhou, K. Han, C. Luo,et al., "SVO-PL: Stereo Visual Odometry with Fusion of Points and Line Segments," 2018 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, Shenzhen, China, 2018: 900-905.
[25]J. Engel, V. Koltun and D. Cremers. "Direct Sparse Odometry," IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 40: 611-625.
[26]J. Engel, S. Thomas and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in 13th European Conference on Computer Vision(EECV), Zurich, Switzerland. 2014: 834-849.
[27]W. Zhao and C. Ngo, "Flip-Invariant SIFT for Copy and Object Detection," IEEE Transactions on Image Processing. 2012; 22: 980-991.
[28]A. D. Sorbo, S. Panichella, C. V Alexandru, et al., "SURF: Summarizer of User Reviews Feedback," in 39th IEEE International Conference on Software Engineering (ICSE 2017), Buenos Aires, Argentina, 2017: 55-58.
[29]A. Akula, R. Ghosh, S. Kumar, "WignerMSER: Pseudo-Wigner Distribution Enriched MSER Feature Detector for Object Recognition in Thermal Infrared Images," IEEE Sensors Journal. 2019; 19: 4221-4228.
[30]A. Sluzek, "Contextual descriptors improving credibility of keypoint matching: Harris-Affine, Hessian-Affme and SIFT feasibility study," 13th International Conference on Control Automation Robotics & Vision (ICARCV).IEEE, Marina Bay Sands, Singapore, 2014: 117-122.
[31]J. M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," SIAM Journal on Imaging Sciences. 2009; 2: 438-469.
[32]E. Rublee, V. Rabaud, K. Konolige, et al., "ORB: An efficient alternative to SIFT or SURF," in 2011 International Conference on Computer Vision. IEEE, Barcelona, Spain. 2011: 2564-2571.
[33]L. F. O. Chamon and A. Ribeiro, "Greedy Sampling of Graph Signals," IEEE Transactions on Signal Processing. 2017; 99: 1255-1262.
[34]D. Nistér, O. Naroditsky and J. R. Bergen, "Visual odometry for ground vehicle applications," Journal of Field Robotics. 2006; 23: 3-20.
[35]E. Mueggler, H. Rebecq, G. Gallego, et al., "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," The International Journal of Robotics Research. 2017; 36: 142-149.
[36]J. Zhang, M. Kaess and S. Singh, "A real-time method for depth enhanced visual odometry," Autonomous Robots. 2017; 41: 31-43.
[37]S. Leutenegger, M. Chli, R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, 2011: 2548-2555.
[38]Z. Ji, A. Li, T. Feng, et al., "The benefits of Tai Chi and brisk walking for cognitive function and fitness in older adults," PeerJ.2017; 5: 13-30.
[39]Z. Xie, D. Qiu and G. Cai, "Quantum algorithms on Walsh transform and Hamming distance for Boolean functions," Quantum Information Processing. 2018; 17: 139.
[40]V. Fedorov and C. Ballester, "Affine Non-Local Means Image Denoising," IEEE Transactions on Image Processing.2018; 26: 2137-2148.