

## Research Article

# Loop Closure Detection Based on Differentiable Manifold

Tianzhen Dong, Bin Xue , Qing Zhang, Yuepeng Zhao, Wenju Li, and Mengying Li

*School of Computer and Information Engineering, Shanghai Institute of Technology, Shanghai201400, China*

Correspondence should be addressed to Bin Xue; 206141138@mail.sit.edu.cn

Received 6 May 2022; Accepted 17 May 2022; Published 16 June 2022

Academic Editor: M. Praveen Kumar Reddy

Copyright © 2022 Tianzhen Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Loop closure detection is an important part of SLAM (simultaneous location and mapping), which can effectively reduce the cumulative error of the system after long period of exploration. The existing loop closure detection methods are mainly to evenly distribute the accumulated error in the robot trajectory, but the motion error of the actual robot is also related to its motion speed and rotation angle, while the corrected motion trajectory of the robot is difficult to match the real trajectory. Based on the analysis of the mechanism of robot motion error, this paper proposes a novel loop closure detection method based on differentiable manifold, which mainly includes real-time pose based on manifold tangent space and smooth motion trajectory model of robot based on differential geometry. Firstly, we introduce the Frenet framework structure and establish the corresponding manifold tangent space theory for the keyframe pose nodes. The real-time problem of robot motion is equivalent to the problem of finding the optimal angle tangent vector. Secondly, the motion speed between keyframes is used to determine the characteristics of the robot motion trajectory. We calculate the curvature and torsion of the curve composed of several nodes based on the manifold tangent space and then combine the curve interpolation and fitting of the keyframe nodes to achieve the approximation of the robot motion trajectory, and the smooth curve of the robot trajectory is obtained. Finally, the experiment verifies that the method in this paper can effectively ensure the continuity and smoothness of the robot's trajectory, thereby reducing the cumulative error of the system and improving the accuracy of loop closure detection.

## 1. Introduction

The mobile robot creates an environment map based on its own position and sensor data in unfamiliar scenes, which is used to guide the robot's autonomous positioning and navigation, i.e., simultaneous location and mapping (SLAM) is the key to the realization of autonomous mobile robots [1, 2]. For example, autonomous mobile robots, such as mobile robots, outdoor robots, and autonomous vehicle, need to know where they are and their surroundings and maps in order to know how to get there. SLAM technology mainly solves the problem of map construction and real-time positioning of robots. It can effectively guide robot movement and build a map of the surrounding environment, so it is very important for autonomous mobile robots. Loop closure detection is an important module in the SLAM system, and adding this module can effectively reduce the cumulative error in the entire SLAM system and form a globally consistent trajectory and map [3–5].

Most of the existing loop closure detection methods are based on the appearance of the scene and mainly include image pairs-based methods [6–12], image sequences-based methods [13–17], and image matching methods integrating spatial information [18–23]. The image pairs-based methods use effective image retrieval and matching methods to analyze the similarity between the current view image and the previously collected image. If the similarity is greater than the set threshold value, it will be selected as a closed-loop candidate frame. Finally the wrong candidate matching frames will be eliminated by the method based on geometric verification. However, the features are artificially designed and can only cope with limited scene changes in those methods. They are hard to extract high-level information of image and cannot express complex structural information, so it is difficult to deal with drastic appearance changes. The image sequences-based methods consider the time continuity of the images in SLAM, using the feature matching method to dynamically segment the image

sequence to find the local optimal match for each query image in all image sequence segments. The image sequences-based methods mainly consider the temporal continuity of images in SLAM. Firstly, the image sequence is dynamically divided into multiple small sequence segments by the feature matching method, and then all possible image pairs are queried in each small sequence segment. Secondly, calculate their sequence similarity and create the similarity matrix and then select the local template sequence with the highest statistical score in this matrix as the loop closure candidate region. Finally, using the RANCAS algorithm to determine the final loop closure, the RANSAC algorithm is used to calculate the transformation matrix between the candidate loop closure frame and other frames. If the transformation matrix is close enough to the identity matrix, the loop closure is determined to be successful; otherwise, it fails. But these methods do not consider the spatial geometric relationship of features in the image, so they are difficult to use when facing the change of viewpoint. The image matching method integrating spatial information is to construct each image into a graph structure composed of several nodes and edges and then calculate the similarity to determine the optimal target image from the candidate images according to their matrix information or topology information.

In recent years, the research of deep learning technology in computer vision has also achieved rapid development and has achieved great success in image classification, image segmentation, image detection, and other fields [24–28]. Among them, the loop closure detection problem based on deep learning can be regarded as a classification task based on a large number of scene images in essence [29]. It is a feasible idea to apply the ability of deep learning to learn feature rules in a hierarchical manner in massive data to loop closure detection. The main contents of visual odometry include extracting and matching image feature points and then estimating the camera motion between adjacent cameras based on the extracted image information. Konda and Memisevic [30] first proposed an end-to-end convolutional neural network and used it in visual odometry, which can effectively predict changes in camera direction and speed in the actual environment. Constante et al. [31] found that the use of convolutional neural networks can effectively estimate the position between frames when the image is subject to illumination changes and motion blur. Xia et al. [32] performed feature extraction based on the AlexNet model and then used the SVM algorithm for secondary training loop detection model, which is more robust than artificially designed feature algorithms. After extracting image features through the self-encoding algorithm in unsupervised learning, Gao and Zhang [33] used the similarity matrix as a measurement method for loop detection and achieved good results on the dataset. There are also loop closure detection methods with the aid of some auxiliary tools such as gyroscopes, inertial navigation, and other equipment [34, 35], which improves the accuracy but also increases expensive costs.

However, most of the above methods calculate the similarity between the images based on the image matching

method, and the cumulative errors in the robot movement process are equally distributed to each keyframe or use optimized methods (such as the bundle adjustment method) to reduce these cumulative errors so as to obtain more accurate pose estimation. Although the existing method can detect the loop region, it does not consider that the accuracy of the loop closure detection is also closely related to the motion trajectory of the robot. Because the motion trajectory of the robot in the real scene is complicated and changeable due to various noises, when the robot is running for a long time, there are errors between the real-time pose and the real pose of the robot calculated only by the image matching method. The gradual accumulation will eventually cause the robot's trajectory to shift, making it difficult to form a globally consistent trajectory and map.

In order to solve the above problems, in this paper, we propose a loop closure detection method based on differentiable manifold (Figure 1). Firstly, we obtain the relative pose between the cameras according to the image feature descriptor and use the image matching algorithm to determine the keyframe pose nodes. Secondly, the Frenet framework structure is introduced to establish the manifold tangent space for each keyframe pose node. In this space, the robot real-time motion problem is equivalent to solving the optimal angle tangent vector problem to ensure the accuracy and real-time performance of the robot motion. Thirdly, the robot motion trajectory is studied from coarse to fine. The simple robot motion trajectory is analogous to the first-order differentiable manifold structure, while complex conditions of trajectory can be composed of several simple segments. We guaranteed the continuity and smoothness in each trajectory segments to reduce the error of whole trajectory. Finally, compared with the robot motion trajectory obtained by only using the image matching method, the robot motion trajectory obtained by this method is smoother and more consistent with the real trajectory, which can effectively reduce the cumulative error of the entire system. In general, the contributions of this paper are as follows:

- (1) The trajectory model of the robot is established. The Frenet framework is introduced for each keyframe pose node, and the manifold tangent space theory is introduced on the basis of this framework. The real-time motion problem of the robot is equivalent to the problem of finding the optimal angle tangent vector in the manifold tangent space to ensure the accuracy and real time of robot movement.
- (2) We propose a robot smooth motion trajectory model based on differentiable manifold. We judge the characteristics of the robot's motion trajectory based on the motion speed between adjacent frames and use the continuity and smoothness in the differentiable manifold according to its motion trajectory characteristics. Then, the curvature and torsion of the curve formed by several nodes are calculated. Finally, the robot trajectory is approximated by the

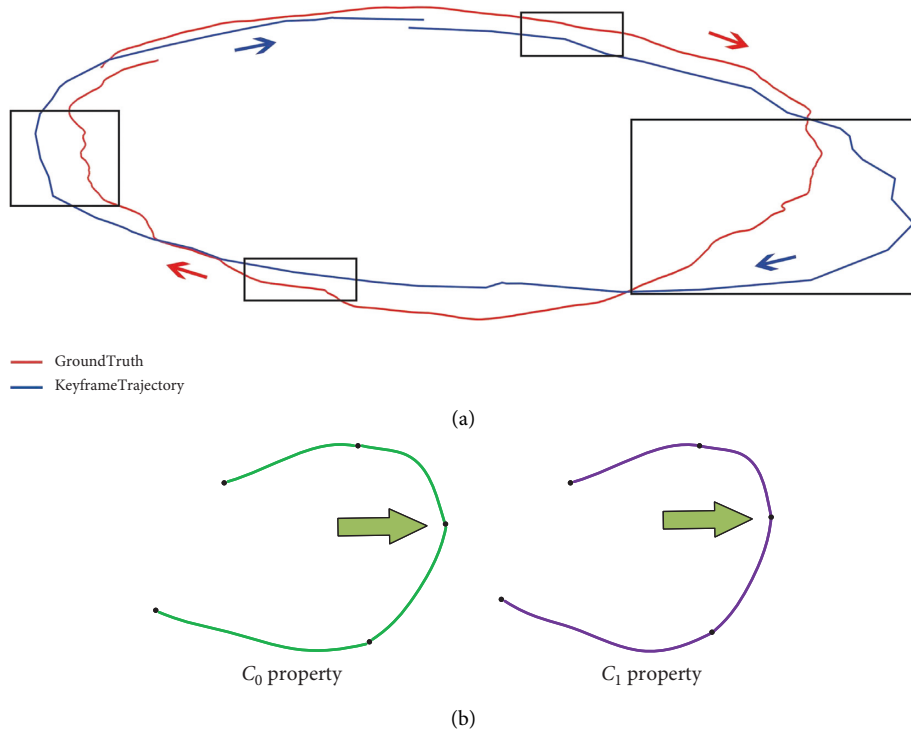


FIGURE 1: (a) Schematic diagram of loop closure detection based on differentiable manifold. The arrow indicates the direction of the robot. The blue trajectory represents the robot motion trajectory calculated only by the image matching algorithm, while the red trajectory represents the robot motion trajectory in the real scene, and the black box marks the trajectory smoothness problem proposed in this paper. (b) Black nodes represent keyframe pose nodes,  $C_0$  property means only connection, and  $C_1$  property means continuous and smooth.

curve interpolation and fitting method, and the smooth curve of the robot trajectory is obtained.

The rest of this paper is organized as follows. In the second section, we summarize the related research on loop closure detection. In the third section, we describe our proposed method in detail. Our experimental design and comparison results are presented in Section 4. Conclusions and future work are discussed in Section 5.

## 2. Related Work

This section discusses traditional loop closure detection methods based on image matching, the topics of which contain image pairs-based methods, image sequences-based methods, and image matching methods integrating spatial information.

**2.1. Image Pairs-Based Methods.** The matching method based on image pairs mainly use the bag of words (BoWs) model [10] and its improved methods. The purpose of bag of words model is to describe an image with features on the image. It mainly includes the following three steps. Firstly, determining the words, which are the combination of a certain type of features, and these words form a dictionary. Then, describing the entire image with the occurrence of words or histograms and converting an image into a vector description. Finally, comparing the degree of similarity described in the previous step. Its basic idea is to convert the

descriptors of local feature points of the image into words and then form a dictionary from these words. Finally, word bag vectors are counted for the words of the whole image. The distance between word bag vectors represents the difference between images. There are also manual keypoints extracted from the image, and then the similarity between feature descriptors is calculated. For example, SIFT (scale invariant feature transform) algorithm [36] is used in FAB-MAP [6] to extract features and generate corresponding feature descriptors. Common image feature descriptors also include SURF (speed up robot features) [37] and ORB (oriented fast and rotating brief) [38, 39]. In addition, there have been many methods based on global image descriptors. Protzel and Sunderhauf [40] applied GIST [41] to place recognition, encoding the response of the image in different directions and scales as a global description through Gabor filters. Nourani-Vatani et al. [42] proposed the optical flow information method (OFM) to detect and describe the image; this is achieved by using the intensity values of adjacent pixels. They used a Canny edge detector to subdivide the image into small patches of  $20 * 20$  pixels, extracted the robust flow direction based on the optical flow signal, and then calculated optical flow based on the Lucas-Kanade algorithm. Naseer et al. [43] modeled image matching as a minimum cost flow problem in the data association graph and matched image pairs using the hog descriptor of the image. Negre Carrasco et al. [44] generated a global signature for each image of video sequences by using multiple orthogonal projections. Guclu and Can [45]

performed image histogram and autocorrelogram comparison between images separately to find a similar candidate group.

*2.2. Image Sequences-Based Methods.* Milford and Wyeth [13] constructed short image sequences by using local contrast enhancement processing method according to the deviation matrix between image sequences and then found local best matching for each query image in all short image sequences. Based on the hidden Markov model, Hansen and Browning [17] retrieved the dataset image sequence matching the query image sequence by calculating the image similarity probability value matrix. Bampis et al. [46] segmented the image dataset into groups based on their spatiotemporal proximity and then used sequence-visual-word-vectors as a means to find matches between them. Bampis et al. [15] dynamically segmented the robot's motion according to the picture content and created sequence-vision-word vectors combined with the visual words in each sequence. Abdollahyan et al. [16] modeled image sequences to form strings by using directed acyclic graph and then used partial order kernel to compare strings. Tsintotas et al. [18] dynamically divided the input image stream and assigned with VVs by using an online clustering algorithm. Arroyo et al. [47] used all possible image pairs in the template to calculate the sequence similarity, queried the sequence to create a similarity matrix, and then selected the local template sequence with the highest statistical score from the matrix as the loop closure candidate. Rodrigues et al. [48] proposed a three-layer sequence-based loop closure detection method. The first layer uses the recently visited place sequence as the query to search for candidate sequences in the topological map composed of previously visited places. In the second layer, among all the candidate sequences, the candidate sequences that are consistent with the previous response time are selected for matching. Finally, the image sequence belonging to the query sequence is matched with the candidate sequence selected by the second layer.

*2.3. Spatial-Based Methods.* Finman et al. [21] performed a convolution operation in the dataset to detect features and connected these features to construct a sparse object map for location recognition. Chen et al. [19] used the method of scene segmentation to integrate the visual features of the fusion spatial information into BoW and then set up a k-means hierarchical association dictionary to associate multiple visual words. Valigi et al. [20] proposed to observe the underlying landmarks together, then connect the two nodes, and model the image as a graph structure composed of nodes and edges based on co-visibility graph. Gawel et al. [49] utilized a graph structure to encode the spatial relationship of landmark regions, and their model had strong robustness against viewpoint changes. An et al. [22] used the convolutional neural network (CNN) and SURF [37] to extract frame features and then constructed the graph vocabulary based on the hierarchical navigable small world

(HNSW) graph to effectively detect loop areas. Wang et al. [50] introduced semantic topological graphs to encode the spatial information of landmarks and used random walk descriptors to characterize topological graphs for graph matching. Gao and Zhang [51] proposed a multi-order graph matching method for loop closure detection in addition to vector-based descriptors.

How to effectively reduce the cumulative error in robot motion has always been a problem studied by researchers. Although the above methods can detect the area where the loop occurs, they mostly use methods (such as bundle adjustment [52]) to spread the cumulative error of the robot to each keyframe for optimizing the entire robot motion trajectory and reducing the error of the entire system. However, the actual motion trajectory of the robot is mainly affected by its motion speed and rotation angle, while the existing methods are difficult to ensure the continuous smoothness of the robot motion trajectory. To solve the above-mentioned problems, this paper proposes a loop closure detection method based on differentiable manifolds. This method mainly includes the following. Firstly, the accuracy and real-time performance of the robot movement pose are guaranteed in the manifold tangent space. Secondly, we judge the characteristics of the robot's motion trajectory based on the motion velocity between adjacent frames and reasonably use the continuity and smoothness in the differentiable manifold according to its motion trajectory characteristics. Then, the curvature and torsion of the curve formed by several nodes are calculated, and the curve interpolation and fitting method are combined to achieve the approximation of the robot's motion trajectory so as to make it consistent with the real trajectory, thereby solving the problem that the pose trajectory calculated by the image matching algorithm is not smooth enough. Finally, in the mobile robot visual loop closure detection experiment, the method in this paper can be used effectively to ensure the smoothness of the robot's motion trajectory, thereby reducing the cumulative error of the system and improving the accuracy of loop closure detection.

### 3. Our Approach

The existing loop closure detection methods mainly use a method such as the bundle adjustment [53] to distribute the cumulative error of the robot motion trajectory to the keyframes for optimizing the entire robot trajectory. However, the motion error of the robot in the real scene is also related to its motion speed and rotation angle. It is difficult to effectively reduce the cumulative error of the system only by relying on simple keyframe amortization strategies or optimization methods. This paper proposes a novel loop closure detection method based on differentiable manifold through the analysis of the mechanism of robot motion error, which mainly includes real-time pose based on manifold tangent space and smooth motion trajectory model of robot based on differential geometry. The method used in this paper will be described in detail below.

**3.1. Robot Motion Trajectory Model.** The trajectory equation of robot movement in space can be abstracted as

$$S = S(t) = \{S^1(t), S^2(t), \dots, S^n(t)\} (t \in I), \quad (1)$$

where  $S^i(t)$  represents the pose of the camera at  $i$  at time  $t$  and  $I$  represents the temporal and spatial region. In the process of robot motion, this paper introduces a new frame of coordinate system for the robot motion trajectory equation  $S$ , namely, the Frenet frame (Figure 2), and the specific expression is as follows:

$$\begin{aligned} T &= \frac{dS}{dt}, \\ B &= T \times N, \\ N &= B \times T. \end{aligned} \quad (2)$$

Among them,  $T$ ,  $N$ , and  $B$ , respectively, represent the tangent vector, main normal vector, and binormal normal vector of the robot motion trajectory curve.  $dS/dt$  represents the first derivative of the trajectory equation  $S$  with respect to time  $t$ .

Since the robot is doing a rigid body motion which is constantly moving, when the robot moves along the curve, the frame  $\{P; T, N, B\}$  also moves as a rigid body, which is more consistent with the robot's motion state. Based on this framework, we can accurately describe the robot motion state (motion speed and rotation angle). Moreover, since the motion state of the robot in a certain pose node will be affected by the neighboring state, the introduction of this framework is convenient for us to study the neighborhood information of each pose node on the trajectory equation  $S$ .

**3.2. Real-Time Motion Based on Manifold Tangent Space.** In the process of robot motion, compared with European space, the characteristics of curve can be better described in manifold tangent space; especially in describing the smoothness of curve, manifold tangent space has more advantages, and the actual motion of robot is also a curve with different smoothness. In Section 3.1, we gave the general form of the robot's motion trajectory, but the robot will encounter a basic and important problem in the actual operation process, namely, how to ensure the accurate and real-time movement of the robot. In order to solve this problem, we introduce the manifold tangent space and seek the optimal tangent vector in it, so that the robot can not only ensure the accuracy of the motion when moving but also help in accurately calculating the next pose and consequently solving the offline issues.

Given the robot motion trajectory  $S$ , as shown in Figure 3, the corresponding tangent space will be found at each pose node  $p$  of the trajectory as follows:

$$T_p M = \left\{ \frac{d\gamma(0)}{dt} \mid \gamma \in C^\infty(R^1, M), \gamma(0) = p \in M \right\}, \quad (3)$$

where  $T_p M$  represents the tangent space of the surface  $M$  at the point  $p$ ,  $T_p M$  is the set of tangent vectors of all the

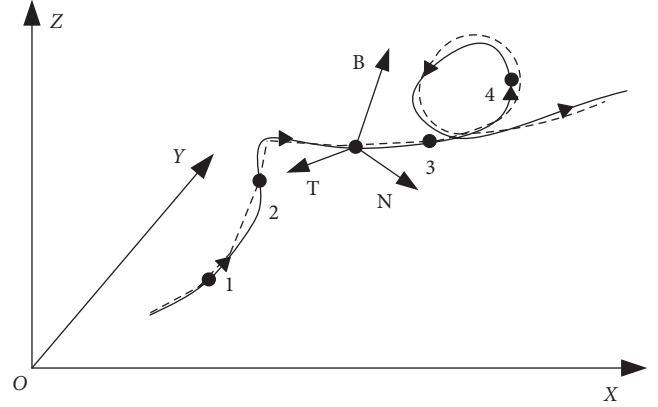


FIGURE 2: Schematic diagram of robot trajectory. The solid line represents the real motion trajectory of the robot, and the dotted line represents the motion trajectory of the robot calculated only by the image matching algorithm, while the arrow represents the motion direction of the robot. The Frenet framework is composed of  $\{T, (N) B\}$ , where  $T$ ,  $N$ , and  $B$  represent tangent vector, the main normal vector, and the binormal vector, respectively.

smooth curves passing through the point  $p$  on  $M$ ,  $d\gamma(0)/dt$  represents the tangent vector of  $M$  at point  $p$ , and  $\gamma$  is a smooth curve on the surface  $M$  that passes through point  $P$ , that is,  $\gamma \in C^\infty(R^1, M)$ ,  $\gamma(0) = p$ . Essentially speaking, the tangent space is the space generated by the infinite linear expansion of the infinitesimal neighborhood of  $M$  at the point  $p$  (it is a flat infinitesimal  $n$ -dimensional vector space).

When the robot is moving, the pose  $p_t$  on the trajectory at time  $t$  will be affected by other nodes in its neighborhood (mainly adjacent moments). The prediction of the pose is to screen out the next motion state  $\psi$  from  $\psi_1, \psi_2, \dots, \psi_i$ . When selecting the optimal tangent vector  $\psi$  in this manifold tangent space, not only can the robot face the correct orientation to avoid offline problems but also the precise pose of the robot at the next moment can be obtained.

**3.3. Smooth Motion Trajectory of Robot Based on Differential Geometry.** After Section 3.2, we ensured the accuracy and real time of the robot's trajectory, but there is still a problem to be solved, how to optimize the robot's trajectory to match the real trajectory as much as possible. As shown in Figure 2, it can be found that in the real scene, the robot performs a linear motion between the keyframe pose nodes 1 and 2, while the trajectory calculated by the image matching algorithm is a straight line. It is more obvious that in real scenes the robot performs curve motions between nodes 2 and 3 and between nodes 3 and 4. Although the existing calculation methods calculated are also curves, the error between them and the actual trajectory is too large. As the robot moves incrementally, the errors of every frame accumulate, and these accumulated errors will eventually lead to the collapse of the entire SLAM system. The reason is that the traditional method simply connects the pose nodes between the keyframes. They are composed of a segment of broken lines, which makes it difficult to ensure the smoothness of the robot's trajectory. The properties of the

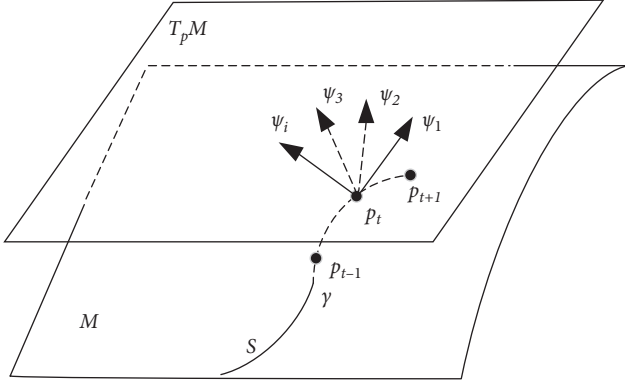


FIGURE 3: Schematic diagram of manifold tangent space.  $S$  represents the motion trajectory of the robot,  $p$  represents a certain pose node on the trajectory, and  $\psi_1, \psi_2, \dots, \psi_i$ , respectively, correspond to different tangent vectors in the manifold tangent space, indicating different robot motion directions.

differentiable manifold used in this paper can be effectively ensure the smoothness of robot movement.

In the Frenet framework, each node on the trajectory represents the pose of keyframe. Since the shape of the robot's motion trajectory is closely related to the robot's motion speed and the rotation angle, we judge the characteristics of the robot's motion trajectory based on the motion speed of the adjacent keyframe and the curvature of the composition trajectory. The specific methods are as follows.

For any two adjacent pose nodes  $P(t)$  and  $P(t + \Delta t)$  on the robot motion trajectory  $S$  (as shown in Figure 4), the corresponding binormal vectors of the two are  $B(t)$  and  $B(t + \Delta t)$ , respectively, and  $\Delta\varphi$  is the angle between them. The torsion rate is expressed as

$$\tau = \lim_{\Delta t \rightarrow 0} \frac{1}{|\Delta t|} |B(t + \Delta t) - B(t)|, \quad (4)$$

thanks to  $|B(t + \Delta t) - B(t)| = 2|\sin \Delta\varphi/2|$ , when  $\Delta t \rightarrow 0$ ,  $\Delta\varphi \rightarrow 0$ , so  $\sin(\Delta\varphi/2)/(\Delta\varphi/2) \rightarrow 1$ , and the above equation is equivalent to

$$|\tau| = \lim_{\Delta t \rightarrow 0} \left| \frac{\Delta\varphi}{\Delta t} \right|. \quad (5)$$

$|\tau|$  in (6) describes the degree of distortion of the robot motion trajectory  $S$ , which is closely related to the robot motion speed. The robot moves at different speeds at different times. When the speed is faster, the value of  $\tau$  is larger, vice versa. Specifically, when the value of  $\tau$  is smaller, the robot moves at a slower speed, and the trajectory of the robot composed of keyframe poses can only achieve the  $C_0$  property and cannot reach the degree of smoothness. When the value of  $\tau$  is larger, the robot moves faster, and the robot motion trajectory composed of keyframe pose nodes can ensure not only  $C_0$  property but also  $C_1$  property, which can ensure the continuous smoothness of the robot's motion.

However, since the motion of the robot in the real scene is complex and changeable, the motion trajectory characteristics of the robot cannot be well described only according to the speed of the robot. For example, when the robot is

moving with a relatively large corner change, the keyframe speed at this time is sometimes fast or slow, and it is hard to accurately describe the keyframe pose at this moment according to the speed. However, considering that the robot motion trajectory is also related to the curvature composed of the keyframe pose nodes, we can also add a standard quantity describing the robot motion trajectory: curvature, so as to make the analysis of the robot motion trajectory characteristics more reasonable.

Given two adjacent pose nodes  $P(t)$  and  $P(t + \Delta t)$  on the trajectory curve  $S$  of the robot, their corresponding unit tangent vectors are  $T(t)$  and  $T(t + \Delta t)$ , respectively,  $\Delta\theta$  is the angle between them, and the curvature expression is

$$\tau = \lim_{\Delta t \rightarrow 0} \frac{1}{|\Delta t|} |T(t + \Delta t) - T(t)|, \quad (6)$$

owing to  $|T(t + \Delta t) - T(t)| = 2|\sin \Delta\theta/2|$ ; when  $\Delta t \rightarrow 0$ ,  $\Delta\theta \rightarrow 0$ ,  $\sin(\Delta\theta/2)/(\Delta\theta/2) \rightarrow 1$ , and the above equation is equivalent to

$$\tau = \lim_{\Delta t \rightarrow 0} \left| \frac{\Delta\theta}{\Delta t} \right|. \quad (7)$$

$\kappa$  in formula (8) describes the degree of curvature of the robot's motion trajectory  $S$ . When it is difficult to judge the motion characteristics of the keyframe pose nodes at the corners based on the speed, we reasonably classify them according to the curvature value of their constituent curves. More specifically, the robot motion trajectory is analyzed according to the robot motion speed  $\tau$  and the curvature  $\kappa$  composed of the keyframe pose nodes. Then, according to the analysis results, it is analyzed to select the appropriate trajectory shape between keyframes. According to their different characteristics, the common trajectories of robot can be roughly divided into parabola, hyperbola, sine-cosine curve, etc. How to choose the optimal robot motion trajectory is also the problem that we need to discuss below.

The shape of the robot's trajectory in space can be seen as the intersection of two curved surfaces, namely:

$$\begin{cases} F_1(x, y, z) = 0 \\ F_2(x, y, z) = 0 \end{cases}, \quad (8)$$

where  $F_i(x, y, z)$  represents the  $i$ -th surface equation. This paper uses simple to complex ideas to solve this problem. First, we project it to the xoy plane, the intersection line of the two plane  $F_1(x, y, z) = 0$  and the plane  $F_2(x, y, z) = 0$  constitute the trajectory of the robot. In this plane, we study the shape of the robot's trajectory and give its implicit expression, namely,

$$F(x, y) = \alpha_{(n,0)}x^n + \alpha_{(0,n)}y^n + \dots + \alpha_{(i,j)}x^i y^j + \dots + \alpha_{(0,0)}, \quad (9)$$

where  $F(x, y)$  is the trajectory of the robot in the xoy plane,  $\alpha_{(n,0)}$  is a binary representation, and it represents the coefficient before the polynomial. The coefficients  $\alpha_{(n,0)}, \alpha_{(0,n)}, \dots, \alpha_{(i,j)}$  can be used to calculate the corresponding trajectory shape of the robot, which can be



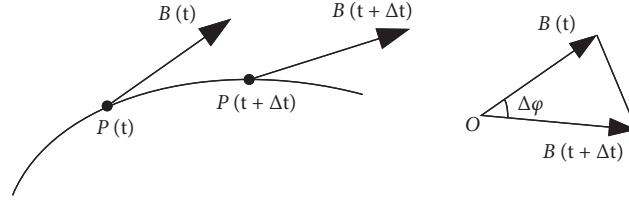


FIGURE 4: Schematic diagram of torsion rate.  $P(t)$  and  $P(t + \Delta t)$  represent adjacent keyframe pose nodes on the robot motion trajectory  $S$ , the corresponding binormal vectors of the two are  $B(t)$  and  $B(t + \Delta t)$ , respectively, and  $\Delta\varphi$  is the angle between them.

similarly projected on the  $yoz$  plane and  $xoz$  plane, and the trajectory equation of the movement in other planes can be obtained.

When the robot moves in the  $xyz$  plane, its trajectory form can also be given, namely,

$$F(x, y, z) = \beta_{(n,0,0)}x^n + \beta_{(0,n,0)}y^n + \beta_{(0,0,n)}z^n + \dots + \beta_{(i,j,k)}x^i y^j z^k + \dots + \beta_{(0,0,0)}, \quad (10)$$

where  $F(x, y, z)$  is the implicit expression of the robot's trajectory in the  $xyz$  plane.  $A_{(i,j,k)}$  is the triplet representation, representing the coefficient before the polynomial. The coefficients  $\beta_{(n,0,0)}, \beta_{(0,n,0)}, \beta_{(0,0,n)}, \dots, \beta_{(i,j,k)}$  can be used to calculate the trajectory shape of the robot. At the same time, different trajectory structures are selected and their fitting coefficients are calculated to determine the best trajectory of the robot, so that the fitted trajectory of the robot matches the real trajectory as much as possible.

Since the cubic B-spline curve is a smooth curve and has continuous second-order derivative, it is accepted by various disciplines because of its good smoothness. Therefore, this method is also used in this paper to make the robot trajectory obtained through keyframe pose nodes as smooth as possible. The detailed process is as follows.

Given  $n + 1$  control points  $F_i (i = 0, 1, \dots, n)$ , the corresponding expression of the B-spline curve is as follows:

$$C(u) = \sum_{i=0}^n P_i N_{i,k}(u). \quad (11)$$

In formula (12),  $N_{i,k}(u)$  is the harmonic function; because it exists as the basis of the function, it is also called the basis function, and according to the recursive formula, it can be equivalent to

$$N_{i,k}(u) = \frac{u - m_t}{m_{t+k-1} - m_t} N_{i,k-1}(u) + \frac{m_{t+k} - u}{m_{t+k} - m_{t+1}} N_{i+1,k-1}(u). \quad (12)$$

When  $u \in [ti, ti + 1)$ ,  $N_{i,1} = 1$ ; otherwise,  $N_{i,1} = 0$ . Among them,  $ti$  represents the node value, and the node used is a nondecreasing sequence, which constitutes the node vector of the  $K$ -order B-spline function.

It should be noted that in our keyframe-based robot motion trajectory processing, it may not support us to perform interpolation or fitting processing due to insufficient numbers. At this time, we need to manually select some

frames and insert them. We put these artificially selected ones which are called quasi-keyframes, and the selection principles of quasi-keyframes are as follows:

- (1) Analyze it in the time domain. Since the frame rate of the camera is 30, if the system runs for a long time and there is no keyframe for a long time, we artificially select several frames as quasi-keyframes.
- (2) Analyze it from the spatial domain. If there is no common viewing area between two keyframes, it indicates that there may be tracking loss or violent motion at this time, and several frames need to be selected as quasi-keyframes.

Finally, this paper needs to consider the smoothness of the joints of the trajectory segments after fitting. For this reason, we obtain the first-order and second-order derivatives at the joints of each trajectory segment, so that the second-order derivatives are equal to ensure the smoothness and stability of the joints of each trajectory curve.

## 4. Experiment

In order to evaluate the performance of the loop closure detection based on the differentiable manifold proposed in this paper, we conducted a series of experiments and compared it with the most popular methods. In order to make the measurement result more accurate, we define five error indicators to evaluate the difference between the robot motion trajectory and the real trajectory obtained by the method in this paper: maximum error, minimum error, average error, absolute trajectory error, and relative trajectory error, and give qualitative analysis and quantitative analysis results to prove the robustness of our method.

**4.1. Datasets.** The evaluation dataset in this paper is from TUM benchmark tool [48], which contains two public image sets `rgb_d_dataset_freiburg1_room` and `rgb_d_dataset_freiburg3_long_office_household`. These two datasets can

objectively evaluate the performance of the method proposed in this paper. The details are as follows:

- (a) `rgbd_dataset_freiburg1_room`: this dataset is taken along the trajectory of the office. It starts with four tables and then moves around the outer wall of the room until the loop is closed. This dataset is very suitable for evaluating the ability of the SLAM system to process loop closure detection. The dataset consists of 1362 pictures with resolution of  $640 * 480$ , and the real track length is 15.989 m.
- (b) `rgbd_dataset_freiburg3_long_office_household`: this dataset is moved by ASUS Xtion sensors along a large circle, passing through home and office scenes with many textures and structures. The end point of the trajectory overlaps with the start point, so there is a large loop. The dataset consists of 2585 pictures with resolution of  $640 * 480$ , and the true track length is 21.455 m.

**4.2. Experimental Results and Analysis.** (a) In the given dataset `freiburg1_room`, we first use the image matching algorithm to filter out 180 keyframe pose nodes from 1362 images and then analyze the motion trajectory characteristics of the robot according to the robot motion speed  $\tau$ . By calculating the motion velocity of adjacent frame, robot movement speed  $\tau$  is concentrated between 0.0593 m/s and 1.3648 m/s, and then we classify the movement speed between keyframes without much change into one category; in other words, the movement speed between keyframes is within 0.33. Next step is to calculate the curvature  $\kappa$  between the keyframe nodes of the track segment with the same trajectory characteristics, where the curvature value threshold is set to 0.75. If the calculated curvature of a keyframe node is greater than this threshold, it is considered that this corresponds to the critical point with a relatively large corner amplitude, and the pose nodes at this time are merged into the next type of trajectory segment, and so on, until all the keyframe pose nodes with the same motion trajectory characteristics are found and the division is finished. In the processed keyframe segment, finding the optimal orientation  $\psi$ , the real-time motion of the robot here is affected by multiple frames, so we analyze the influence of several frames to filter out the optimal orientation. After the above steps, we study the  $y$ -axis error and  $z$ -axis error with the same trajectory characteristics in each segment.

For the robot motion trajectory in the  $xoy$  plane, we select  $N$  keyframe pose nodes in each segment of the robot motion trajectory for processing. The details are as follows. The cubic  $B$ -spline interpolation method is firstly used to make the robot motion trajectory pass the keyframe pose node, combining the fitting method to get the expression that best fits the robot motion shape. In order to make the obtained trajectory consistent with the real situation as much as possible, we consider using  $N$  keyframe pose nodes ( $N=3, 4, 5$ ) to fit the trajectory and use the fitted trajectory equation to find the corresponding  $y$  value. This paper

qualitatively and quantitatively analyze the fitted robot trajectory, and the result output by our method is compared with the ORB SLAM2 and the ground truth. The following shows the trajectory visualization diagram after the local optimization and the trajectory visualization diagram after the global optimization.

Since this paper classifies the robot's motion trajectory into one category according to the similar motion characteristics, there will be several trajectory segments, and only a few are given randomly here. From Figures 5(a), 5(c), and 5(e), it can be concluded that when the robot movement speed  $\tau$  is between 0.3845 m/s and 0.7014 m/s and the curvature value  $\kappa$  is between 0.002 and 0.04, the parabolic trajectory fitted by the three keyframe pose nodes is the most consistent with the real robot trajectory, that is,  $F(x, y) = \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 + \alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , where the corresponding  $\alpha_{(i,j)}$  are 1.238, 0.07842, 1, and  $-0.5163$ , respectively. From Figures 5(b), 5(d), and 5(f), it can be concluded that when the robot movement speed  $\tau$  is between 0.2214 m/s and 0.4508 m/s and the curvature value  $\kappa$  is between 0.005 and 0.018, the one-dimensional cubic equation trajectory fitted by the four keyframe pose nodes is the most consistent with the real robot trajectory, that is,  $F(x, y) = \alpha_{(3,0)}x^3 + \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 + \alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , where the corresponding  $\alpha_{(i,j)}$  are  $-1.779$ , 0.4036, 0.7173, 1, and  $-17.71$ , respectively. This paper also uses  $N$  keyframe pose nodes ( $N=3, 4, 5$ ) to process the robot's global motion trajectory and uses the obtained trajectory equation to draw the robot's global motion trajectory. The experimental results are shown in Figure 6; it can be seen from Figures 6(a), 6(c), and 6(e) that when the robot movement speed  $\tau$  is between 0.1331 m/s and 0.4580 m/s and the curvature value  $\kappa$  is between 0.0071 and 0.0265, the parabolic trajectory fitted by the four keyframe pose nodes is most consistent with the real trajectory, that is,  $F(x, y) = \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 + \alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , the corresponding  $\alpha_{(i,j)}$  are 1.336,  $-0.2775$ , 1, and  $-0.7645$ , respectively. Through experiments, we found that in more special cases when the robot movement speed  $\tau$  is between 0.1006 and 0.1770 and the curvature value  $\kappa$  is between 0.0548 and 0.0586, simple parabola and one-dimensional cubic equations are difficult to match the real trajectory, but a simple sine function is used,  $F(x, y) = a * \sin(b * x + c)$ , corresponding to  $a = 2.007$ ,  $b = 104.6$ , and  $c = -32.08$ .

In the same way, for the more complex robot motion trajectory in three-dimensional space, considering the error in the  $z$ -axis direction, we select  $M$  keyframe pose nodes ( $M=6, 7, 8$ ) in each segment of the robot motion trajectory. We first use the spline interpolation method to make the robot's motion trajectory pass the keyframe pose nodes and then combine the fitting method to obtain the expression that best matches the robot's motion shape. In order to make the obtained trajectory consistent with the real situation as much as possible, we use different motion shapes for processing. The global trajectory result can also be seen from Figures 6(b), 6(d), and 6(f). From the global trajectory error evaluation, it can be concluded that  $M=8$  keyframe poses is relatively small. The specific trajectory shape is  $F(x, y, z) = \beta_{(2,0,0)}x^2 + \beta_{(0,2,0)}y^2 + \beta_{(0,0,1)}z^1 + \beta_{(1,1,0)}$



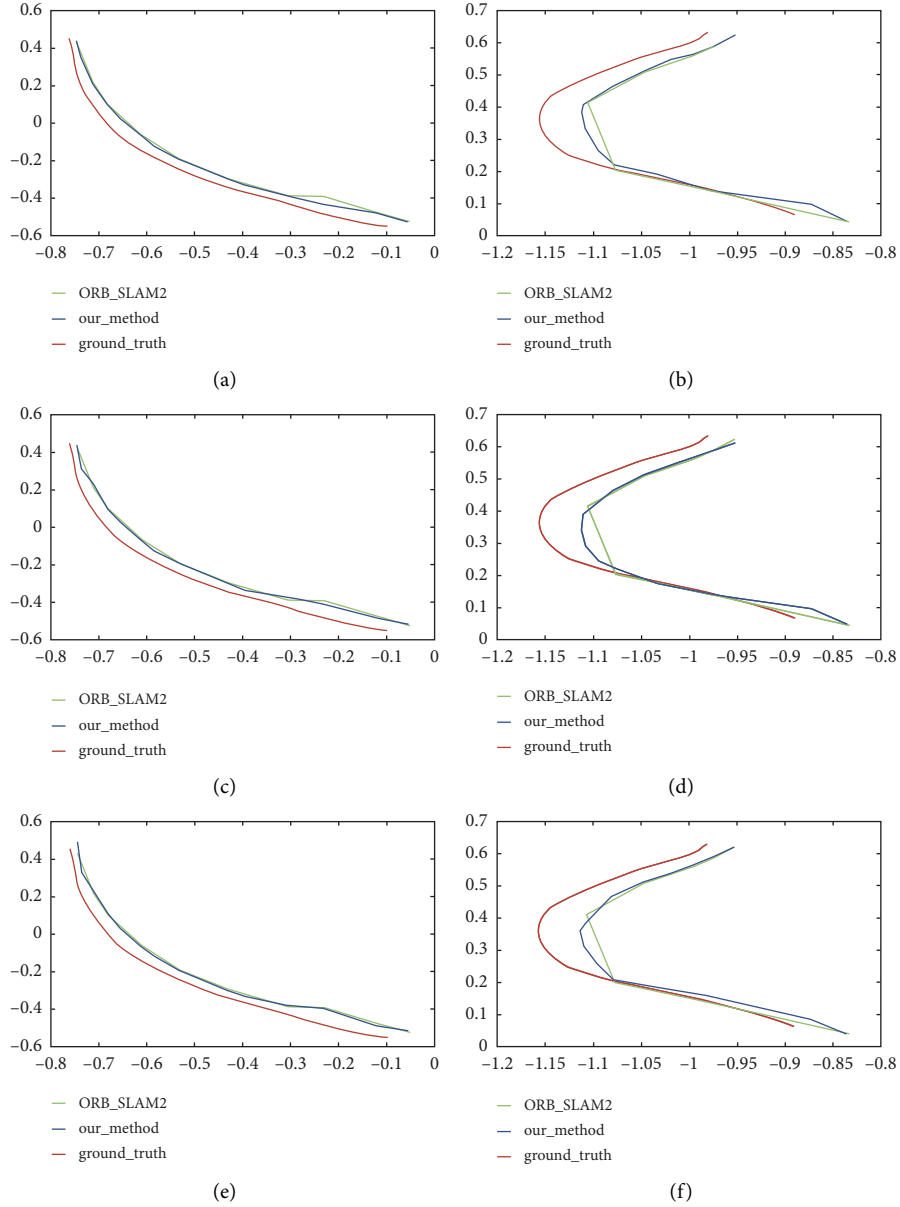


FIGURE 5: A partial experimental comparison diagram of the robot motion trajectory after using the obtained motion trajectory equation and the real trajectory. Among them, the red curve represents the true trajectory value of the robot, the blue curve represents the trajectory value of the robot obtained by the method in this paper, and the green curve represents the trajectory value of the robot in the ORB\_SLAM2 method. (a)  $N1 = 3$ . (b)  $N2 = 3$ . (c)  $N3 = 4$ . (d)  $N4 = 4$ . (e)  $N5 = 5$ . (f)  $N6 = 5$ .

$x^1 y^1 + \beta_{(0,1,0)} y^1 + \beta_{(1,0,0)} x^1 + \beta_{(0,0,0)}$ , where the corresponding  $\beta_{(i,j,k)}$  are  $-0.06873$ ,  $-0.07766$ ,  $1$ ,  $0.02006$ ,  $0.1498$ ,  $-0.03972$ , and  $1.421$ . The quantitative analysis results of the robot's global motion trajectory error are given in Tables 1 and 2. It can be found that our method can achieve the same level as the ORB\_SLAM2 method as a whole and is better than ORB\_SLAM2 in some areas. Compared with the robot motion trajectory obtained by using the image matching algorithm, the error between the robot motion trajectory and the real trajectory after we use the differentiable manifold is smaller.

(b) In the given dataset freiburg3\_long\_office\_household, we first use an image matching algorithm to filter out

204 keyframe pose nodes from 2585 images and then analyze its motion trajectory characteristics according to the robot motion speed  $\tau$ . By calculating the motion velocity of adjacent frame, robot movement speed  $\tau$  is concentrated between  $0.0372$  m/s and  $0.4292$  m/s, and then we attribute the close motion speed between keyframes to one category, where little change means that the motion speed error between keyframes is within  $0.25$ . Next step is to calculate the curvature  $\kappa$  between the keyframe nodes of the track segment with the same trajectory characteristics, where the curvature value threshold is set to  $0.42$ . If the calculated curvature of a keyframe node is greater than this threshold, it is considered that this corresponds to the critical point with a

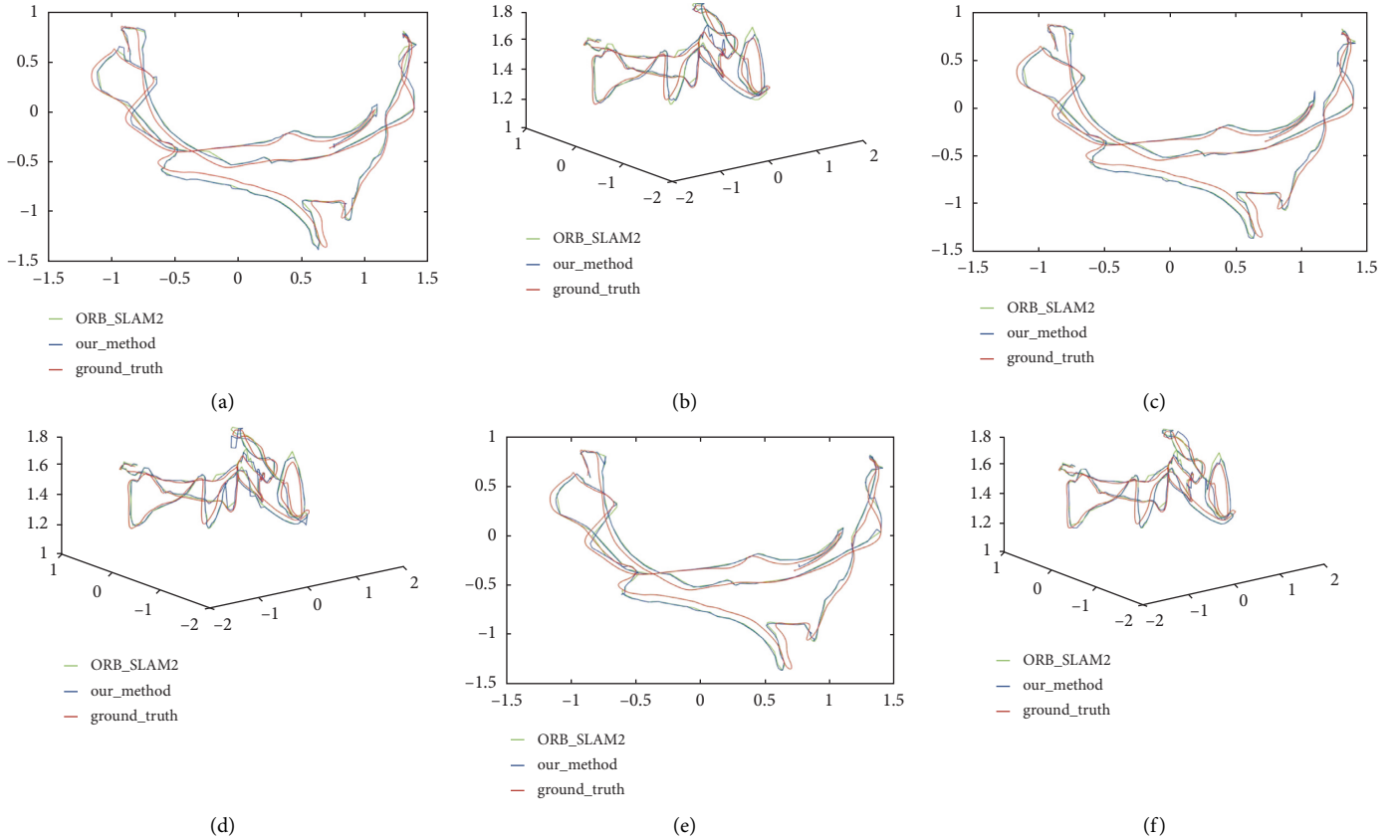


FIGURE 6: The left image shows the global comparison diagram of the robot motion trajectory and the real trajectory in the xoy plane after using the obtained motion trajectory equation. The figure on the right shows the global comparison of the robot's motion trajectory and the real trajectory in the xyz plane after using the obtained motion trajectory equation. Among them, the red curve represents the true trajectory value of the robot, the blue curve represents the trajectory value of the robot obtained by the method in this paper, and the green curve represents the trajectory value of the robot in the ORB\_SLAM2 method. (a)  $N1 = 3$ . (b)  $M1 = 6$ . (c)  $N2 = 4$ . (d)  $M2 = 7$ . (e)  $N3 = 5$ . (f)  $M3 = 7$ .

relatively large corner amplitude, and the pose nodes at this time are merged into the next type of trajectory segment, and so on, until all the keyframe pose nodes with the same motion trajectory characteristics are found and the division is finished. We find the optimal orientation  $\psi$  in the processed keyframe segment, while the real-time motion of the robot here is affected by multiple frames, so we analyze the influence of several frames to filter out the optimal orientation. After the above steps, we also study from the two directions of  $y$ -axis error and  $z$ -axis error in each segment with the same trajectory characteristics.

For the robot motion trajectory in the  $xoy$  plane, we select  $N$  keyframe pose nodes in each segment of the robot motion trajectory for processing, and the specific steps are described in 4.2.(a). The following shows the trajectory visualization diagram after the local optimization and the trajectory visualization diagram after the global optimization.

From Figures 7(a), 7(c), and 7(e), it can be concluded that when the robot movement speed  $\tau$  is between 0.2017 m/s and 0.3832 m/s and the curvature value  $\kappa$  is between 0.0357 and 0.0512, the parabolic trajectory fitted by the four keyframe pose nodes is most consistent with the real robot trajectory, that is,  $F(x, y) = \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 +$

$\alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , and the corresponding  $\alpha_{(i,j)}$  are  $-1.695$ ,  $3.683$ ,  $1$ , and  $-0.8821$ , respectively. From Figures 6(b), 6(d), and 6(f), it can be concluded that when the robot movement speed  $\tau$  is between 0.2670 m/s and 0.2774 m/s and the curvature value  $\kappa$  is between 0.0011 and 0.0032, the one-dimensional cubic equation trajectory fitted by the five keyframe pose nodes and the real robot trajectory is the most consistent with the real trajectory, that is,  $F(x, y) = \alpha_{(3,0)}x^3 + \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 + \alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , and the corresponding  $\alpha_{(i,j)}$  are 1.966, 8.281, 15.32, 1, and 5.843, respectively. This paper also uses  $N=3,4,5$  keyframe pose nodes to process the robot's global motion trajectory and uses the obtained trajectory equation to draw the robot's global motion trajectory. The experimental results are shown in Figure 8; it can be seen from the left of Figure 8 that when the robot movement speed  $\tau$  is between 0.1998 m/s and 0.3035 m/s and the curvature value  $\kappa$  is between 0.0071 and 0.0265, the parabolic trajectory fitted by the four keyframe pose nodes is most consistent with the real trajectory,  $F(x, y) = \alpha_{(2,0)}x^2 + \alpha_{(1,0)}x^1 + \alpha_{(0,1)}y^1 + \alpha_{(0,0)}$ , and corresponding  $\alpha_{(i,j)}$  are  $-2.943$ ,  $-19.3$ ,  $1$ , and  $-31.76$ , respectively. Through experiments, we found that in more special cases when the robot movement speed  $\tau$  is between 0.1006 m/s and 0.1770 m/s and the curvature value  $k$  is between 0.1271

TABLE 1: rgbd\_dataset\_freiburg1\_room dataset y-axis direction error.

Methods	Maximum error (m)	Minimum error (m)	Average error (m)	ATE (m)	RPE (m)	Keyframe ( $N$ )
ORB_SLAM2	0.0900	0.0008	0.0670	0.0557	0.0125	180
	0.0888	0.0003	0.0296	0.0365	0.1045	3
our_method	0.1417	0.0001	0.0329	0.0410	0.0089	4
	0.1332	0.0001	0.0335	0.0423	0.1184	5

The units in the table are meters.

TABLE 2: rgbd\_dataset\_freiburg1\_room dataset z-axis direction error.

Methods	Maximum error (m)	Minimum error (m)	Average error (m)	ATE (m)	RPE (m)	Keyframe ( $M$ )
ORB_SLAM2	0.0773	0.0001	0.0144	0.0193	0.0135	180
	0.0720	0.0001	0.0130	0.0173	0.0106	6
our_method	0.2073	0.0002	0.0198	0.0341	0.0232	7
	0.0697	0.0001	0.0153	0.0215	0.0113	8

The units in the table are meters.

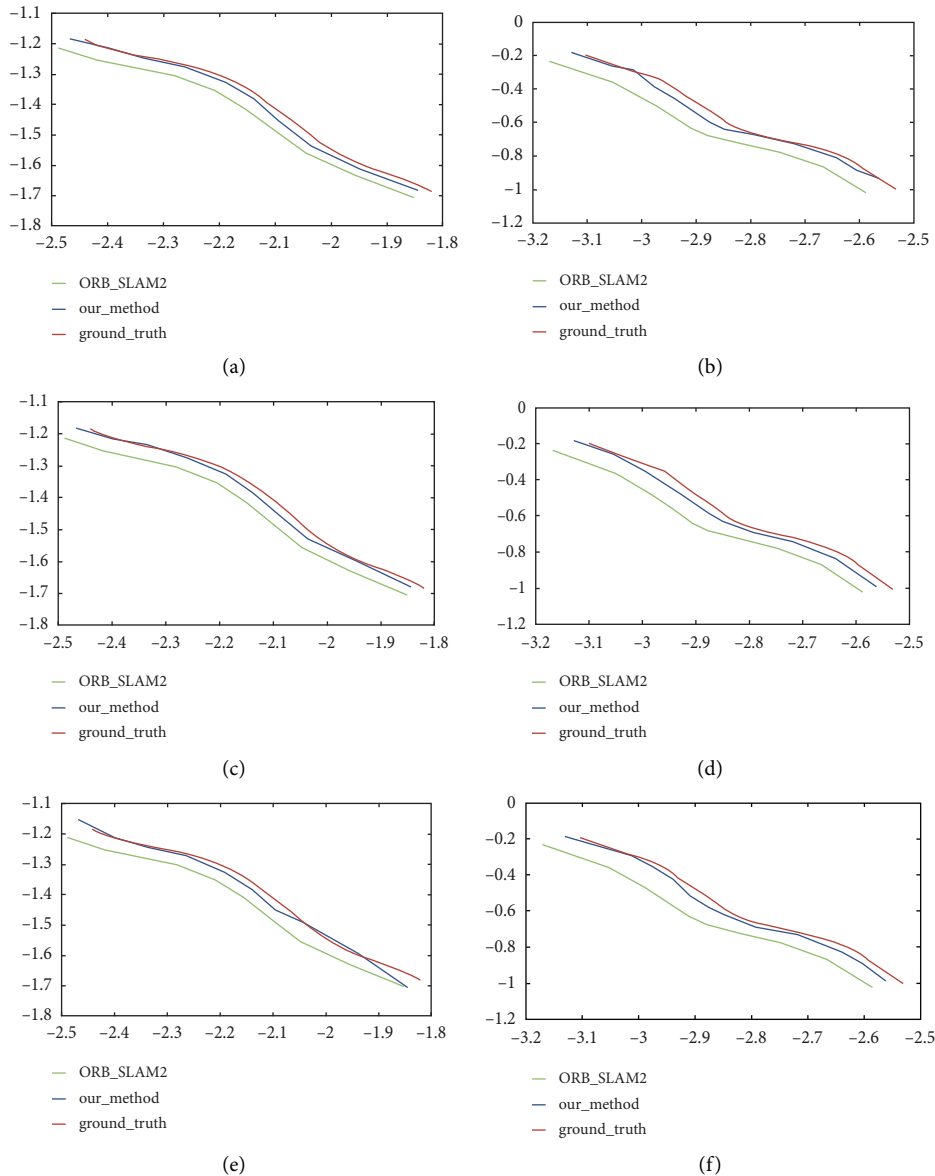


FIGURE 7: A partial experimental comparison diagram of the robot motion trajectory after using the obtained motion trajectory equation and the real trajectory. Among them, the red curve represents the true trajectory value of the robot, the blue curve represents the trajectory value of the robot obtained by the method in this paper, and the green curve represents the trajectory value of the robot in the ORB\_SLAM2 method. (a)  $N_1 = 3$ . (b)  $N_2 = 3$ . (c)  $N_3 = 4$ . (d)  $N_4 = 4$ . (e)  $N_5 = 5$ . (f)  $N_6 = 5$ .

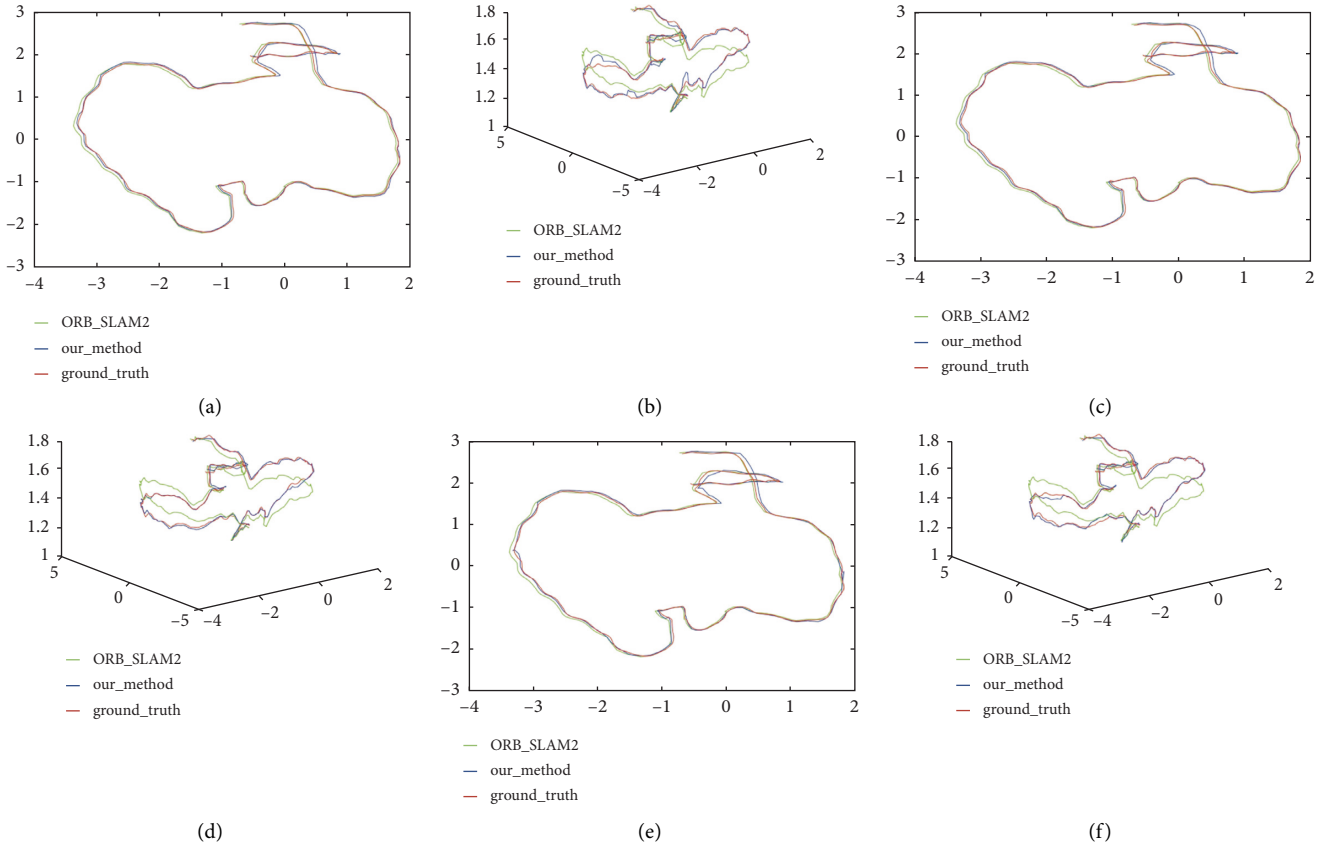


FIGURE 8: The global comparison of the robot motion trajectory and the real trajectory in the xoy plane after using the obtained motion trajectory equation. The figure on the right shows the global comparison of the robot's motion trajectory and the real trajectory in the xyz plane after using the obtained motion trajectory equation. Among them, the red curve represents the true trajectory value of the robot, the blue curve represents the trajectory value of the robot obtained by the method in this paper, and the green curve represents the trajectory value of the robot in the ORB\_SLAM2 method. (a)  $N1 = 3$ . (b)  $M1 = 6$ . (c)  $N2 = 4$ . (d)  $M2 = 7$ . (e)  $N3 = 5$ . (f)  $M3 = 8$ .

TABLE 3: freiburg3\_long\_office\_household dataset y-axis direction error.

Methods	Maximum error (m)	Minimum error (m)	Average error (m)	ATE (m)	RPE (m)	Keyframe ( $N$ )
ORB_SLAM2	0.1466	0.0001	0.0203	0.0234	0.0183	204
	0.1401	0.0002	0.0178	0.0231	0.0196	3
our_method	0.1063	0.0001	0.0174	0.0238	0.0185	4
	0.1675	0.0005	0.0272	0.03997	0.0186	5

The units in the table are meters.

TABLE 4: freiburg3\_long\_office\_household dataset z-axis direction error.

Methods	Maximum error (m)	Minimum error (m)	Average error (m)	ATE (m)	RPE (m)	Keyframe ( $M$ )
ORB_SLAM2	0.1466	0.0001	0.0622	0.0780	0.1068	204
	0.1727	0.0001	0.0364	0.0262	0.1691	6
our_method	0.1490	0.0001	0.0154	0.0332	0.0141	7
	0.1564	0.0002	0.0146	0.0314	0.0133	8

The units in the table are meters.

and 0.2248, simple parabola and one-dimensional cubic equations are difficult to match the real trajectory, but use  $y = p/(x^2 + q1x + q2)$ , and the shape trajectory is more consistent, corresponding to  $p = 1.253$ ,  $q1 = 6.981$ , and  $q2 = -8.05$ .

In the same way, for the more complex robot motion trajectory in three-dimensional space, considering the error in the z-axis direction, we select  $M$  keyframe pose nodes ( $M = 6, 7, 8$ ) in each segment of the robot motion trajectory. We first use the spline interpolation method which makes

the robot's motion trajectory pass the keyframe pose node and then combine the fitting method to obtain the expression that best matches the robot's motion shape. In order to make the obtained trajectory consistent with the real situation as much as possible, we use different motion shapes for processing. The experimental results are shown on the right side of Figure 8. From the global trajectory error evaluation, it can be concluded that  $M = 7$  keyframe poses is relatively small. The specific trajectory shape is  $F(x, y, z) = \beta_{(2,0,0)}x^2 + \beta_{(0,2,0)}y^2 + \beta_{(0,0,1)}z^1 + \beta_{(1,1,0)}x^1y^1 + \beta_{(0,1,0)}y^1 + \beta_{(1,0,0)}x^1 + \beta_{(0,0,0)}$ , where the corresponding  $\beta_{(i,j,k)}$  are  $-4.73, 0.482, 1, 1.685, -1.523, 10.77,$  and  $-4.589,$  respectively. The quantitative analysis results of the robot's global motion trajectory error are given in Tables 3 and 4. It can be found that our method can achieve the same level as the ORB SLAM2 method as a whole and is better than ORB\_SLAM2 in some areas. Compared with the robot motion trajectory obtained by using the image matching algorithm, the error between the robot motion trajectory and the real trajectory after we use the differentiable manifold is smaller.

## 5. Conclusion

In this paper, we propose a loop closure detection method based on differentiable manifold. The motion trajectory is divided according to the motion characteristics of the robot, the curvature and torsion of the curve composed of several nodes are calculated based on the manifold tangent space, curve interpolation and fitting methods are used to achieve the approximation of the real-time motion trajectory of the robot, and finally the smooth curve of the robot's motion trajectory is obtained, thereby reducing the cumulative error and system error and improving the calculation accuracy of the robot's real-time pose and trajectory. Compared with the robot trajectory obtained by only using the image matching algorithm, the robot trajectory obtained in this paper is more stable and smooth, which effectively reduces the cumulative error of the entire SLAM system and improves the accuracy of loop closure detection.

## Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was partially supported by the Shanghai Alliance Program (LM2019043) and the Open Project of State Key Laboratory of Pattern Recognition (201600033).

## References

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [2] D. Ribas, P. Ridao, and J. Neira, *Simultaneous Localization and Mapping, Underwater SLAM for Structured Environments Using an Imaging Sonar*, Springer, Berlin, Germany, 2010.
- [3] S. Lowry, N. Sunderhauf, P. Newman et al., "Visual place recognition: a survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [4] E. Garcia-Fidalgo and A. Ortiz, *Methods for Appearance-Based Loop Closure Detection: Applications to Topological Mapping and Image Mosaicking*, Springer Tracts in Advanced Robotics, Springer, Berlin, Germany, 2018.
- [5] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: direct sparse odometry with loop closure," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, Spain, 21 October 018.
- [6] M. Cummins and P. Newman, "Fab-map: probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [7] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.
- [8] D. Galvez-López and J. Tardós, "Real-time loop detection with bags of binary words," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, San Francisco, CA, USA, 25–30 Sep. 2011.
- [9] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [10] J. Engel, T. Schops, and D. Cremers, "Lsd-slam: large-scale direct monocular slam," in *Proceedings of the European Conference on Computer Vision*, Zurich, Switzerland, 6 December 2014.
- [11] E. G. Karakasis, A. Amanatiadis, A. Gasteratos, and S. A. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model," *Pattern Recognition Letters*, vol. 55, pp. 22–27, 2015.
- [12] N. Kejrival, S. Kumar, and T. Shibata, "High performance loop closure detection using bag of word pairs," *Robotics and Autonomous Systems*, vol. 77, pp. 55–65, 2016.
- [13] M. J. Milford and G. F. Wyeth, "Seqslam: visual route-based navigation for sunny summer days and stormy winter nights," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA, 14 May 2012.
- [14] S. M. Siam and H. Zhang, "Fast-seqslam: a fast appearance based place recognition algorithm," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 29 MAY 2017.
- [15] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Fast loop-closure detection using visual-word-vectors from image sequences," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 62–82, 2018.
- [16] M. Abdollahyan, S. Cascianelli, E. Bellocchio et al., "Visual localization in the presence of appearance changes using the partial order kernel," in *Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 3 September 2018.
- [17] P. Hansen and B. Browning, "Visual place recognition using hmm sequence matching," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 14 September 2014.
- [18] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "Assigning visual words to places for loop closure detection," in

- Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 21 May 2018.
- [19] J. Chen, J. Li, X. Yang, G. Shen, and Y. Gao, "A compact loop closure detection based on spatial partitioning," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, China, 2 June 2017.
- [20] P. Valigi, E. Bellocchio, S. Cascianelli, G. Costante, and T. Ciarfuglia, "Robust visual semi-semantic loop closure detection by a covisibility graph and cnn features," *Robotics and Autonomous Systems*, vol. 92, pp. 53–65.
- [21] R. Finman, L. Paull, and J. J. Leonard, "Toward Object-Based Place Recognition in Dense Rgb-D Maps," [https://people.csail.mit.edu/lpaull/publications/Finman\\_ICRAW\\_2015.pdf](https://people.csail.mit.edu/lpaull/publications/Finman_ICRAW_2015.pdf).
- [22] S. An, G. Che, F. Zhou, X. Liu, X. Ma, and Y. Chen, "Fast and Incremental Loop Closure Detection Using Proximity Graphs," <https://arxiv.org/abs/2010.11703>.
- [23] Z. Liang and C. Wang, "A semi-direct monocular visual slam algorithm in complex environments," *Journal of Intelligent and Robotic Systems*, vol. 101, no. 1, p. 25, 2021.
- [24] P. Siva, C. Russell, T. Xiang, and L. Agapito, "Looking beyond the image: unsupervised learning for object saliency and detection," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, Portland, OR, USA, 23 June 2013.
- [25] T. H. N. Le, K. Luu, C. Zhu, and M. Savvides, "Semi self-training beard/moustache detection and segmentation simultaneously," *Image and Vision Computing*, vol. 58, pp. 214–223, 2017.
- [26] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) - recurrent multimodal interaction for referring image segmentation*, pp. 1280–1289, Portland, OR, USA, 22 October 2017.
- [27] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 18 June 2018.
- [28] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Fusion and binarization of cnn features for robust topological localization across seasons," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, Daejeon, Korea (South), 9 October 2016.
- [29] G. Xiang and Z. Tao, "Unsupervised learning to detect loops using deep neural networks for visual slam system," *Autonomous Robots*.
- [30] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, Berlin Germany, 11 March 2015.
- [31] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with cnns for frame-to-frame ego-motion estimation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, 2016.
- [32] Y. Xia, L. Jie, Q. Lin, Y. Hui, and J. Dong, "An evaluation of deep learning in loop closure detection for visual slam," in *Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Exeter, UK, 21 June 2018.
- [33] X. Gao and T. Zhang, "Loop closure detection for visual slam systems using deep neural networks," in *Proceedings of the 2015 34th Chinese Control Conference (CCC)*, Huddersfield, UK, 7 September 2015.
- [34] H. Wang, C. Wang, and L. Xie, "Intensity scan context: coding intensity and geometry relations for loop closure detection," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 31 May 2020.
- [35] P. Geneva, J. Maley, and G. Huang, "An efficient schmidt-ekf for 3d visualinertial slam," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 15 June 2019.
- [36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [37] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," in *Proceedings of the 9th European conference on Computer Vision - Volume Part I*, Graz Austria, 7 May 2006.
- [38] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: an efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV 2011*, Barcelona, Spain, 6 November 2011.
- [39] 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*.
- [40] P. Protzel and N. Sunderhauf, "Brief-gist – closing the loop by simple means," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 25 September 2011.
- [41] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [42] N. Nourani-Vatani, P. V. K. Borges, J. M. Roberts, and M. V. Srinivasan, "On the use of optical flow for scene change detection and description," *Journal of Intelligent and Robotic Systems*, vol. 74, no. 3-4, pp. 817–846, 2014.
- [43] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows," in *Proceedings of the Conference on Advancements of Artificial Intelligence (AAAI)*, Québec, Canada, 27 July 2014.
- [44] P. L. Negre Carrasco, F. Bonin-Font, and G. Oliver-Codina, "Global image signature for visual loop-closure detection," *Autonomous Robots*, vol. 40, no. 8, pp. 1403–1417, 2016.
- [45] O. Guclu and A. B. Can, "Integrating global and local image features for enhanced loop closure detection in rgb-d slam systems," *The Visual Computer*, vol. 36, no. 5.
- [46] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Encoding the description of image sequences: a two-layered pipeline for loop closure detection," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), 9 October 2016.
- [47] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 7 October 2012.
- [48] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards life-long visual localization using an efficient matching of binary sequences from images," in *Proceedings of the IEEE International Conference on Robotics & Automation*, IEEE, Seattle, WA, USA, 26 May 2015.
- [49] A. Gawel, C. D. Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.



- [50] Y. Wang, Y. Qiu, P. Cheng, and X. Duan, "Robust loop closure detection integrating visual-spatial-semantic information via topological graphs and CNN features," *Remote Sensing*, vol. 12, no. 23, p. 3890, 2020.
- [51] P. Gao and H. Zhang, "Long-term loop closure detection through visual-spatial information preserving multi-order graph matching," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 06, pp. 10369–10376, 2020.
- [52] B. Triggs, *Bundle Adjustment – a Modern Synthesis*, Springer-Verlag.
- [53] T. Schps, T. Sattler, and M. Pollefeys, "Bad slam: bundle adjusted direct rgb-d slam," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 15 June 2020.