IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 1, 2021, accepted April 24, 2021, date of publication May 4, 2021, date of current version May 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077591

MS-SLAM: Motion State Decision of Keyframes for UAV-Based Vision Localization

YIMIN LUO, YONG LI, ZHITENG LI, AND FENG SHUANG, (Member, IEEE)
College of Electrical Engineering, Guangxi University, Nanning 530000, China

Corresponding authors: Yong Li (yongli@gxu.edu.cn) and Feng Shuang (fshuang@gxu.edu.cn)

This work was supported in part by the Guangxi Key Laboratory of Manufacturing System and Advanced Manufacturing Technology under Grant 20-065-40S005, and in part by the National Natural Science Foundation of China under Grant 61720106009 and Grant 61773359.

ABSTRACT To improve the robustness of UAVs in rotating and motion scenes, we propose a stereo simultaneous localization and mapping (SLAM) for UAVs with a new keyframe strategy that differs from current SLAM systems. Moreover, it has two strategies. 1) In the dominant strategy based on image quality, we filter and retain the strong feature points in each image frame by our own defined rules; this leads to longer survival time and attribute invariance in image tracking, and we save the image frames containing more strong feature points as keyframes. 2) In the secondary strategy based on motion state, we quantify the motion state during the UAV's motion (called the compound rotation amount), and characterize the intensity of the motion. Since we want the UAV to have better robustness in the rotating scene, this strategy generates keyframes when the compound rotation amount meets the threshold. These two strategies are used to cope with gentle motion scenes and rotational motion scenes, respectively. Thus, the insertion of our keyframes is determined by the motion state; our SLAM system is proposed based on the motion state (MS-SLAM). In the back-end part of the system we construct a new weighted cost function to optimize the pose. Finally, through comparison experiments on the public dataset EuRoc, we demonstrate that our algorithm is more advantageous than some current mainstream algorithms. In difficult sequences, our algorithm compares the absolute trajectory error with ORB-SLAM2, SVO+gtsam, and VINS-Mono, the absolute trajectory error of our algorithm can be reduced by 87%.

INDEX TERMS Feature point survival time, SLAM, motion state, keyframes selection.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) require recognition, localization, and obstacle avoidance navigation capabilities for autonomous missions; accurate localization is the basis for safe flight and navigation avoidance of UAVs [1], [2]. There are many solutions to the UAV localization problem, and one of them is the camera-based simultaneous localization and mapping (SLAM) method [3]. SLAM solves the problems of weak GPS signal, limited power, and load of UAVs, and it achieves localization using only the camera carried by the UAV. However, the visual localization algorithm for UAVs suffers from cumulative drift, and the accuracy is highly susceptible to the effects of lighting and other scene characteristics. Without relying on other hardware assistance, the algorithm can easily fail to track in scenes with motion, such as rotation; and such scenes become a challenge for SLAM algorithms. In order to improve the algorithm robustness, many scholars have conducted research on the front end of the algorithm (depth estimation, keyframes selection, feature extraction, and matching), back end of the algorithm, and loop closure detection part of the algorithm [4]–[9], and have achieved promising results. However, due to the limitations of the time and scenario in which the study was conducted, many algorithms have large errors or even tracking failures in UAV motion scenes. Therefore, in this paper, we propose an algorithm based on motion state decision keyframes for UAV motion scenes. The algorithm framework (Fig. 1) is based on that of ORB-SLAM2, and is divided into four parts: front end, back end, loop closing, and map building. After ORB feature point extraction, we filter out the strong feature points. These points participate in the tracking thread, and the strong feature points contained in each frame are the condition for us to evaluate whether the quality of the current frame meets the keyframe. Then, for the current camera pose, we calculate its compound rotation amount relative to the reference frame as a

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrahmane Lakas.

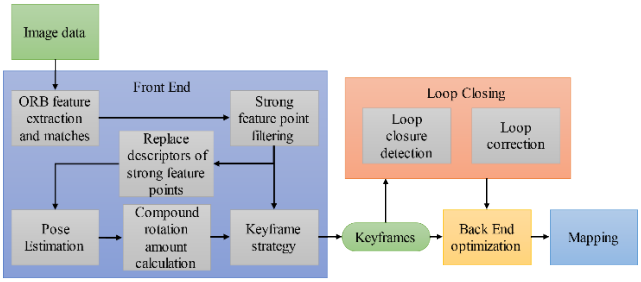


FIGURE 1. Algorithm framework.

basis for judging whether the current motion state needs to be added to the keyframe. In the back end, we set a weighted cost function based on the strong feature points and the compound rotation amount to better optimize the pose.

The main contributions of this paper are as listed below.

- We propose a method of strong feature point screening based on the survival time. The method takes the number of frames that a feature point can be continuously tracked in a tracking thread as a metric, which we call life value, and filters out strong feature points as stable feature points based on the life value of each feature point. Such feature points have reliable values and descriptor invariants, and tracking them reduces cumulative errors; meanwhile, we propose to use the number of strong feature points as an indicator of frame quality. Unlike traditional algorithms that analyze image quality with pixel brightness and blurring, our method selects keyframes that contain more information about the previous frames and motion states.
- We propose a method to quantify the motion state based on the poses. The method can characterize the degree of motion intensity of the UAV, which is quantified in this paper by the compound rotation amount (this is calculated as the sum of the norms of the change in the bit pose between two adjacent keyframes). We synthetically consider the degree of change of the motion state where rotation and translation exist simultaneously, and then use the Lie algebraic form of the bit pose to perform the operation. In this paper, we experimentally verify the feasibility and reliability of this quantization method.
- We propose a keyframe strategy for UAV visual localization. Unlike the ORB-SLAM2 algorithm, which generates keyframes based on temporal and spatial distances, our keyframe strategy takes into account the image quality and motion state, and adds constraints such as the degree of overlap. Our strategy is based on strong feature points as the main strategy and compound rotation amount as the secondary strategy, by experimenting on a public dataset, we verified that our algorithm is more advantageous than the current mainstream algorithms under this keyframe strategy.

II. RELATED WORKS

In recent years, there have been several studies on SLAM-based UAV localization algorithms. For example, PTAM [10], a milestone SLAM system, innovatively proposed a framework of front-end tracking threads and back-end map building threads, and became the reference standard for many subsequent algorithms. Blösch *et al.* [11]–[13] improved PTAM so that it could be applied to UAVs, but since PTAM is a monocular algorithm designed for small indoor scenes in AR applications, it could not operate consistently and effectively in large scenes [11]. In 2014, Forster *et al.* [14] proposed the SVO algorithm for UAVs, which is based on the sparse direct method of visual odometry. The author also proposed a novel depth filter concept for position estimation of key points to better compute the positions of feature points; however, in order to improve the operating speed and make the system more lightweight, the system does not have a back-end optimization and loop closure detection link to loop closure to correct the drift, and thus the error is large. In 2017, Mur-Artal and Tardós [15] proposed ORB-SLAM2 with back end and loop closing; like the PTAM framework, it is also based on the feature point method for front-end tracking and uses bundle adjustment (BA) [16] and graph optimization in the back end. Unlike SVO, it adds complete loop closing and relocation [17], and utilizes a method based on the binary bag-of-words [18] for offline training. Thus, they successfully improved the real-time performance and robustness of the algorithm in large scenarios, and the algorithm achieved good results on the UAV dataset EuRoc [19]. Algorithms with the inclusion of back end and loop closing continuously generate redundant information under large scenes and long runs. To improve system efficiency and reliability, the selection of keyframes is crucial. Experiments by Leutenegger *et al.* [20]–[23] all demonstrated the improvement in system accuracy brought by keyframe selection. Meanwhile, Nerurkar *et al.* [24] proposed C-KLAM, which is a maximum a posteriori (MAP) estimator-based keyframe approach for SLAM. C-KLAM projects both proprioceptive and exteroceptive information from the non-keyframes to the keyframes using marginalization while maintaining the sparse structure of the associated information matrix. Thus, C-KLAM enhances the information content and quality of keyframes. Chen *et al.* [25] proposed a fast keyframe selection and switching algorithm to replace unsuitable keyframes with qualified backup frames, their method was based on ICP iterative implementation. Alonso *et al.* [26] proposed a novel keyframe selection strategy based on image quality and semantic information; this strategy was executed using a MiniNet network with a CNN architecture, and while it is capable of semantic filtering of images, it imposes requirements on hardware.

Inspired by the above work, the present paper considers the robustness of UAVs in motion scenarios with more rotation, and proposes a system for deciding keyframes based on motion states.

III. MOTION STATE-BASED KEYFRAME DECISION METHOD

A. SCREENING OF STRONG FEATURE POINTS IN GENTLE MOTION

The selection of feature points is a key aspect of the front end; Nister *et al.* [27] mentions that in feature point selection one should choose those points that have existed for a longer time. Meanwhile, in a paper on the SOFT algorithm [28], Cvišić *et al.* argues that feature stability leads to improved accuracy. Thus, we should select stable and strong feature points in the feature point selection session, and track them.

In this paper, we consider the characteristics of strong feature points as 1) feature points of map points that appear in consecutive frame images; 2) feature points that can be observed by cameras with multiple viewpoints. Such feature points have a longer survival time; and have some property invariants in the tracking of consecutive frames, the matching process can be simplified and the descriptors of the first observed feature points can be used for substitution when stable feature points are subsequently observed, thereby directly reducing drift errors.

In this paper, the survival time is added to the feature point attributes and described by the *life* value. The rules for defining the *life* value are as follows:

(1) The *life* value of the feature point that appears for the first time is 0.

(2) When the feature point is tracked to the next frame, the *life* value of the feature point is increased by 1; otherwise, it is set to 0.

Let the camera internal reference matrix be C, the current frame be z_{cur} and z_n be the previous n frames. Two adjacent frames have N feature points and correspond one to another. For example, the current frame z_{cur} and the previous frame z_1 are expressed as

$$z_{cur} = \left\{ z_{cur}^1, z_{cur}^2, \ldots, z_{cur}^N \right\}, \quad z_1 = \left\{ z_1^1, z_1^2, \ldots, z_1^N \right\}. \tag{1}$$

Then the strong feature point z_{str}^* will be

$$z_{str}^* = z_{cur}^k, \quad k \in [1, N], \quad \text{if } life\left(z_{cur}^k\right) \geq l. \tag{2}$$

where $life\left(z_{cur}^i\right)$ denotes the *life* value of feature point z_{cur}^i, and l denotes the *life* attribute, i.e., the feature points can be tracked in $z_{cur}, z_1 \ldots z_l$ frames. We specify the number of strong feature points in a frame as the sum of all feature points satisfying a given *life* value:

$$M = \sum_{i \geq l} \sum_{j \in (0,N]} f\left(z_i^j, l\right), \tag{3}$$

where

$$f\left(z_i^j, l\right) = \begin{cases} 1, & life\left(z_i^j\right) \geq l \\ 0, & life\left(z_i^j\right) < l \end{cases}. \tag{4}$$

i denotes the number of frames that can be tracked, M denotes the total number of feature points that exist to satisfy the *life* condition, and $f\left(z_i^j, l\right)$ denotes whether the feature point z_i^j with $life\left(z_i^j\right) \geq l$ exists, i.e., whether the $j-$th feature point can be tracked continuously for l frames. The number of strong feature points present inside each frame can be calculated by this formula.

In general, the number of such feature points is smaller than the conventionally extracted ones, and the 3D point cloud generated using the filtered strong feature points is more reliable. Meanwhile, we exclude the point cloud generated by the unstable feature points detected at the occlusion. The number of strong feature points contained in each frame also represents the quality of that frame, distinguishing our work from that of I. Alonso [26], who considered the quality of the frame based on a combined score of image blur and pixel brightness. We believe that in the feature point method, the image containing more strong feature points has better quality, contains more information about the previous frames, and is more representative of the camera motion during that time. Thus, the number of strong feature points is the basis of our dominant strategy for keyframe generation. When the UAV is in a state of gentle motion, we use the number of strong feature points as the basis for inserting keyframes.

B. CALCULATION OF COMPOUND ROTATIONAL AMOUNT IN ROTATIONAL MOTION

In rotated scenes, which result in a sharp decrease in feature points, there are fewer strong feature points to filter, and the dominant strategy fails. We need an auxiliary keyframe strategy for these scenes in order to insert keyframes quickly. To quantitatively analyze the motion state of the camera, we propose a compound rotation amount. We believe that the greater challenge to the algorithm is the rotation scene, and therefore we focus on using this amount to reflect the rotation. We specify the compound rotation amount for any two frames as

$$L_{ij} = \sum_{k=i}^{j-1} \|\xi_{k+1} - \xi_k\|, (i < j), \quad \text{where } \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6. \tag{5}$$

ξ is the Lie algebraic form of the pose of two frames, which is a 6×1 matrix. The first three dimensions ρ are related to translation, and the last three dimensions ϕ are related to rotation.

$$\rho = \frac{t}{J}, \tag{6}$$

where

$$J = \frac{\sin\theta}{\theta} I + \left(1 - \frac{\sin\theta}{\theta}\right) ii^T + \frac{1 - \cos\theta}{\theta} i^{\wedge}, \tag{7}$$

$$\phi = \theta i = \ln R^{\vee}. \tag{8}$$

R is the rotation matrix, and t is the translation vector, and I is the identity matrix.

The compound rotation amount uses the pose ξ as the operator, which reflects the motion state through equation (5).

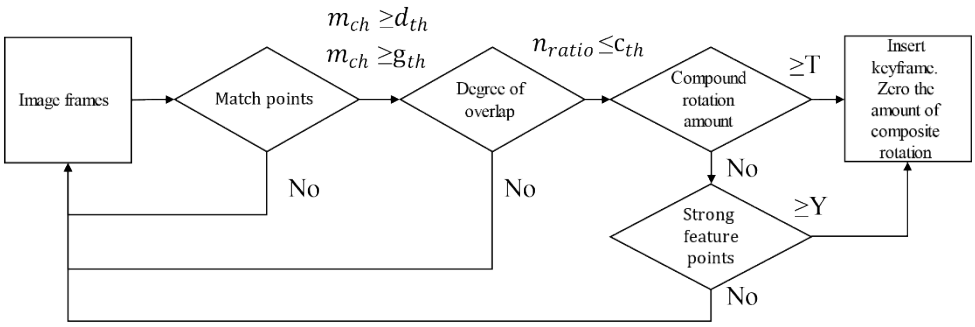


FIGURE 2. Keyframe strategy flow chart.

We want to obtain a numerical reflection when the camera encounters difficult scenes such as rotation, while inserting keyframes more intensively in such scenes. The result obtained from the calculation is a scalar quantity, and in this paper we ignore the directionality of the rotation, i.e., whenever rotation occurs, it positively increments the compound rotation amount L.

In this paper, the keyframe strategy based on the compound rotation amount is used as a secondary strategy that runs only when the camera is rotating. Thus, we set two measures to avoid inserting too many redundant keyframes.

(1) First, the degree of overlap of frames is added. Since gentle motion generates many redundant keyframes (e.g. the translation movement of long straight lines and rotational motion that is not violent), in order to avoid inserting too many redundant keyframes due to gentle motion, the judgment of the degree of overlap between the feature points and map points in any two frames is carried out. This judgment is described in section C.

(2) Second, we set the current compound rotation amount to 0 after each keyframe insertion. In other words, after each keyframe insertion effect, a certain amount of compound rotation must be reached again before inserting a frame. This enables us to reduce the amount of system operations. Due to the existence of this rule, in the actual algorithm, the compound rotation amount is the sum of the compound rotation amounts of all frames between two adjacent keyframes, that is, the formula reflects whether the camera pose changes drastically in two adjacent keyframes.

C. KEYFRAME STRATEGY

Zhang *et al.* [29] considered that the keyframe detection criterion is the selected keyframe, which can reasonably contain the pose in the map and cover the environment. Alonso *et al.* [26] set the indices for evaluating image quality (such as brightness and blur) in the keyframe strategy. Inspired by these works, our algorithm is based on two strategies: one of image quality, and one of motion state (fig. 3). That is, the keyframes extracted by our algorithm should contain a higher number of strong feature points, and more keyframes should be added in the case of severe rotation. The keyframe strategy is thus developed as shown in Fig. 2.

The keyframe strategy first determines the current number m_{ch} of feature point matches. If it is lower than the threshold d_{th}, the current tracking is considered to have failed and the process needs to be terminated for repositioning. Solving the motion for the camera can be considered as a Perspective-n-Point (PnP) problem, we use the EPnP (Efficient PnP) [30] method to solve the motion; this requires at least four point pairs to perform the operation. We refer to the pair-pole geometry method to ensure the adequacy and accuracy of matching points, and d_{th} selects eight point pairs as the threshold. At the same time, we judge whether the number m_{ch} of matching map points between the current frame and the reference frame exceeds g_{th}. If it does not, the current tracking effect and matching map points are decreasing sharply, and the strategy needs to insert keyframes and insert the current frame into the keyframe sequence to retain it.

We then proceed to judge the degree of image overlap. As mentioned in section B, in order to avoid gentle motion (e.g. the translation movement of long straight lines and rotational motion that is not violent) from mistakenly triggering the frame insertion mode, the current frame is compared with the reference frame to calculate the proportion of feature points that overlap with the reference frame. The calculation is as follows:

$$n_{ratio} = \frac{n_{cur}}{n_{total}}. \tag{9}$$

where n_{cur} is the number of map points observed in the current frame, and n_{total} is the total number of map points in the current frame. The closer this ratio is to 1, the more map points on the trace; conversely, the closer it is to 0, the less map points on the trace.

We set the compound rotation amount threshold T (refer to section V and fig. 5) as the entry basis for judging the rotation frame insertion strategy. If the compound rotation amount threshold is satisfied, the strategy enters the frame insertion mode based on the compound rotation amount. If the compound rotation amount threshold is not satisfied, the strategy enters the frame insertion mode based on the number of strong feature points. The number of strong feature points

counted in the current frame reaches the threshold Y, and then the keyframe is inserted. We set B as the Boolean of the compound rotation amount, and D as the Boolean of strong feature points; we then judge whether these two amounts reach the threshold value. Meanwhile, V is the Boolean for generating keyframes. The keyframe strategy to judge whether to insert frames is as follows:

$$V = \begin{cases} 1, & \text{if } m_{ch} \geq d_{th} \text{ and } m_{ch} \geq g_{th} \\ & \text{and } n_{ratio} \leq c_{th} \\ & \text{and } (B = 1 \text{or} D = 1) \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

where

$$B = \begin{cases} 1, & \text{if } L \geq \mathrm{T} \\ 0, & \text{otherwise} \end{cases}, D = \begin{cases} 1, & \text{if } M \geq \mathrm{Y} \\ 0, & \text{otherwise} \end{cases}. \tag{11}$$

After each execution of the keyframe insertion strategy, the compound rotation amount accumulated between the two keyframes is reset, thus starting a new round of compound rotation amount calculation and keyframe strategy.

IV. POSITION OPTIMIZATION

To improve accuracy, we use bundle adjustment (BA) and loop closing, which optimize the keyframe bit pose. The back end of the algorithm in this paper uses g2o [31] as a method for graph optimization.

The real coordinates of any feature point existing in three-dimensional space are $X^j = (xyz)^T$, and the coordinate in the corresponding z_{cur} and z_1 coordinate system is z^j_{cur}, z^j_1. Thus, by the projection relation, we obtain

$$\lambda_1 \begin{bmatrix} z^j_{cur} \\ 1 \end{bmatrix} = CX^j, \quad \lambda_2 \begin{bmatrix} z^j_1 \\ 1 \end{bmatrix} = C\left(RX^j + t\right). \tag{12}$$

where λ_1 and λ_2 are the depth values of the two pixels, R is the rotation matrix, and t is the translation vector. Then the camera pose under this condition is transformed into the optimal solution of the following formula:

$$\min_{X,R,t} \sum_{j=1}^{N} w_j \left(\left\| \frac{1}{\lambda_1} CX^j - \left[z^j_{cur}, 1 \right]^T \right\|^2 + \times \left\| \frac{1}{\lambda_2} C\left(RX^j + t\right) - \left[z^j_1, 1 \right]^T \right\|^2 \right). \tag{13}$$

This is our cost function. We refer to the work of Geiger *et al.* [32], but add the weight w_j of the j-th point, which is calculated based on the strong feature points and the amount of compound rotation. The weighting function is

$$w_j = w_{lens} w_{sa}, \tag{14}$$

where

$$w_{lens} = \left(\frac{u_j - c_u}{c_u} + p_{lens} \right)^{-1}, \tag{15}$$

$$w_{sa} = 1 + \log\left(life\left(z^j_{cur} \right) + L + 1 \right). \tag{16}$$

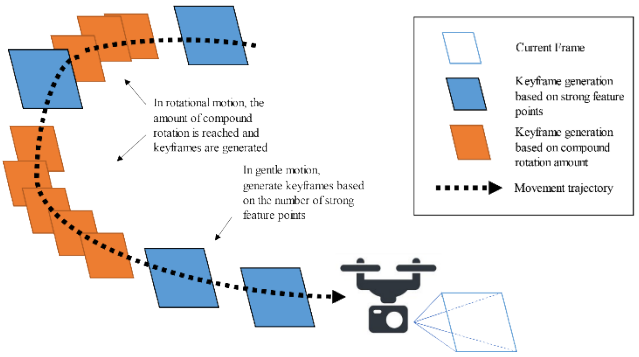


FIGURE 3. Schematic diagram of keyframe strategy. The blue frames are generated by the dominant keyframe strategy, which decides whether to generate keyframes based on image quality (the number of strong feature points reflects the quality of the current frame image), this strategy is used to gentle motion; when the camera starts to rotate or the motion is intense, the secondary strategy based on the compound rotation amount comes into effect (orange frames), which generates the current frame as a keyframe.

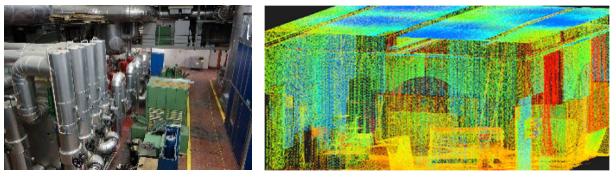


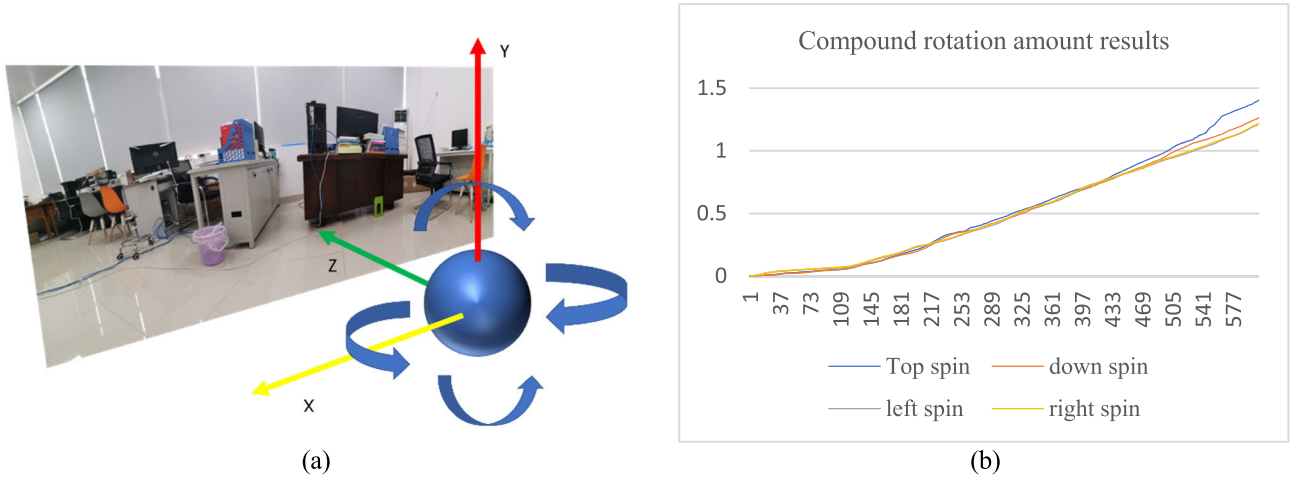
FIGURE 4. ETH Machine hall (left) ground-truth 3D scan of the Vicon environment (right).

u_j belongs to $[uv1]^T$, which is the homogeneous image coordinates; c_u belongs to $[c_u c_v]^T$, which is the image principal point; and parameter p_{lens} depends on the stereo camera and lens setup. The weighting function integrates the life value and compound rotation of the current point, which makes feature points closer to the image center in the horizontal direction more meaningful. This function enables the algorithm to focus more on feature points close to the image center during bitwise pose optimization, to increase feature point weights appropriately during rotation and in the presence of strong feature points, and to reduce the influence of distant and edge feature points.

In order to obtain better accuracy, this paper uses feature point-based loop closure detection by means of image-to-image matching. Since the bag-of-words library [18], [33] detection method used in ORB-SLAM2 is extremely efficient, scalable, easy to use, and can be trained offline, we use the loop closure detection scheme of ORB-SLAM2 to perform loop closure detection. Loop closing reduces drift by performing positional correction.

V. EXPERIMENT AND ANALYSIS

To verify the effectiveness of the proposed algorithm, we tested it on the public dataset EuRoc [19]. The EuRoC dataset contains 11 sequences recorded with the Asctec Firefly hex-rotor helicopter. Two batches of datasets are available. The first batch was recorded in the ETH machine hall (see Fig. 4) and contains millimeter-accurate position ground-truth from a Leica MS50 laser tracker. The second



(a) (b)

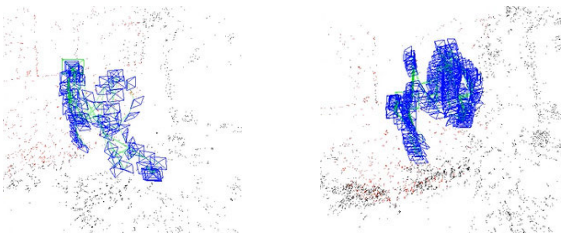
FIGURE 5. Experimental results of the compound rotation amount in a real environment. (a) the experimental environment. (b) the compound rotation amount in four directions.

batch contains Vicon 6D pose ground truth and a precise, registered 3D scan of the environment (see Fig. 4). Depending on the texture, brightness, and UAV dynamics, the sequences are classified as easy, medium, and difficult. The computer employed was a laptop with Intel i5-4210M CPU, 8G RAM, 2.60GHz main frequency, and Ubuntu 16.04 operating system.

A. COMPOUND ROTATION AMOUNT VERIFICATION EXPERIMENT

In order to verify that the formula in section III reflects the degree of rotation, we analyze the change in compound rotation amount during the rotation motion by recording video of the corresponding rotation motion. We used the handheld gimbal device, placed the binocular camera ZEDmini in the center of the screen, and rotated it at a uniform speed to 50 degrees up and down, and left and right. We then used the proposed algorithm to calculate the compound rotation amount of the captured video. In Fig. 5, we plot the compound rotation amount in four directions.

It can be seen that since the directionality of rotation is not set, the compound rotation amount in each direction changes in a consistent pattern. During the left and right rotation, the curves overlap to a high degree; during the up and down rotation, differences in values appear in the later angles (however, the errors are within a very small range). In addition to the inconsistency caused by the speed of movement, the monotony of the scene texture and the difficulty of feature extraction during the up and down rotation also have a certain relationship. Taking 50k as an example, we can observe from the curve that the corresponding compound rotation amount is about 1.2, and thus the threshold value T of compound rotation is set with this value (1.2 Corresponding to 50°) as a reference. Note that a smaller value of T is set if the scene to be dealt with is more complicated. In this paper, the reference range of T is 0.4–1.4.



(a) (b)

FIGURE 6. Screenshot of the keyframes of Ours(-) and Ours; (a) Ours(-):the algorithm with the thread of the keyframe strategy based on strong feature points only; (b) Ours: the algorithm with complete keyframe strategy. The blue rectangle in the picture is the keyframe, and the black points are map points. It can be seen that after adding the keyframe strategy based on the amount of compound rotation, the keyframes are inserted intensively during the rotation movement.

B. KEYFRAME STRATEGY EXPERIMENTS

To demonstrate the effect of the keyframe strategy, we extracted keyframe screenshots of the algorithm while the dataset was running. Fig. 6 shows the keyframe graph of the thread based on strong feature points only (marked as Ours(-)), along with the complete system (marked as Ours). To demonstrate an obvious effect, T is 0.24. We cannot implement a keyframe strategy that only has a thread based on a compound rotation amount because the thread only takes effect under certain rotation conditions, and it is difficult to track camera movement independently.

The Fig. 7 is a visualization of the keyframe insertion process on the sequence MH_03_medium. Frames 1000–1200 (when the UAV was moving in a lateral rotation) and 1400–1500 (when the UAV was moving in a long straight line back and forth) were selected to demonstrate the keyframe insertion.

By comparing the left and right graphs, we can observe three main findings. 1) Since the keyframe strategy in this paper is based on the judgment of compound rotation amount, strong feature points and degree of overlap,

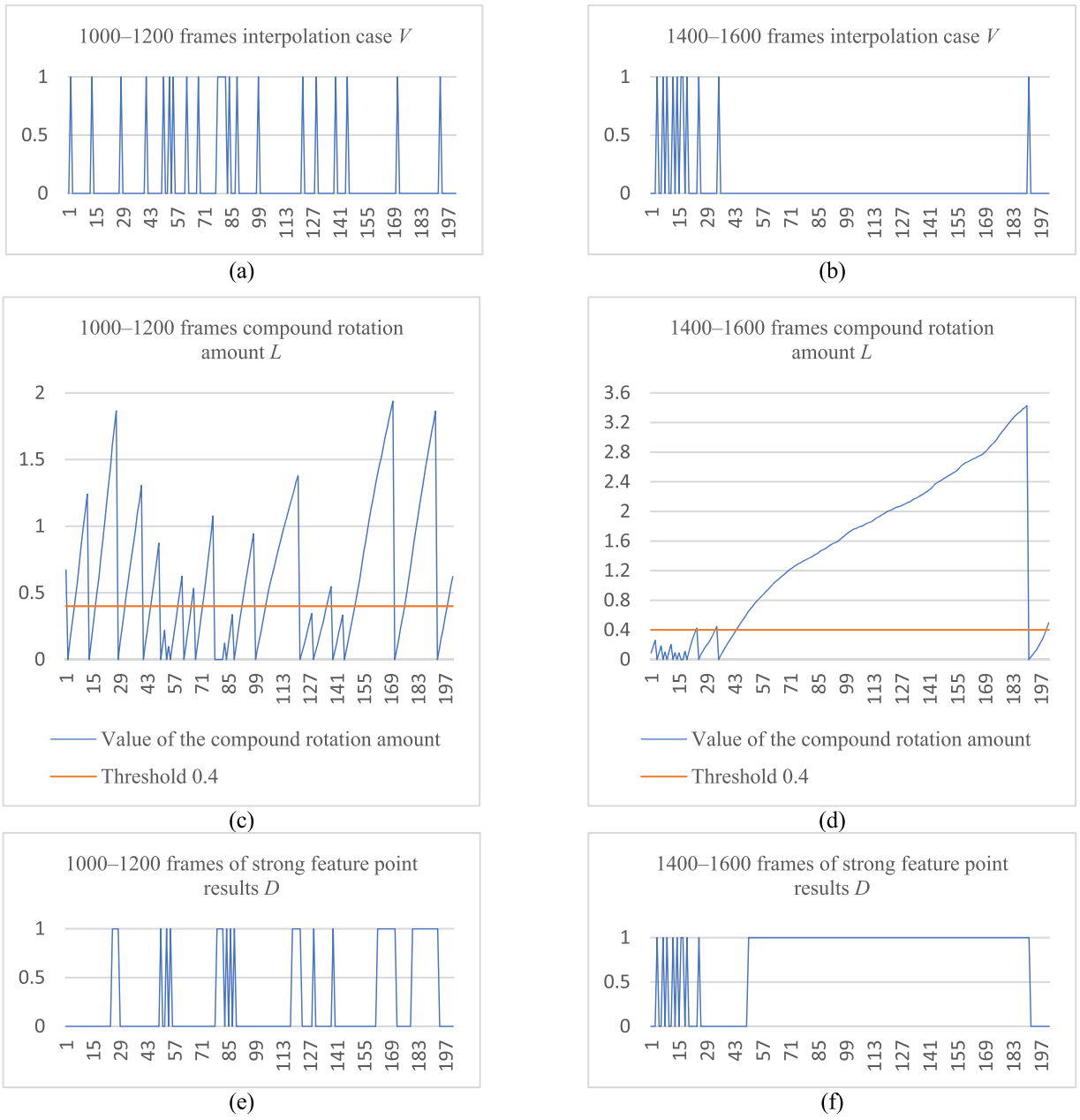


FIGURE 7. Line diagrams of each process of keyframe generation. (a) and (b) represent the frame ID and Boolean value *V* of the inserted keyframes; (c) and (d) represent the *L* value of the compound rotation amount; (e) and (f) represent the Boolean value *D* of the strong feature points; (g) and (h) represent the n_{ratio} value of the degree of overlap. (a), (c), (e), and (g) represent frames 1000–1200 of the sequence MH_03_medium. (b), (d), (f), and (h) represent frames 1400–1600 of the sequence MH_03_medium.

and only when the threshold of the three is met can it be included as a keyframe to participate in subsequent calculation and optimization. 2) Due to the setting of the degree of overlap, the false frame insertion caused by long linear motion is effectively suppressed, as seen in the case of frame insertion in the middle of 1400–1600 frames. 3) The compound rotation amount is set to 0 after the keyframe insertion, and thus the compound rotation amount curve starts and ends at the keyframe insertion each time, which reflects the rotation intensity between the two keyframes.

C. COMPARISON WITH OTHER ALGORITHMS

In order to demonstrate the overall performance of the proposed algorithm, we compared it with other algorithms on the EuRoc dataset for UAV scenes. To ensure fairness, we did not adjust the parameters of each algorithm. The evaluation criteria are listed in Table 1, and each algorithm was run five times. We used the absolute translation root mean square error (RMSE) [34] for the calculation. ORB-SLAM2 Stereo data is obtained by running the algorithm on our equipment, while data for SVO+gtsam and VINS-Mono are from [35]. We highlight the sequences from the difficult category

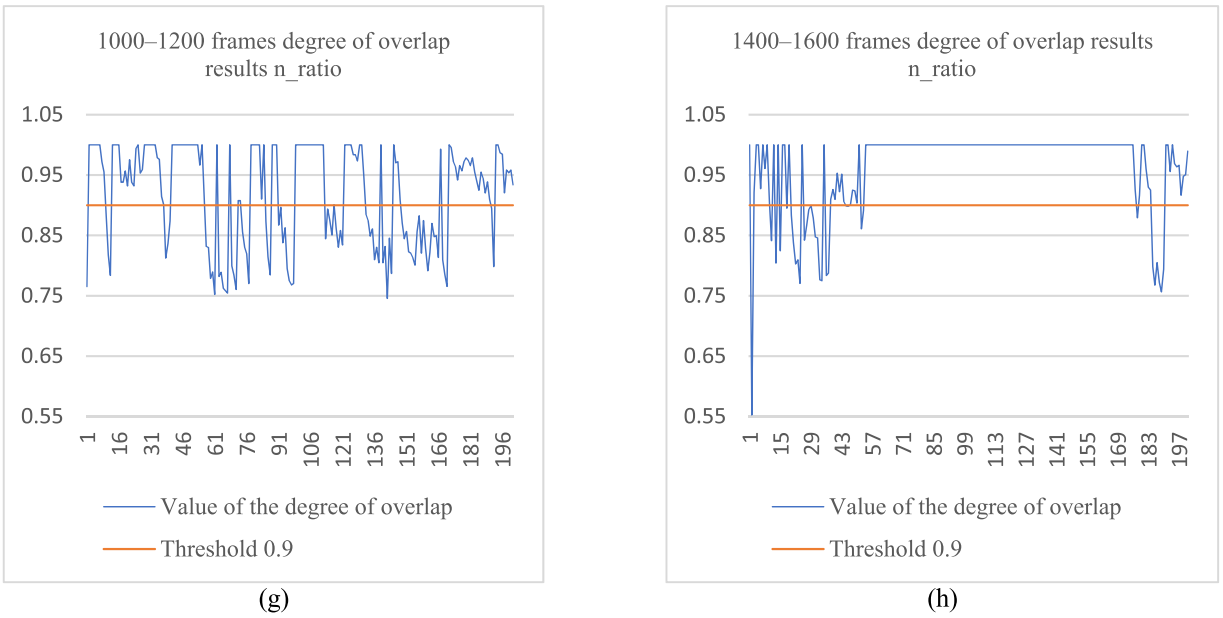


FIGURE 7. *(Continued.)* **Line diagrams of each process of keyframe generation. (a) and (b) represent the frame ID and Boolean value *V* of the inserted keyframes; (c) and (d) represent the *L* value of the compound rotation amount; (e) and (f) represent the Boolean value *D* of the strong feature points; (g) and (h) represent the n_{ratio} value of the degree of overlap. (a), (c), (e), and (g) represent frames 1000–1200 of the sequence MH_03_medium. (b), (d), (f), and (h) represent frames 1400–1600 of the sequence MH_03_medium.**

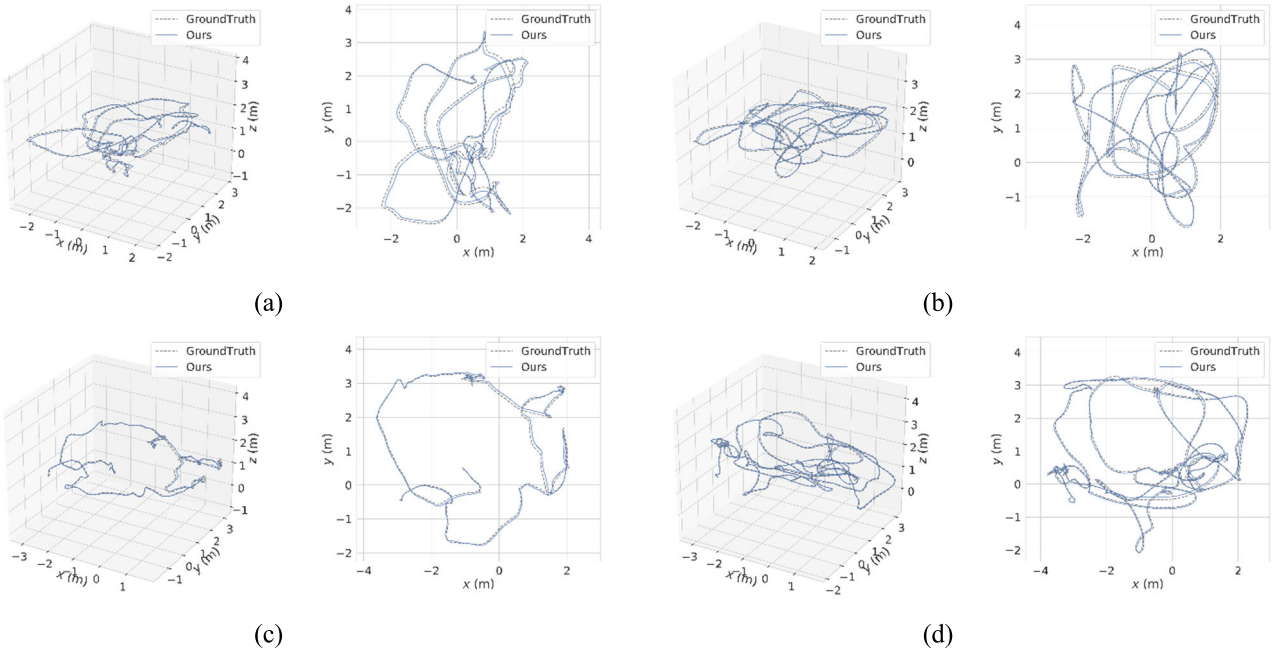


FIGURE 8. **A plot of the estimated trajectory of our algorithm on easy and medium sequences, where the blue line represents the estimated trajectory of our algorithm and the gray dotted line represents the ground truth trajectories. (a) V1_01_easy; (b) V1_02_medium; (c) V2_01_easy; (d) V2_02_medium.**

because their samples have more rotations and rapidly changing scenes.

(1) Compared with ORB-SLAM2 Stereo, SVO+gtsam, and VINS-Mono, our method has good performance in 11 sequences (see bolded number). Our method achieves significant results in the sequences of V1_03_difficult, V2_03_difficult, MH_04_difficult, and MH_05_difficult (see boxed numbers). This is consistent with our intention to improve the robustness of the algorithm in difficult scenarios such as drone rotation. In MH_04_difficult, the error is reduced by 71.5%; in MH_05_difficult, the error is reduced by 87%; in V1_03_difficult, the error is reduced by 50.4%; and in V2_03_difficult, the error is reduced by 33.6%.

TABLE 1. RMSE error (m) for each algorithm on the EuRoc dataset, where $\bar{E}$ is the mean, and SR_E is the standard deviation. The bolded numbers represents the algorithm that achieves the smallest result in the sequence. The boxed numbers represent the algorithm that achieves the smallest result in the difficult sequence.

Method	$\bar{E} \pm SR_E$				
	ORB-SLAM2 Stereo	Ours(-)	Ours	SVO+gtsam*	VINS-Mono*
V1_01_easy	0.0863±0.0002	0.0867±0.0003	0.0862±0.0004	**0.07**	**0.07**
V1_02_medium	**0.0640±0.0004**	0.0661±0.0003	0.0642±0.0004	0.11	0.10
V1_03_difficult	0.0746±0.0074	0.0911±0.0040	**0.0645±0.0031**	x	0.13
V2_01_easy	0.0596±0.0027	0.0592±0.0008	**0.0570±0.0024**	0.07	0.08
V2_02_medium	0.0593±0.0050	0.0550±0.0008	**0.0532±0.0021**	x	0.08
V2_03_difficult	0.2207±0.0392	0.1907±0.0165	**0.1466±0.0166**	x	0.21
MH_01_easy	0.0370±0.0008	0.0346±0.0005	**0.0341±0.0007**	0.05	0.27
MH_02_easy	0.0432±0.0016	0.0445±0.0006	0.0425±0.0018	**0.03**	0.12
MH_03_medium	0.0441±0.0051	0.0388±0.0004	**0.0363±0.0012**	0.12	0.13
MH_04_difficult	0.0940±0.0194	0.0770±0.0091	**0.0656±0.0174**	0.13	0.23
MH_05_difficult	0.0670±0.0261	0.0754±0.0060	**0.0455±0.0040**	0.16	0.35

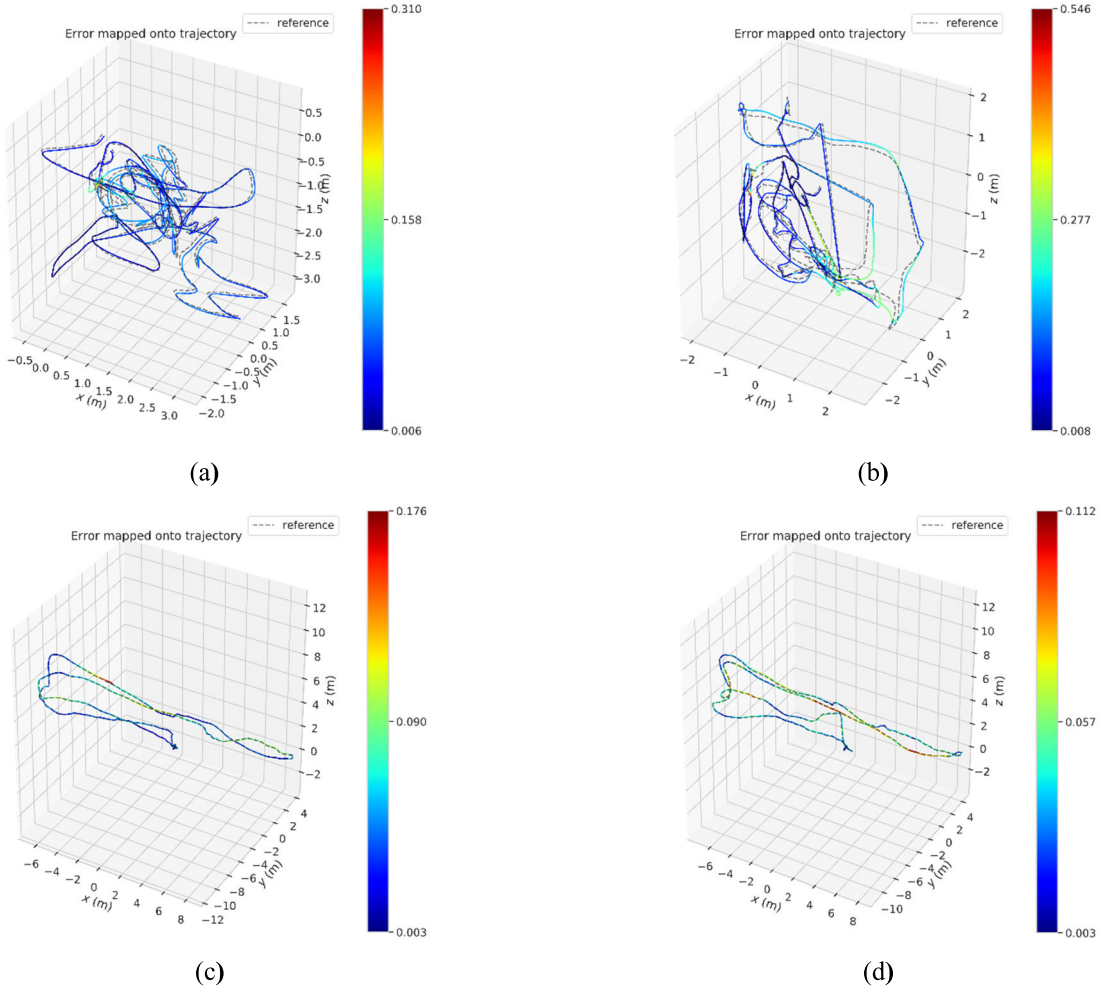


FIGURE 9. A plot of the estimated trajectory of our algorithm on difficult sequences. These are four heat maps color-coded, red corresponds to higher error levels, and blue to lower ones. The gray dotted line is the reference (ground truth trajectories). (a) V1_03_difficult; (b) V2_03_difficult; (c) MH_04_ difficult; (d) MH_05_ difficult.

(2) Comparing the results obtained in the easy and medium sequences, we find that the proposed algorithm and ORB-SLAM2 achieved similar error results. This is because these two types of sequences do not have rotational motions that have a large impact on the system. Thus, in both types of sequences where the rotational motion is not drastic, Ours(−) can achieve good results, the results for Ours and Ours(−) are nearly identical. But in the difficult sequences, Ours achieves significant results; this is related to the setting of our dominant strategy (strong feature point-based keyframe strategy).

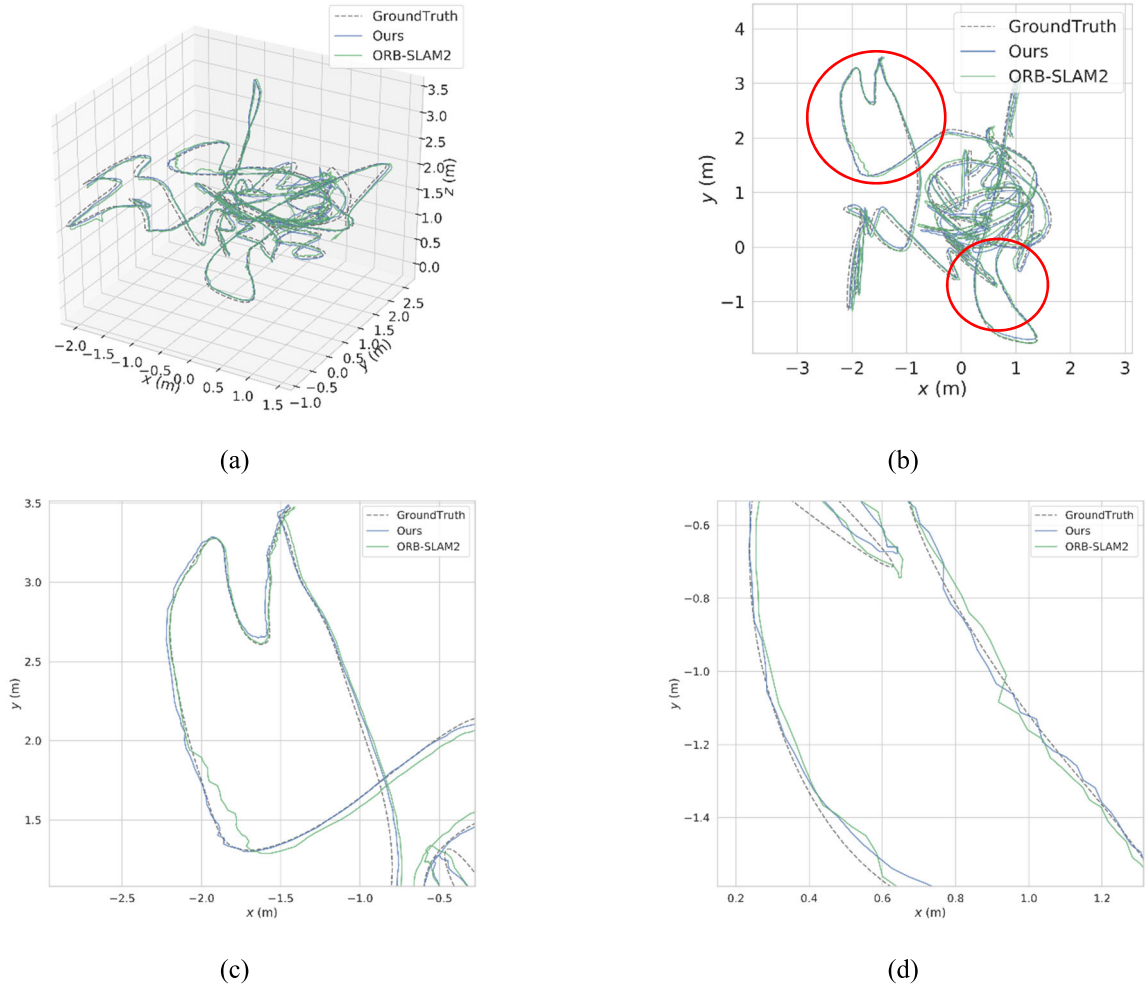


FIGURE 10. In the V1_03_difficult sequence, we compare the estimated trajectories of our own algorithm with those of ORB-SLAM2. The blue line represents the estimated trajectory of our algorithm, the green line represents the estimated trajectory of ORB-SLAM2, and the black dashed line represents the ground truth. (a) Estimated trajectory in the xyz coordinate axis;(b) Estimated trajectory in the xy coordinate axis; (c) Enlarge the upper left corner of the (b); (d) Enlarge the lower right corner of the (b).

Since the quality of feature point extraction and quantity depend on the scene, unlike the variation of the bit pose (which is less dependent on the scene), the strong feature point-based keyframe strategy is less capable than the strategy based on the compound rotation amount; also, in all 11 sequences with different scenes, the algorithm(Ours) with the addition of the compound rotation amount outperforms the algorithm(Ours(−)) with the thread of the keyframe strategy based on strong feature points only. This also proves our point that the dependence of the strong feature point-based strategy is stronger on the scenes and performs more inconsistently for different scenes.

(3) Our algorithm also achieved some unexpected results: it performed well in MH_03_ medium. Besides the fact that the sequences scenario did not interfere much with our dominant strategy, this may be due to the fact that the parameter settings of this experiment happened to match the sequence.

We recorded the processing time of the proposed algorithm for each sequence of the dataset. From Table 2, we can see that the processing time of our algorithm is less than 50 ms, that is, it can run above 20 Hz and thus can meet the real-time requirements of UAVs.

Fig. 8 shows the estimation trajectory of our algorithm under partial easy and medium sequences. We focus on the estimated trajectory of the difficult sequence, and we will show the effect of our algorithm in two aspects: the global estimated trajectory error (fig. 9) and the local trajectory effect (fig. 10 and fig. 11).

We compare the trajectory of our algorithm and ORB-SLAM2 in V1_03_difficult in Fig. 10. It can be seen that, at the rotation in the trajectory, compared with the ground truth, our algorithm obtains better results and our trajectory is closer to the ground truth. The error variation in the xyz axis direction is presented in Fig. 11.

By comparing our algorithm with ORB-SLAM2 on the xyz axis, we can find that there is not much difference between the proposed algorithm and ORB-SLAM2 in the x and y axes. Moreover, in the z axis, our algorithm is more robust and

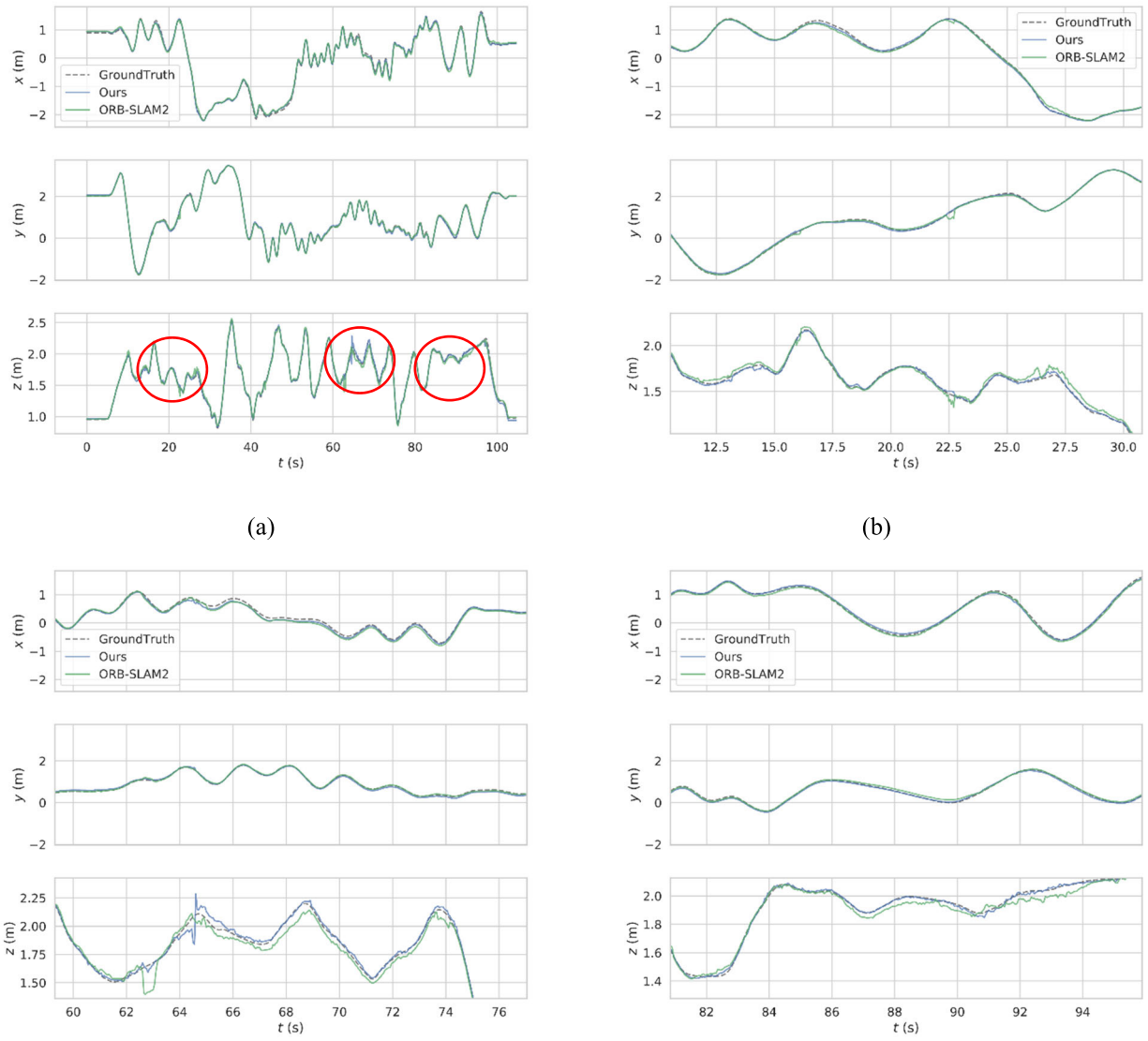


FIGURE 11. Error variation graph and enlarged graph in each axis of xyz. (a) Variation of the estimated trajectories of our algorithm and ORB-SLAM2 under the three directions of the xyz axis. We have used red circles and circled three places from left to right for the analysis. (b) Enlarged view of the first red circle. (c) Enlarged view of the second red circle. (d) Enlarged view of the third red circle.

TABLE 2. Number of keyframes and tracking time of our algorithm, under all sequences of the EuRoC dataset.

Seq.	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03	MH_01	MH_02	MH_03	MH_04	MH_05
keyframes	110	154	269	106	284	307	490	425	464	323	387
T_{media} (ms)	39.2	39.7	38.7	37.7	42.2	35.9	44.3	42.9	45.9	41.4	41.6
T_{mean} (ms)	39.3	39.5	38.2	37.6	43.6	35.8	47.3	45.5	46.3	41.8	42.8

accurate than ORB-SLAM2. Since we introduce a compound rotation amount reflecting the motion state to cope with the rotation situation, our algorithm can insert keyframes containing strong feature points when the UAV performs up-and-down spatial sway and rotation in the z axis; this effectively improves the rotation resistance and positioning accuracy of our algorithm.

VI. CONCLUSION

Moving and rotating scenes are challenges for UAV vision localization algorithms. In order to improve the accuracy and robustness of UAVs in this type of scene, we propose a localization algorithm based on the keyframe of the motion state decision. The algorithm is developed based on a complete SLAM framework. When selecting features, we select

strong feature points that survive longer, and use these strong feature points to construct stable and reliable point clouds. We believe that the criterion for evaluating the image quality is the number of strong feature points available, and thus we construct the keyframe strategy as the dominant strategy to improve the accuracy of our algorithm. In terms of robustness, we propose a compound rotation amount to cope with the rotating scene to characterize the current state of motion, and achieve multiple keyframe insertion in the rotating scene by setting a loose threshold. To avoid redundant insertion in the non-rotating case, we add constraints such as overlap determination, so that the final keyframe strategy forms a primary and a secondary pattern; this ensures accuracy and robustness. Then, in the back end, we used a weighted cost function based on life values and compound rotation amounts for selective pose optimization. Through experiments conducted on the EuRoc dataset, we visualize and analyze the running process and performance of our algorithm. We verify the effectiveness of our algorithm, and its ability to achieve strong results without tuning the parameters.

The current algorithm is based on point features, which are dependent on the scene environment. In the future, we will consider adding line features and plane features. The multi-feature method will be able to track and locate stably in more scenes.

REFERENCES

[1] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza, "Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 111–118, doi: 10.1109/ICRA.2015.7138988.

[2] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *J. Intell. Robotic Syst.*, vol. 87, no. 1, pp. 141–168, Jul. 2017, doi: 10.1007/s10846-017-0483-z.

[3] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part I: The first 30 years and fundamentals," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78–90, Dec. 2011, doi: 10.1109/MRA.2011.943233.

[4] Y. Xia, J. Li, L. Qi, and H. Fan, "Loop closure detection for visual SLAM using PCANet features," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 2274–2281, doi: 10.1109/IJCNN.2016.7727481.

[5] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM–learning a compact, optimisable representation for dense visual SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2560–2568, doi: 10.1109/CVPR.2018.00271.

[6] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.

[7] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 835–852, doi: 10.1007/978-3-030-01237-3_50.

[8] M. Poggi, D. Pallotti, F. Tosi, and S. Mattoccia, "Guided stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 979–988, doi: 10.1109/CVPR.2019.00107.

[9] X. Song, X. Zhao, H. Hu, and L. Fang, "Edgestereo: A context integrated residual pyramid network for stereo matching," in *Proc. 14th Asian Conf. Comput. Vis. (ACCV)*, Perth, QLD, Australia, 2018, pp. 20–35.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nara, Japan, Nov. 2007, pp. 225–234, doi: 10.1109/ISMAR.2007.4538852.

[11] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *J. Field Robot.*, vol. 30, no. 5, pp. 803–831, Sep. 2013, doi: 10.1002/rob.21466.

[12] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 21–28, doi: 10.1109/ROBOT.2010.5509920.

[13] J. Engel, J. Sturm, and D. Cremers, "Accurate figure flying with a quadrocopter using onboard visual and inertial sensing," *Proc. Work. Int. Conf. Intell. Robot Syst.*, 2012, vol. 320, no. 240, pp. 2815–2821, doi: 10.1007/s10846-013-9918-3.

[14] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May 2014, pp. 15–22, doi: 10.1109/ICRA.2014.6906584.

[15] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.

[16] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 2657–2664, doi: 10.1109/ROBOT.2010.5509636.

[17] R. Mur-Artal and J. D. Tardos, "Fast relocalisation and loop closing in keyframe-based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May 2014, pp. 846–853, doi: 10.1109/ICRA.2014.6906953.

[18] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012, doi: 10.1109/TRO.2012.2197158.

[19] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016, doi: 10.1177/0278364915620033.

[20] Z. Dong, G. Zhang, J. Jia, and H. Bao, "Keyframe-based real-time camera tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 1538–1545, doi: 10.1109/ICCV.2009.5459273.

[21] J. Stalbaum and J.-B. Song, "Keyframe and inlier selection for visual SLAM," in *Proc. 10th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Jeju, South Korea, Oct. 2013, pp. 391–396, doi: 10.1109/URAI.2013.6677295.

[22] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015, doi: 10.1177/0278364914554813.

[23] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 5110–5117, doi: 10.1109/ICRA.2017.7989597.

[24] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-KLAM: Constrained keyframe-based localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May 2014, pp. 3638–3643, doi: 10.1109/ICRA.2014.6907385.

[25] C.-W. Chen, W.-Y. Hsiao, T.-Y. Lin, J. Wang, and M.-D. Shieh, "Fast keyframe selection and switching for ICP-based camera pose estimation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, May 2018, pp. 1–4, doi: 10.1109/ISCAS.2018.8351436.

[26] I. Alonso, L. Riazuelo, and A. C. Murillo, "Enhancing V-SLAM keyframe selection with an efficient ConvNet for semantic analysis," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 4717–4723, doi: 10.1109/ICRA.2019.8793923.

[27] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Washington, DC, USA, Jun./Jul. 2004, pp. I–I, doi: 10.1109/CVPR.2004.1315094.

[28] I. Cvisic and I. Petrovic, "Stereo odometry based on careful feature selection and tracking," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Lincoln, U.K., Sep. 2015, pp. 1–6, doi: 10.1109/ECMR.2015.7324219.

[29] H. Zhang, B. Li, and D. Yang, "Keyframe detection for appearance-based visual SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, Oct. 2010, pp. 2071–2076, doi: 10.1109/IROS.2010.5650625.

[30] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009, doi: 10.1007/s11263-008-0152-6.

[31] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 3607–3613, doi: 10.1109/ICRA.2011.5979949.
[32] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 963–968, doi: 10.1109/IVS.2011.5940405.
[33] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. New York, NY, USA, Jun. 2006, pp. 2161–2168, doi: 10.1109/CVPR.2006.264.
[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura-Algarve, Portugal, Oct. 2012, pp. 573–580, doi: 10.1109/IROS.2012.6385773.
[35] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 2502–2509, doi: 10.1109/ICRA.2018.8460664.

YIMIN LUO received the B.S. degree in electrical engineering and automation from the Hefei University of Technology, Hefei, China, in 2017. He is currently pursuing the M.S. degree in control science and engineering with Guangxi University, Nanning, China.

YONG LI received the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, China, in 2020. He is currently an Assistant Professor with the School of Electrical Engineering, Guangxi University, Nanning, China. His research interests include intelligent robot, point cloud processing, computer vision, and pattern recognition.

ZHITENG LI received the B.S. degree in measurement and control technology and instrumentation program from the Harbin Institute of Technology, Harbin, China, in 2012. He is currently pursuing the M.S. degree in control science and engineering with Guangxi University, Nanning, China.

FENG SHUANG (Member, IEEE) received the bachelor's degree from the Special Class of Gifted Young, University of Science and Technology of China (USTC), in 1995, and the Ph.D. degree from the Department of Chemical Physics, USTC, in 2000. He worked as the Director of the Robot Sensor Laboratory, Institute of Intelligent Machines (IIM), USTC. From 2001 to 2003, he was a Research Associate with Princeton University. From 2004 to 2009, he was a Research Staff Member with Princeton University. He joined IIM, as a Full Professor, in 2009. He was selected as the member of One Hundred Talented People of Chinese Academy of Sciences. He transferred to Guangxi University, in 2018, as the Dean of the School of Electrical Engineering. He is currently the Dean and a Professor with the School of Electrical Engineering, Guangxi University. He has published more than 60 articles and applied more than ten national invention patents. His research interests include intelligent mobile robot, multi-dimensional force sensors, and quantum system control.