IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 21, 2019, accepted December 18, 2019, date of publication December 20, 2019,
date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2961266

A Stereo Visual-Inertial SLAM Approach for Indoor Mobile Robots in Unknown Environments Without Occlusions

CHANG CHEN, HUA ZHU, LEI WANG, AND YU LIU
School of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China
Jiangsu Collaborative Innovation Center of Intelligent Mining Equipment, China University of Mining and Technology, Xuzhou 221008, China

Corresponding author: Hua Zhu (zhuhua83591917@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0808000, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), China.

ABSTRACT When mobile robots are working in indoor unknown environments, the surrounding scenes are mainly low texture or repeating texture. This means that image features are easily lost when tracking the robots, and poses are difficult to estimate as the robot moves back and forth in a narrow area. In order to improve such tracking problems, we propose a one-circle feature-matching method, which refers to a sequence of the circle matching for the time after space (STCM), and an STCM-based visual-inertial simultaneous localization and mapping (STCM-SLAM) technique. This strategy tightly couples the stereo camera and the inertial measurement unit (IMU) in order to better estimate poses of the mobile robot when working indoors. Forward backward optical flow is used to track image features. The absolute accuracy and relative accuracy of STCM increase by 37.869% and 129.167%, respectively, when compared with correlation flow. In addition, we compare our proposed method with other state-of-the-art methods. In terms of relative pose error, the accuracy of STCM-SLAM is an order of magnitude greater than ORB-SLAM2, and two orders of magnitude greater than OKVIS. Our experiments show that STCM-SLAM has obvious advantages over the OKVIS method, specifically in terms of scale error, running frequency, and CPU load. STCM-SLAM also performs the best under real-time conditions. In the indoor experiments, STCM-SLAM is able to accurately estimate the trajectory of the mobile robot. Based on the root mean square error, mean error, and standard deviation, the accuracy of STCM-SLAM is ultimately superior to that of either ORB-SLAM2 or OKVIS.

INDEX TERMS Indoor mobile robots, multi-sensor fusion, nonlinear optimization, SLAM.

I. INTRODUCTION

The recent development of artificial intelligence and computer vision has led to unprecedented growth in the robotics industries. Among many kinds of robots that have been developed, the autonomous mobile robot has become increasingly popular. In order to achieve autonomous walking, however, one of the most important features to perfect is real-time localization with mapping. When a mobile robot is performing in an unknown environment, it needs to sense its poses in real time, make control decisions based on these poses in the global map, and walk autonomously in order to fulfill the tasks.

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

The aim of simultaneous localization and mapping (SLAM) technology is to allow a robot to sense its surrounding environmental information in real time in an unknown environment, based on data obtained by sensors such as cameras, lidars, and ultrasonic range finders. Using this information, the robot constructs a map, within which it can locate itself [1], [2]. Over a development period of 30 years, SLAM technology has achieved brilliant results and has been widely used in mobile robots [3], micro air vehicles (MAV) [4], unmanned aerial vehicles (UAV) [5], unmanned vehicles [6], virtual reality and augmented reality [7], and other areas. Applying SLAM technology to the mobile robot enables it to sense the surrounding environment, locate itself, and construct a 3D map, all of which greatly enhance its autonomy. However, in low-texture, and repetitive

texture indoor scenes, where the mobile robot is required to climb, accelerate, decelerate, and emergency stop, image features can be hard to track, and the scale estimation currently has errors. To solve this issue, robust real-time localization and mapping systems are essential. In addition, achieving an estimation of a robot's pose in real time on the application platform while continuously building a 3D environment map remains difficult.

In recent years, visual-inertial SLAM (VI-SLAM) [8]–[10] technology has emerged as a result of improved computing ability. Compared with other sensors or combinations, the combination of the stereo camera and the inertial measurement unit (IMU) has resulted in an overall superior performance, and comes with a range of advantages. First, VI-SLAM is able to obtain precise poses in situations where the global navigation satellite system signal fails. In this scenario, the robot is still able to construct a 3D environment map in either an unknown environment or an indoor environment. Second, the stereo camera is able to provide useful information and construct 3D environment scenes; it can be used for place recognition and loop closure. In addition, the stereo camera also offers information that enables the robot to sense scale and localization within an environment. Third, the IMU provides motion information that allows for the recovery of scale from the monocular. IMU can also estimate the direction of gravity and determine the precise pitch and roll of the sensor. Fourth, even when the robot is stationary or moving at a constant speed, the stereo camera can obtain scale information and suppress the drift caused by the IMU. Finally, compared with other sensor fusion methods, the stereo visual-inertial fusion method is less costly and consumes less energy than other methods, as it can utilize consumer-grade sensors to obtain accurate poses.

However, the combination of stereo camera and IMU still comes with its disadvantages. In scenarios that include severe motion, camera images are blurred and contain undesirable amounts of noise. Furthermore, the image obtained by this method is easily influenced by the texture information and the level of illumination in its environment. Moreover, owing to the biases of the IMU's accelerometer and gyroscope, data reliability is low during initialization, leading to the accumulation of errors over time.

In this paper we propose a novel stereo visual-inertial simultaneous localization and mapping method (STCM-SLAM) for use in mobile robots. Our approach is illustrated in Fig. 1. In addition, our implementation is available at https://github.com /cumtxz/STCM-SLAM.

Our method uses image tiling to extract image features and carry out forward backward optical flow in order to track features. We also propose one circle feature matching method to manage features. This marginalization strategy is designed to improve the system's overall robustness. Our experiments reveal that STCM-SLAM performs better than ORB-SLAM2 and OKVIS under real-time conditions. In addition, in the indoor experiments, STCM-SLAM can estimate the trajectory of the mobile robot with more accuracy than either ORB-SLAM2 or OKVIS.

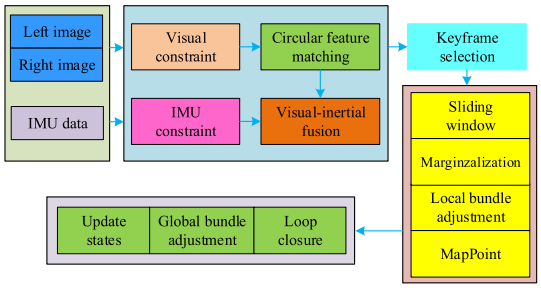


FIGURE 1. Structural architecture of our STCM-SLAM.

II. RELATED WORK

Visual SLAM technologies have achieved a range of valuable research results in the last few decades. Davison *et al.* [11] proposed a filtering-based SLAM system that was successfully used for monocular cameras in 2007, projecting scene points onto a probability ellipse, extracting Shi-Tomasi corners on the image, and using an extended Kalman filter (EKF) for optimization. PTAM (parallel tracking and mapping) [12] treated tracking and mapping as two separate threads, whereas tracking results did not depend on a specific probability's mapping process. By using a highly robust tracking algorithm, data association between tracking and mapping did not need to be shared, thus solving the computational burden when updating the map per frame. However, this method cannot provide scale estimation. Collaborative Visual SLAM (CoSLAM) [13] maintained the uncertainty of the location of each map point, and collaboratively constructed a global map using cameras on different platforms, which incorporated the static background and the trajectory of the pre-movement attraction. Mur-Artal and Tardós [14] proposed a sparse-featured ORB-SLAM system. This algorithm includes tracking, mapping, and loop closure threads. With ORB-SLAM, tracking loss could be recovered, and the real-time relocation has rotation invariance. Lin [15] proposed an automatic method for key-frame selection based on the motion state of a vehicle.

The above methods all follow a similar procedure: first, they extract features from images and then match image features in order to obtain matching points. The camera pose can then be estimated based on these points. In contrast, direct methods do not rely on the extraction and matching of features; rather they take samples from an image's pixels (including edges and pixels whose smooth intensity changes) to generate a more complete environmental model. This process minimizes photometric error and allows the camera motion to be estimated, improving the strength of weak texture features. Engel *et al.* [16] proposed LSD-SLAM, which is able to construct dense or semi-dense maps with photometric errors and geometric prior information. This method performs direct matching on the Sim3 lie group and uses

point clouds as representation, allowing it to construct a semi-dense 3D environment map in real time. Semi-direct visual odometry (SVO) [17] makes use of the direct method to calculate the initial estimation of camera motion and feature matching. This method also optimizes the feature-based non-linear re-projection error, and adopts optical flow to solve the localization problem. In addition, the relocalization residual is converted to a bundle adjustment (BA) problem, which is solved by an iterative nonlinear least square method. Dense tracking and mapping (DTAM) [18], LSD-SLAM, and SVO not only make use of geometric prior information, but also minimize photometric errors as they reconstruct dense environmental models. However, direct sparse odometry (DSO) [19] is able to directly optimize the photometric error without needing to consider the prior geometric information. In addition to perfecting the error model of direct pose estimation, DSO also includes additional features such as affine brightness transformation, photometric calibration and depth optimization.

RGB-D cameras not only provide environmental information, but also generate depth information for building dense 3D maps. SLAM methods that make use of RGB-D cameras have also been very successful. Dense visual SLAM (DVO-SLAM) [20] minimizes the photometric error and the depth error of pixels, as well as making better use of environmental information when compared with sparse methods or feature-based methods. Furthering this technology, an entropy-based keyframe selection and loop closure method was proposed. Dynamicfusion [21] established a model under a canonical frame, allowing changes of scene to be mapped to the model by geometric transformation. Each new depth map is merged into the model by way of geometric transformation. This method can be applied to a variety of moving objects and scenes without temporary or a priori scene models. Furthermore, ElasticFusion [22] fused the RGB-D camera, enabling it to estimate poses through the color consistency constraint. In addition, the localization of point clouds was estimated by the iterative closest points (ICP) algorithm. The surface element (surfel) model is used to detect the constraint optimization using global and local loop closure, and a dense map is constructed for AR platforms. Zhang *et al*. [23] proposed a semantic SLAM system that built semantic maps with object-level entities, which was integrated into the RGB-D SLAM framework. However, this system requires GPU acceleration.

SLAM methods combined with AprilTag [24] or artificial square markers [25] have also obtained rich results. Munoz-Salinas *et al*. [26] solved the problems of mapping and localization from a set of squared planar markers. However, this method is not incremental and needs to repeat the whole process from start in case of requiring the expansion of the map. DeGol *et al.* [27] presented an incremental structure from motion algorithm using fiducial markers matching, but this system cannot run real time. Sarmadi *et al*. [28] estimated the camera poses, the three-dimensional structure of planar markers and the relative pose between them. TagSLAM [29] leveraged AprilTags and the GTSAM factor graph optimizer [30] to obtain vision based ground truth poses and extrinsic calibration non-overlapping views. SPM-SLAM [31] initialized the map from a set of ambiguously detected markers seen from at least two different locations and proposed a method for loop closure detection and correction using squared planar markers. UcoSLAM [32] fused keypoints with squared fiducial markers in the SLAM system. However, these methods require a large number of artificial square markers to be placed in the scene. These methods can only be used in known environments and cannot be applied to unknown environments or large outdoor environments.

Currently, multi sensor fusion has become a popular form of SLAM technology, with VI-SLAM methods the focus of contemporary research. Multi-state constraint Kalman filter (MSCKF) [33] is a classic VI-SLAM system, consisting of a multi-state constrained visual-inertial navigation system based on an EKF filter. This method obtains a measurement model for expressing the geometric constraints produced when a static feature is observed by multiple cameras. This measurement model does not require the inclusion of a 3D feature position in the EKF filter's state vector and is able to obtain optimal linear error. In addition, Li and Mourikis [34], [35] demonstrated that the standard method of calculating the filter's Jacobian matrix would inevitably leads to inconsistency and a loss of precision. MSCKF has been improved in order to ensure it possesses the correct observation properties without incurring additional computational costs. However, this method does not provide loop closure. Tanskanen *et al*. [36] combined MSCKF with the advantages of an EKF filter, minimizing the photometric error and presenting a direct visual-inertial odometry that can run in real time on the CPU. Bloesch *et al*. [37] proposed a visual-inertial odometry based on the monocular named ROVIO. This made use of image-matched pixel photometric errors, allowing it to achieve accurate, robust tracking results. In addition, the FAST corner [38] was employed to identify a large number of candidate features. A multi-layer image pyramid extracts multi-layer features or pixel blocks, and edge features are also added. However, despite such advances in SLAM technologies, achieving robust and accurate visual-inertial estimations still remains a challenge in the robotics field.

The above VI-SLAM methods are all filter-based methods. As computer technology has advanced, however, optimization-based VI-SLAM methods (which adopt non-linear optimization for pose estimation and map construction) have also begun to attract attention. OKVIS [39] is an optimization-based VI-SLAM method based on keyframes. This method constructs a loss function, fuses the re-projection error term and the IMU error term, and maintains a sliding window by marginalizing old keyframes. SOFT-SLAM [40] is a stereo SLAM system that relies on special features; it uses SOFT visual odometry instead of BA for pose estimation, selects high quality features using circle matching, and loosely couples the visual and gyroscope data. Moreover, VINS-Mono [41] is also considered to be an excellent

VI-SLAM method; with the system's front-end tracking Harris corners [42] based on optical flow, and the back-end using sliding window and loop closure for nonlinear optimization. The spherical camera model is employed at the front-end, and outliers of the fundamental matrix are removed by the random sample consensus (RANSAC) method. ICE-BA [43] offers a relative marginalization method to improve global consistency. Techniques for fusing IMU data in classic visual SLAM frameworks have also been highly successful. Mur-Artal *et al*. [44] proposed a novel IMU initialization method based on the original ORB-SLAM2 framework [14], which calculated the scale, gravity direction, velocity, and deviation of the gyroscope and accelerometer. VI-DSO [45] is a direct sparse visual-inertial odometry system based on DSO, where a dynamic marginalization strategy is used to partially marginalize old variables, which can then be calculated in a reasonable amount of time. The initial scale estimation is far from optimal but satisfactory results are still obtained. Liu *et al*. [46] used both points and lines to increase the robustness of their visual–inertial SLAM system.

Progress has also been made in applying deep learning to parts of the SLAM system. Gomez-Ojeda *et al*. [47] proposed two different deep neural networks that enhanced monocular images to more informative representations for visual odometry. In addition, Li *et al*. [48] used deep neural networks to estimate the 6-DoF pose of a monocular camera and the depth of its view, and proposed a novel monocular VO system based on unsupervised deep learning scheme. DeTone *et al*. [49] presented a self-supervised framework to train interest point detectors and descriptors suitable for a large number of multiple-view geometry problems named SuperPoint. However, these methods impose a large computational burden, especially for low-power devices such as mobile robots or MAVs.

III. POSE ESTIMATION

This section establishes the error function of the stereo camera and IMU according to motion states of the mobile robot. In addition, visual and IMU constraints are also discussed.

A. ERROR FUNCTION

The mobile robot mainly performs translation in the direction of the x-axis during motion, and the amount of rotational change is small. If pose estimation is performed using only the robot's vision, the pose changes would not be fully perceived on the road's surface due to the presence of repeated textures or variations in gradient. To counter this problem, we propose a strategy of tightly coupling the camera and IMU data, allowing more accurate estimation of the pose and velocity of the camera, as well as the IMU bias. Because the acceleration calculated by the consumer-grade IMU is prone to errors, the rotation of the IMU is not as a state variable of the system. The 15-dimensional state variables of the system in time i are defined as:

$$\mu_i = [R_i, p_i, v_i, b_{ai}, b_{gi}] \in \mathbb{R}^{15} \tag{1}$$

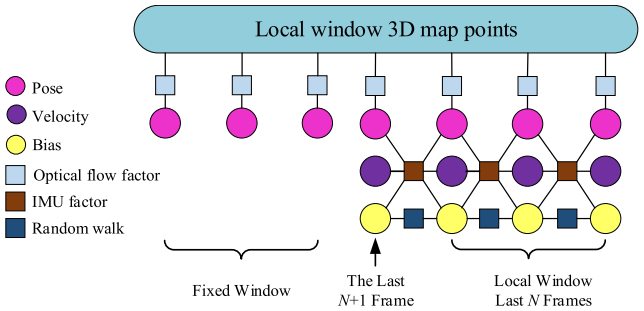


FIGURE 2. The schematic diagram of the sliding window.

where R, p and v are the rotation, translation and speed of the camera, respectively, and b_a and b_g are the bias of the accelerometer and gyroscope, respectively. The pose (R, p) belong to $SE(3)$, $v, b_a, b_g \in \mathbb{R}^3$.

Optimization-based SLAM systems are generally incremental motion estimation systems, where the rotation and translation errors are continuously accumulated. Therefore the loop closure mechanism is used in accordance with [14]. When a robot is running within the same scene, all poses are optimized using loop closure, and sliding window optimization is constructed to optimize the keyframe pose. Loop closure and keyframe selection are based on our previous work [3]. We present the factor graph [50] of the sliding window in Fig. 2.

To construct the optimization equation for the sliding window, we integrate visual measurements, IMU measurements, and the *a priori* marginalization. The optimization objective equation is as follows:

$$\hat{\mu} = \underset{\mu}{\operatorname{argmin}}\Big(\sum_{(l,j)\in C} ||e_c(\mu)||^2_{P_l^j} + \sum_{k\in B} ||e_{IMU}(\mu)||^2_{P_{k+1}^k} + ||e_{marg}(\mu)||_2^2\Big) \tag{2}$$

where $e_c(\mu)$, $e_{IMU}(\mu)$ and $e_{marg}(\mu)$ are the error (residual) of visual measurements, IMU measurements, and the *a priori* marginalization, respectively. P_l^j and P_{k+1}^k are the covariance of visual measurements and IMU measurements, respectively. It should be noted that the errors are determined using the Mahalanobis distance, which are weighted error terms. The surge of a single error term can be effectively suppressed. Equation 2 is solved using the Gauss-Netwon algorithm.

B. VISUAL CONSTRAINT

The classic pinhole camera model [51] is used to transform the 3D space point, $\mathrm{X}^i \in \mathbb{R}^3$, under the camera frame into the 2D point, x^i, under the image frame. The coordinates in the left and right images are $u_l = (u_l, v_l)$ and $u_r = (u_r, v_r)$, respectively. In addition, images are rectified. Assuming that v_l is equal to v_r, we define the three coordinates as $\mathrm{x_s} = (u_l, v_l, u_r)$. We build projection function $\pi_s(\cdot)$ according to [14]:

$$\mathrm{x_s} = \pi_s(\mathrm{X}) = \begin{bmatrix} f_x \dfrac{X}{Z} + c_x \\ f_y \dfrac{Y}{Z} + c_y \\ f_x \dfrac{X - b}{Z} + c_x \end{bmatrix} \tag{3}$$

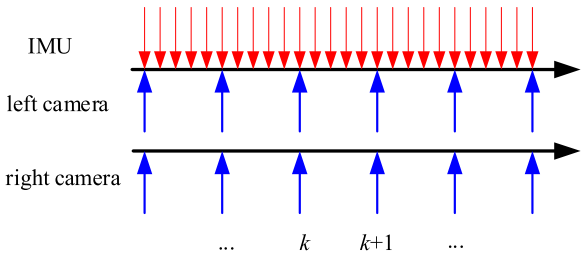


FIGURE 3. The frequency of the IMU and stereo camera.

where $X = (X, Y, Z)$, (f_x, f_y) is the focal length, and b is the baseline of the stereo camera. (c_x, c_y) is the principal point of the camera. In this investigation, we use the re-projection error [51] to represent the visual error term.

$$e_c(\mu) = \mathrm{x}^i - \pi_s(R_i X^i + p_i) \tag{4}$$

C. IMU CONSTRAINT

The measurements of the gyroscope and accelerometer for the body coordinate system at the k-time can be expressed as:

$$\tilde{w}_{b_k} = w_{b_k} + b^g_{b_k} + \eta^g_{b_k} \tag{5}$$

$$\tilde{a}_{b_k} = R^T_{wb}(a_{b_k} - g) + b^a_{b_k} + \eta^a_{b_k} \tag{6}$$

where $b^g_{b_k}$ and $\eta^g_{b_k}$ are the bias and white noise of the gyroscope; $b^a_{b_k}$ and $\eta^a_{b_k}$ are the bias and white noise of the accelerometer; and w_{b_k} and a_{b_k} represent the truth value of gyroscope and accelerometer, respectively.

$$\dot{b}^g_{b_k} = \eta^{bg}_{b_k} \tag{7}$$

$$\dot{b}^a_{b_k} = \eta^{ba}_{b_k} \tag{8}$$

where $\eta^{bg}_{b_k}$ and $\eta^{ba}_{b_k}$ obey zero-mean Gaussian distribution.

The stereo camera provides low-frequency data, while the IMU provides high-frequency data, as shown in Fig. 3. Since the previous frame's state changes during the optimization process, the integration must be recalculated when the initial state of integration changes. To avoid repeatedly calculating the IMU integral after each optimization adjustment, we constrain the relative motion using pre-integration, then parameterized according to the relevant literature [52], [53]. The median discrete form of rotation, velocity, and translation of the IMU at the $k+1$ time are defined as follows:

$$R_{b_{k+1}} = R_{b_k} \mathrm{Exp}((\tilde{w}_{b_k} - b^g_{b_k} - \eta^{bg}_{b_k})\Delta t) \tag{9-1}$$

$$v_{b_{k+1}} = v_{b_k} + g\Delta t + R_{b_{k+1}}(\tilde{a}_{b_k} - b^a_{b_k} - \eta^{ba}_{b_k})\Delta t \tag{9-2}$$

$$p_{b_{k+1}} = p_{b_k} + v_{b_k}\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R_{b_k}(\tilde{a}_{b_k} - b^a_{b_k} - \eta^{ba}_{b_k})\Delta t^2 \tag{9-3}$$

where Δt is the time instant difference between the previous and current IMU data. The increments of the rotation, velocity, and translation of IMU from $k+1$ to k are defined as:

$$\Delta R^{b_k}_{b_{k+1}} \doteq \prod_{k=i}^{j-1} Exp((\tilde{w}_{b_k} - b^g_{b_k} - \eta^{bg}_{b_k})\Delta t) \tag{10-1}$$

$$v^{b_k}_{b_{k+1}} \doteq \sum_{k=i}^{j-1} R^{b_k}_{b_i}(\tilde{a}_{b_k} - b^a_{b_k} - \eta^{ba}_{b_k})\Delta t \tag{10-2}$$

$$\Delta p \doteq \sum_{k=i}^{j-1} \frac{3}{2}\Delta R^{b_k}_{b_i}(\tilde{a}_{b_k} - b^a_{b_k} - \eta^{ba}_{b_k})\Delta t^2 \tag{10-3}$$

This study defines the IMU error as:

$$e_{IMU}(\mu) = \begin{bmatrix} \delta\alpha^{b_k}_{b_{k+1}} \\ \delta\theta^{b_k}_{b_{k+1}} \\ \delta\beta^{b_k}_{b_{k+1}} \\ \delta b_a \\ \delta b_g \end{bmatrix} \times \begin{bmatrix} R_{b_k}(p_{b_{k+1}} - p_{b_k} - v_{b_k}\Delta t + \frac{1}{2}g\Delta t^2) - \alpha^{b_k}_{b_{k+1}} \\ 2]\gamma^{b_{k-1}}_{b_{k+1}} \otimes q^{-1}_{b_k} \otimes q_{b_{k+1}}]_{xyz} \\ R_{b_k}(v_{b_{k+1}} - v_{b_k} + g\Delta t) - \beta^{b_k}_{b_{k+1}} \\ b^g_{b_{k+1}} - b^g_{b_k} \\ b^a_{b_{k+1}} - b^a_{b_k} \end{bmatrix} \tag{11}$$

where $\delta\alpha^{b_k}_{b_{k+1}}$, $\delta\theta^{b_k}_{b_{k+1}}$, and $\delta\beta^{b_k}_{b_{k+1}}$ represent the residual of rotation, translation, and velocity, respectively. δb_a, and δb_g are the bias residual of accelerometer and gyroscope, respectively.

IV. FEATURE TRACKING

This section focuses on feature tracking of our STCM-SLAM. Forward backward optical flow and circle feature matching are used to improve the accuracy of system.

A. FORWARD BACKWARD OPTICAL FLOW

For this investigation, we use image tiling to extract features, such as dividing an image into 25×25 image blocks, extracting a FAST feature for each image block in order to make the feature distribution more uniform, and then using optical flow to track features between the two frames. The classic tracking algorithm, optical flow is a classic optical flow tracking algorithm and has been used in a number of SLAM methods [17], [41], This algorithm assumes three things: first, the target image has uniform brightness; second, the image space is continuous; and third, the image is continuous in time. $I(u, v, t)$ represents the grayscale value of the image pixel at time t. Based on the above assumptions, the brightness conservation equation of pixels can be obtained:

$$I_x\dot{u} + I_y\dot{v} + I_t(x, y) = 0 \tag{12}$$

The three assumptions of optical flow are easy to satisfy when a robot is performing small motions. However, the mobile robot often moves quickly with strong rotations, which means that optical flow tracking is easy to lose. To counter this difficulty, we implement the more accurate forward backward bidirectional optical flow tracking (FB-LK). For two images, I_k and I_{k+1} on the time constraint, we first extract feature set A for the image I_k. Next, we obtain feature set B based on the corresponding feature set A in image, I_k. Then, we

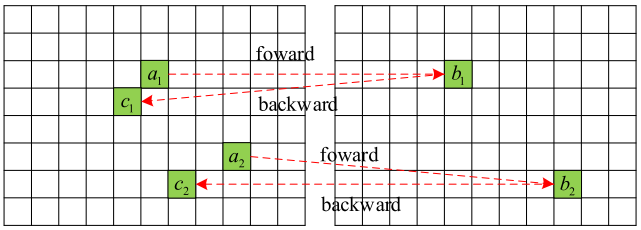


FIGURE 4. Forward backward bidirectional optical tracking diagram.

TABLE 1. Comparing the forward backward optical flow and correlation flow.

Forward backward			Correlation flow		
ATE(m)	RPE(mm)	Time(ms)	ATE(m)	RPE(mm)	Time(ms)
0.217	**6.817**	25.589	0.233	7.260	**20.002**

obtain feature set B based on the corresponding feature set A in image, I_{k+1}. This is done by using optical flow, which is referred to as forward tracking. Next, feature set B returns to track the corresponding feature set C in the image, I_k,; this is referred to as backward tracking. The forward backward bidirectional optical flow tracking diagram is shown in Fig. 4, with the square representing the feature position, and the dot arrow indicating the tracking direction.

Due to the influence of image noise, the features tracked by the forward backward method are occasionally inconsistent. Therefore, matching points are filtered according to the distance threshold. We set the distance threshold to 0.5 pixel. In addition, we adopt a higher frame rate of 20Hz, as well a multiple-level optical flow in order to improve the robustness of the tracking technology. The effect of image feature tracking in this method is shown in Fig. 5.

In this investigation we use the absolute trajectory error (ATE) and the relative pose error (RPE) to compare the accuracy of frame poses inspired by [54], [55]. Table 1 compares correlation flow [56] and forward backward optical flow on Machine Hall 01 of EuRoC dataset [57]. We calculate the root mean square error (RMSE) of the ATE and RPE of frame poses, as well as the mean inter-frame processing time.

$$\mathrm{ATE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||P_i - P_i^{gt}||^2} \tag{13}$$

$$\mathrm{RPE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1} (||P_{i+1} - P_i|| - ||P_{i+1}^{gt} - P_i^{gt}||)^2} \tag{14}$$

where P_i, and P_I^{gt} represent the estimated pose and the ground-truth pose, respectively. The ATE and RPE of frame poses increase 7.373% and 6.498%, respectively. Despite the mean of the inter-frame processing time increasing by 6.688 ms, the system is still able to achieve superior real-time performance.

The complete accuracy curve of frame poses' ATE and RPE across forward backward optical flow and correlation flow are shown in Fig. 6. As shown in Fig. 6(a), the ATE of forward backward optical flow is a priori to correlation flow. Besides, the RPE of forward backward optical flow is more stable than that achieved using correlation flow, as can be seen in Fig. 6(b).

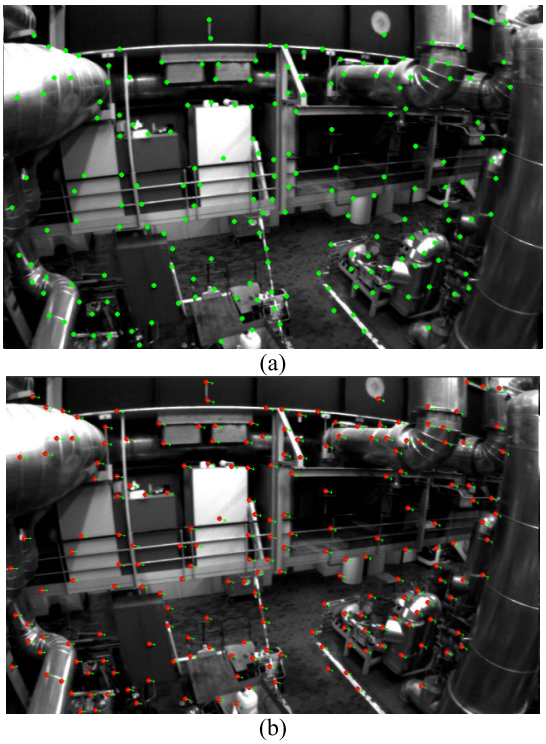
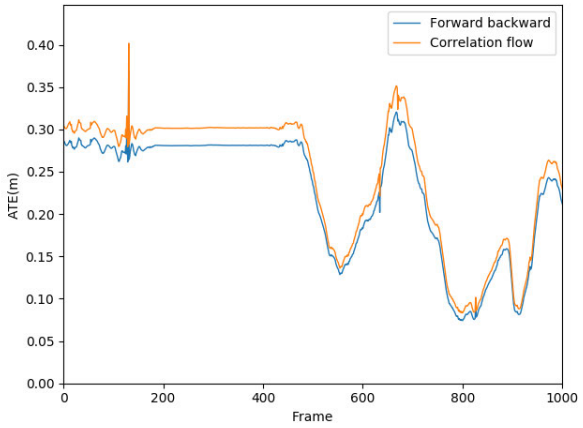


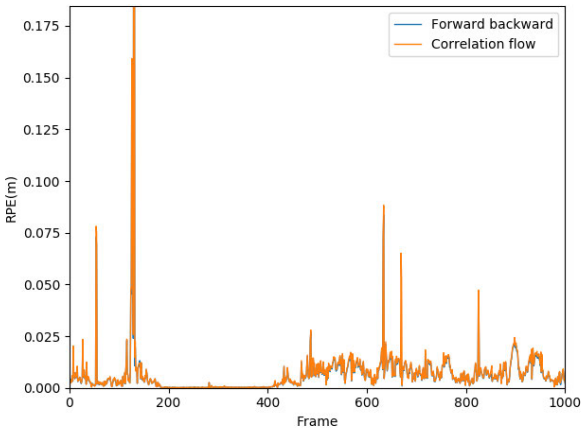
FIGURE 5. Forward backward bidirectional optical tracking result. (a) Uniform distribution of image features. (b) Optical flow tracking image.

B. CIRCLE FEATURE MATCHING

Motion can often contribute to a loss of feature tracking. To counteract this problem, we use the circle feature matching method. This allows us to both manage and improve the quality of features, thereby improving the robustness of textureless and repeated texture scenes. Inspired by previous investigations [40], we use continuous image tracking to increase the circle constraint. After the FAST features are extracted using blocks, the features are determined using circle matching and outliers are removed. For the images produced by the stereo camera in continuous time, we implement time and space constraints. The constraint between the front and back image on the same camera is a temporal constraint, and the constraint between the left and right image of the stereo camera is a spatial constraint. Forward backward optical flow is used to track the adjacent time frames, and Lucas Kanade optical flow is used to track the left and right images simultaneously. For the k-th image, the sequence of the circle matching for the time after space (STCM) is $I_k^l \rightarrow I_k^r \rightarrow I_{k+1}^r \rightarrow I_{k+1}^l \rightarrow I_k^l$. Both sequences are displayed in Fig. 7. The circle feature matching method we used is able to extract features and track them with optical flow without



(a)



(b)

FIGURE 6. ATE and RPE comparison between forward backward optical flow and correlation flow. (a) ATE of frame poses. (b) RPE of frame poses.

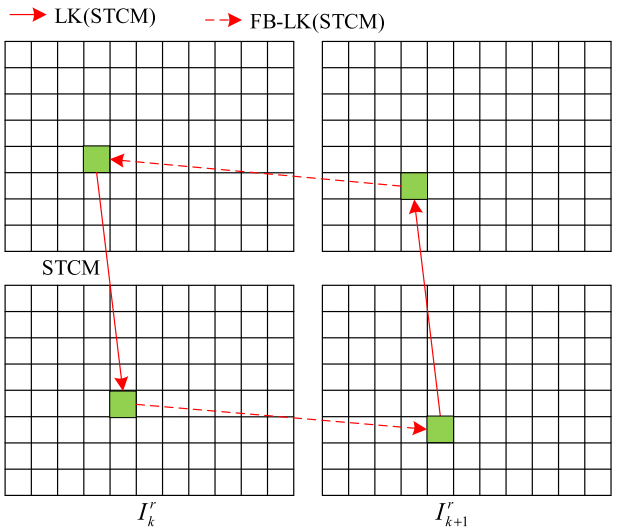


FIGURE 7. The schematic diagram of the STCM method for the stereo camera image features.

any additional computation. After circle matching, the high-quality features are filtered to improve the tracking accuracy of features.

TABLE 2. Comparing parameters across the STCM and correlation flow.

Methods	ATE(m)	RPE(mm)	Time(ms)
STCM	**0.169**	**3.168**	23.859
Forward backward	0.217	6.817	25.589
Correlation flow	0.233	7.260	**20.002**

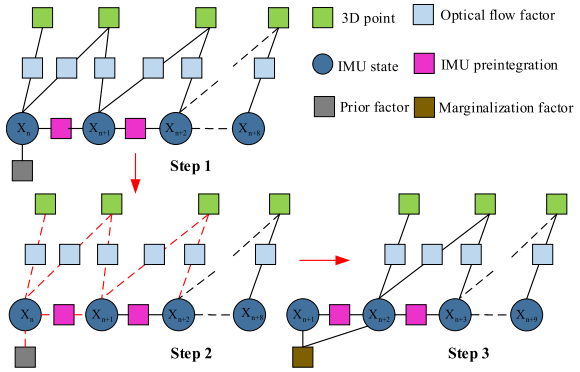


FIGURE 8. System marginalization process.

The parameter comparison between STCM and correlation flow methods is shown in Table 2. The inter-frame processing time of STCM is higher than that achieved with correlation flow, but is lower than the processing times achieved using the forward backward technique. In Table 2, the absolute accuracy and relative accuracy of STCM can be seen to increase by 37.869% and 129.167% respectively, when compared with correlation flow. After this comparison, we select the STCM method to use for feature tracking.

V. MARGINALIZATION

For this investigation, we adopt one marginalization strategy in the sliding window, as shown in Fig. 8. The number of keyframes in the sliding window is set to 9 to balance accuracy and calculation according to [3].

The entire marginalization process is distributed across three steps. As shown in Step 1, keyframes in the sliding window are numbered from X_n to X_{n+8}. When a new keyframe X_{n+9} is inserted into the sliding window, the oldest keyframe X_n is marginalized. The 3D point, optical flow factor, IMU state, and pre-integration constraints of X_n are converted into the marginalization factor, which does not lose any constraints. The marginalization process of error function can be shown as follows:

$$\begin{bmatrix} \Lambda_{rr} & \Lambda_{rm} \\ \Lambda_{rm}^T & \Lambda_{mm} \end{bmatrix} \begin{bmatrix} \delta x_r \\ \delta x_m \end{bmatrix} = \begin{bmatrix} -g_r \\ -g_m \end{bmatrix} \tag{15}$$

where δx_r is the portion that needs to be preserved from marginalization and δx_m is the portion that needs to be marginalized, respectively. Next, we use Shure to eliminate the element:

$$\begin{bmatrix} I & -\Lambda_{rm}\Lambda_{mm}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Lambda_{rr} & \Lambda_{rm} \\ \Lambda_{rm}^T & \Lambda_{mm} \end{bmatrix} \begin{bmatrix} \delta x_r \\ \delta x_m \end{bmatrix} = \begin{bmatrix} I & -\Lambda_{rm}\Lambda_{mm}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} -g_r \\ -g_m \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \Lambda_{rr} - \Lambda_{rm}\Lambda_{mm}^{-1}\Lambda_{rm}^{T} & 0 \\ \Lambda_{rm}^{T} & \Lambda_{mm} \end{bmatrix} \begin{bmatrix} \delta x_r \\ \delta x_m \end{bmatrix} = \begin{bmatrix} -g_r + \Lambda_{rm}\Lambda_{mm}^{-1} g_m \\ -g_m \end{bmatrix} \tag{16}$$

where $\Lambda_{rm}\Lambda_{mm}^{-1}\Lambda_{rm}^{T}$ are the Shure complement of Λ_{mm} in Λ_{rm}. Thus, the prior information equation of δx_r is:

$$(\Lambda_{rr} - \Lambda_{rm}\Lambda_{mm}^{-1}\Lambda_{rm}^{T})\delta x_r = -g_r + \Lambda_{rm}\Lambda_{mm}^{-1} g_m \tag{17}$$

The prior marginalization error is $\mathrm{e}_{marg}(\mu) = \delta x_r$. In the optimization objective, represented in equation 2, the minimum value must first be calculated, and then further converted to the minimum value of the objective function, where the optimization variable has an increment, and the incremental equation is obtained.

$$\begin{aligned} &(\sum H_l^{c_j T} P_l^{c-1} H_l^{c_j} + \sum H_{b_{k+1}}^{b_k T} P_{b_{k+1}}^{b_k - 1} H_{b_{k+1}}^{b_k} + \Lambda_p)\delta\mu \\ &\quad = \sum H_l^{c_j\ T} P_l^{c-1} \mathrm{e}_c(\mu) + \sum H_{b_{k+1}}^{b_k\ T} P_{b_{k+1}}^{b_k - 1} \mathrm{e}_{IMU}(\mu) + b_p \end{aligned} \tag{18}$$

The marginalization factor generated in the third step becomes the prior factor in the next marginalization. Only keyframes from X_{n+1} to X_{n+9} are exist in the sliding window, waiting for the next keyframe.

VI. EXPERIMENTS

This section documents experiments comparing our STCM-SLAM method with state-of-the-art methods on EuRoC datasets and on the mobile robot in an indoor environment.

A. LOCALIZATION ACCURACY

ORB-SLAM2 and OKVIS are state-of-the-art methods used in visual SLAM and VI-SLAM, respectively. In order to effectively assess the performance of our proposed method, we compare STCM-SLAM with ORB-SLAM2 and OKVIS. The EuRoC dataset is a well-known SLAM dataset that has been used to test the localization accuracy of visual SLAM or visual-inertial SLAM methods. We determine that this dataset is the best choice for comparing STCM-SLAM, ORB-SLAM2, and OKVIS, although it is acquired by a micro-aerial vehicle. In addition, the dataset provides stereo camera images, IMU data, and the ground-truth of the robot's motion. The image resolution is 752 × 480, the frequency is 20 Hz, and the IMU frequency is 200 Hz. Especially, the first batch of the dataset is recorded in the ETH machine hall, which is a largely unknown environment for the robots. For our comparative experiment, we adopt ORB-SLAM2's stereo mode, and set default parameters of ORB-SLAM2 (https://github.com/raulmur/ORB_SLAM2) and OKVIS (https://github.com/ethz-asl/okvis). Experiments are performed on an Intel Core i7-6700 × 8 computer equipped with 16 Gb RAM, with an Ubuntu 18.04 LTS operating system. Because the system uses a multi-threaded design, calculation results are susceptible to the allocation of computing resources. To counter this, each dataset repeatedly ran 5 times and got the intermediate value of results.

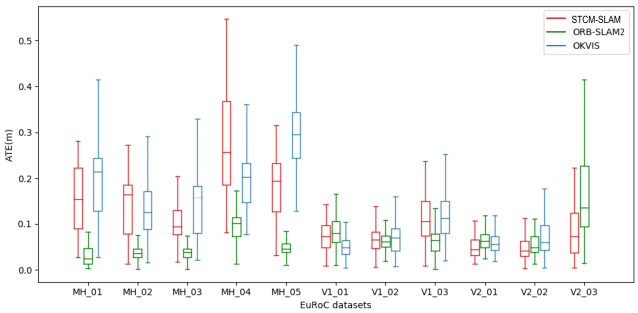


FIGURE 9. The comparison of absolute trajectory error using the EuRoC datasets.

We use ATE and RPE to evaluate the trajectory accuracy. In addition, the mean error, median error, minimum value, RMSE, and standard deviation (STD) of the pose error are calculated according to evo (https://github.com/MichaelGrupp/evo). Table 3 and Table 4 show the ATE and RPE of STCM-SLAM, ORB-SLAM2 and OKVIS, respectively. It should be noted that the subscript of the data indicates the order, and that OKVIS fails to run under the V2_03 dataset. ORB-SLAM2 performs best in terms of absolute trajectory error. In addition, the mean error, median error, minimum value, RMSE, and STD of ORB-SLAM2 are the smallest on MH_01, MH_02, MH_03, and MH_05, as can be seen in Table 3. The overall accuracy of STCM-SLAM is less than that of ORB-SLAM2, but the STCM-SLAM method generates the fewest errors on the V2_03 dataset and has good accuracy on the V1_01, V2_01, and V2_02 datasets. Furthermore, STCM-SLAM's range of error distribution is the smallest on V2_01 and V2_03. This shows that our method can achieve great localization accuracy when the robot moves vigorously.

STCM-SLAM also achieves the best results in terms of relative pose error, with the lowest values for mean error, median error, minimum value, RMSE, and STD across a total of 11 datasets, as shown in Table 4. The average RMSE of absolute trajectory error is only 0.008m. Furthermore, the accuracy of STCM-SLAM is an order of magnitude greater than that of ORB-SLAM2 and two orders of magnitude greater than OKVIS. This is due to our proposed circle feature matching method, achieving the smallest relative pose error and tracking more robust.

The absolute trajectory error and relative pose error of these three methods are shown in Fig. 9 and Fig. 10, respectively. As shown in Fig. 9, ORB-SLAM has a wide range of error distribution on V1_01 and V2_03, and our method achieves best results on these two sequences. Fig. 10 shows that STCM-SLAM achieves the best results with the smallest range of values. OKVIS had the widest error range and the largest error value. The maximum error of OKVIS exceeds 0.6m on many sequences. Based on the above analysis, one may reasonably conclude that the proposed method has good localization accuracy and minimal drift.

TABLE 3. Absolute trajectory error of STCM-SLAM, ORB-SLAM2 and OKVIS (m).

Sequences	STCM-SLAM				ORB-SLAM2				OKVIS			
	mean	median	RMSE	STD	mean	median	RMSE	STD	mean	median	RMSE	STD
MH_01	0.153_2	0.153_2	0.169_2	0.072_2	$\mathbf{0.031_1}$	$\mathbf{0.024_1}$	$\mathbf{0.036_1}$	$\mathbf{0.020_1}$	0.207_3	0.215_3	0.235_3	0.112_3
MH_02	0.141_2	0.165_3	0.154_2	0.063_2	$\mathbf{0.040_1}$	$\mathbf{0.036_1}$	$\mathbf{0.045_1}$	$\mathbf{0.020_1}$	0.143_3	0.125_2	0.168_3	0.089_3
MH_03	0.111_2	0.094_2	0.127_2	0.061_2	$\mathbf{0.038_1}$	$\mathbf{0.039_1}$	$\mathbf{0.041_1}$	$\mathbf{0.016_1}$	0.181_3	0.157_3	0.227_3	0.210_3
MH_04	0.273_3	0.257_3	0.293_2	0.109_2	$\mathbf{0.099_1}$	$\mathbf{0.101_1}$	$\mathbf{0.107_1}$	0.040	0.226_2	0.204_2	0.294_3	0.187_3
MH_05	0.184_2	0.195_2	0.195_2	0.064_2	$\mathbf{0.055_1}$	$\mathbf{0.046_1}$	$\mathbf{0.064_1}$	$\mathbf{0.034_1}$	0.316_3	0.296_3	0.348_3	0.146_2
V1_01	0.072_2	0.073_2	$\mathbf{0.079_1}$	$\mathbf{0.030_1}$	0.082_3	0.080_3	0.087_2	0.031_2	$\mathbf{0.060_1}$	$\mathbf{0.049_1}$	0.088_3	0.065_3
V1_02	0.083_2	0.065_2	0.113_2	0.076_2	$\mathbf{0.063_1}$	$\mathbf{0.061_1}$	$\mathbf{0.065_1}$	$\mathbf{0.017_1}$	0.102_3	0.070_3	0.1890_3	0.1590_3
V1_03	0.110_2	0.106_2	0.120_2	$\mathbf{0.047_1}$	$\mathbf{0.072_1}$	$\mathbf{0.065_1}$	$\mathbf{0.092_1}$	0.058_2	0.137_3	0.112_3	0.182_3	0.119_3
V2_01	$\mathbf{0.057_1}$	$\mathbf{0.044_1}$	0.084_3	0.062_3	0.066_2	0.063_3	$\mathbf{0.071_1}$	$\mathbf{0.025_1}$	0.068_3	0.055_2	0.082_2	0.046_2
V2_02	$\mathbf{0.050_1}$	$\mathbf{0.042_1}$	$\mathbf{0.060_1}$	0.033_2	0.056_2	0.049_2	0.061_2	$\mathbf{0.025_1}$	0.086_3	0.060_3	0.133_3	0.102_3
V2_03	$\mathbf{0.084_1}$	$\mathbf{0.072_1}$	$\mathbf{0.099_1}$	$\mathbf{0.052_1}$	0.159_2	0.136_2	0.184_2	0.092_2	\	\	\	\
Average	0.120_2	0.115_2	0.136_2	0.061_2	$\mathbf{0.069_1}$	$\mathbf{0.064_1}$	$\mathbf{0.078_1}$	$\mathbf{0.034_1}$	0.153_3	0.135_3	0.195_3	0.124_3

TABLE 4. Relative pose error of STCM-SLAM, ORB-SLAM2 and OKVIS (m).

Sequences	STCM-SLAM				ORB-SLAM2				OKVIS			
	mean	median	RMSE	STD	mean	median	RMSE	STD	mean	median	RMSE	STD
MH_01	$\mathbf{0.005_1}$	$\mathbf{0.004_1}$	$\mathbf{0.007_1}$	$\mathbf{0.004_1}$	0.023_2	0.021_2	0.028_2	0.016_2	0.196_3	0.178_3	0.251_3	0.157_3
MH_02	$\mathbf{0.005_1}$	$\mathbf{0.004_1}$	$\mathbf{0.006_1}$	$\mathbf{0.004_1}$	0.024_2	0.023_2	0.028_2	0.015_2	0.207_3	0.190_3	0.261_3	0.159_3
MH_03	$\mathbf{0.009_1}$	$\mathbf{0.008_1}$	$\mathbf{0.010_1}$	$\mathbf{0.006_1}$	0.056_2	0.047_2	0.071_2	0.045_2	0.447_3	0.364_3	0.604_3	0.406_3
MH_04	$\mathbf{0.009_1}$	$\mathbf{0.008_1}$	$\mathbf{0.011_1}$	$\mathbf{0.007_1}$	0.054_2	0.045_2	0.070_2	0.044_2	0.416_3	0.363_3	0.573_3	0.393_3
MH_05	$\mathbf{0.008_1}$	$\mathbf{0.007_1}$	$\mathbf{0.009_1}$	$\mathbf{0.005_1}$	0.049_2	0.043_2	0.062_2	0.038_2	0.364_3	0.293_3	0.481_3	0.314_3
V1_01	$\mathbf{0.007_1}$	$\mathbf{0.006_1}$	$\mathbf{0.008_1}$	$\mathbf{0.004_1}$	0.022_2	0.020_2	0.026_2	0.013_2	0.166_3	0.138_3	0.207_3	0.123_3
V1_02	$\mathbf{0.006_1}$	$\mathbf{0.006_1}$	$\mathbf{0.007_1}$	$\mathbf{0.003_1}$	0.047_2	0.046_2	0.053_2	0.025_2	0.360_3	0.308_3	0.445_3	0.262_3
V1_03	$\mathbf{0.006_1}$	$\mathbf{0.005_1}$	$\mathbf{0.009_1}$	$\mathbf{0.007_1}$	0.039_2	0.034_2	0.047_2	0.027_2	0.342_3	0.289_3	0.445_3	0.285_3
V2_01	$\mathbf{0.003_1}$	$\mathbf{0.002_1}$	$\mathbf{0.005_1}$	$\mathbf{0.003_1}$	0.017_2	0.015_2	0.019_2	0.010_2	0.132_3	0.122_3	0.160_3	0.091_3
V2_02	$\mathbf{0.004_1}$	$\mathbf{0.003_1}$	$\mathbf{0.005_1}$	$\mathbf{0.003_1}$	0.036_2	0.032_2	0.043_2	0.023_2	0.297_3	0.250_3	0.363_3	0.208_3
V2_03	$\mathbf{0.006_1}$	$\mathbf{0.004_1}$	$\mathbf{0.010_1}$	$\mathbf{0.008_1}$	0.055_2	0.035_2	0.209_2	0.201_2	\	\	\	\
Average	$\mathbf{0.006_1}$	$\mathbf{0.005_1}$	$\mathbf{0.008_1}$	$\mathbf{0.005_1}$	0.039_2	0.033_2	0.060_2	0.042_2	0.293_3	0.250_3	0.379_3	0.240_3

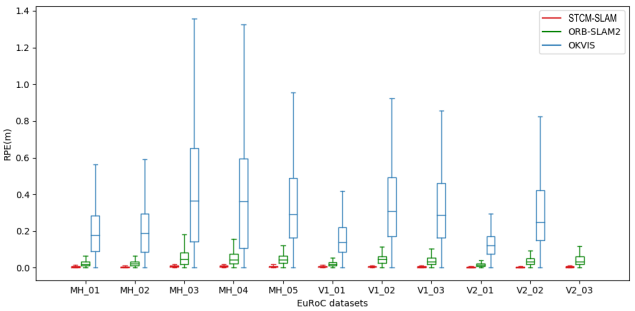


FIGURE 10. The comparison of relative pose error using the EuRoC datasets.

B. PARAMETERS EVALUATION

To assess the effectiveness of these methods, we compare the performances of STCM-SLAM, ORB-SLAM2, and OKVIS in a number of areas, such as scale error (SE), running frame frequency, CPU load (CL), and memory load (ML). We found that the CPU and memory resources are 100% occupied when fully used. We also remove the influence of the computer system from offline experiment results. The original image rate of datasets is 20 Hz, and we make use of a multi-speed mode in order to examine the real-time performance of these methods. When the frame rate is less than 20 Hz, the runtime algorithm experiences data delay. The performance results of the three methods, scale error (%), operating frequency (Hz), CPU load (%), and memory load (%), are shown in Table 5. Unfortunately, OKVIS is unable to provide the scale error for the V2_03 dataset, so the average scale error for OKVIS is an average of the data from the MH_01 to the V2_02 datasets.

In this study we analyze the scale error, running frequency, CPU load, and memory load of STCM-SLAM, ORB-SLAM2, and OKVIS, as presented in Table 5 and Fig. 11. In terms of scale error, ORB-SLAM2 has the best scale accuracy and achieves excellent performances on most of the datasets. STCM-SLAM also possesses good scale estimation and performs better than OKVIS. In terms of running frequency, STCM-SLAM runs at the highest frequency, averaging 36.070 Hz. OKVIS is also able to run in real time, except when applied to the V1_02 dataset. However, ORB-SLAM2 is unable to run in real time for any of datasets in the experiment, and the average frequency it achieved is only 10.165 Hz. As can be seen in Fig. 11(c), STCM-SLAM is inferior to ORB-SLAM2 but better than OKVIS in terms of CPU load. In terms of memory load, STCM-SLAM and ORB-SLAM2 perform similarly, and both require more memory than OKVIS.

Based on the above analyses, it is clear that STCM-SLAM achieves excellent relative pose estimation with the least amount of inter-frame drift, a result significantly better than

TABLE 5. Performance results of STCM-SLAM, ORB-SLAM, and OKVIS.

Datasets	STCM-SLAM				ORB-SLAM2				OKVIS			
	SE	Frequency	CL	ML	SE	Frequency	CL	ML	SE	Frequency	CL	ML
MH_01	0.518$_2$	**36.721$_1$**	43.225$_2$	21.922$_2$	**0.316$_1$**	12.438$_3$	**25.142$_1$**	23.476$_3$	1.372$_3$	23.846$_2$	47.323$_3$	**11.032$_1$**
MH_02	1.211$_2$	**36.825$_1$**	44.140$_2$	21.755$_2$	**0.562$_1$**	6.545$_3$	**23.770$_1$**	23.117$_3$	1.774$_3$	20.828$_2$	45.144$_3$	**11.226$_1$**
MH_03	0.270$_2$	**35.762$_1$**	42.873$_2$	22.762$_3$	**0.358$_1$**	7.740$_3$	**22.224$_1$**	22.490$_2$	0.415$_3$	20.000$_2$	49.013$_3$	**11.034$_1$**
MH_04	0.845$_2$	**36.461$_1$**	41.476$_2$	22.105$_3$	**0.622$_1$**	15.586$_3$	**21.324$_1$**	20.695$_2$	1.076$_3$	20.000$_2$	48.446$_3$	**11.103$_1$**
MH_05	1.223$_2$	**35.951$_1$**	41.934$_2$	21.580$_3$	**0.538$_1$**	8.795$_3$	**20.923$_1$**	20.432$_2$	1.600$_3$	21.509$_2$	45.744$_3$	**11.225$_1$**
V1_01	0.542$_2$	**34.721$_1$**	40.070$_2$	22.745$_3$	**0.483$_1$**	10.441$_3$	**23.028$_1$**	20.828$_2$	0.545$_3$	23.802$_2$	40.668$_3$	**11.286$_1$**
V1_02	**0.216$_1$**	**34.355$_1$**	37.925$_2$	21.972$_3$	1.088$_3$	8.579$_3$	**22.873$_1$**	20.225$_2$	0.438$_2$	16.296$_2$	44.583$_3$	**11.634$_1$**
V1_03	1.565$_3$	**36.315$_1$**	35.003$_2$	22.285$_3$	**0.065$_1$**	9.447$_3$	**25.892$_1$**	21.034$_2$	1.105$_2$	21.584$_2$	63.324$_3$	**11.677$_1$**
V2_01	0.125$_2$	**34.594$_1$**	40.291$_2$	22.839$_3$	**0.077$_1$**	11.107$_3$	**13.432$_1$**	21.206$_2$	0.617$_3$	22.306$_2$	49.672$_3$	**11.678$_1$**
V2_02	0.244$_2$	**34.631$_1$**	40.360$_2$	22.827$_3$	**0.458$_1$**	9.131$_3$	**17.273$_1$**	21.199$_2$	0.734$_3$	20.105$_2$	52.926$_3$	**11.814$_1$**
V2_03	**1.442$_1$**	**40.439$_1$**	38.791$_2$	21.434$_3$	2.733$_2$	12.010$_3$	**14.725$_1$**	20.561$_2$	\	21.425$_2$	56.748$_3$	**12.113$_1$**
Average	0.746$_2$	**36.070$_1$**	40.553$_3$	22.202$_3$	**0.664$_1$**	10.165$_3$	**20.964$_1$**	21.388$_2$	0.968$_3$	21.064$_2$	49.417$_3$	**11.438$_1$**

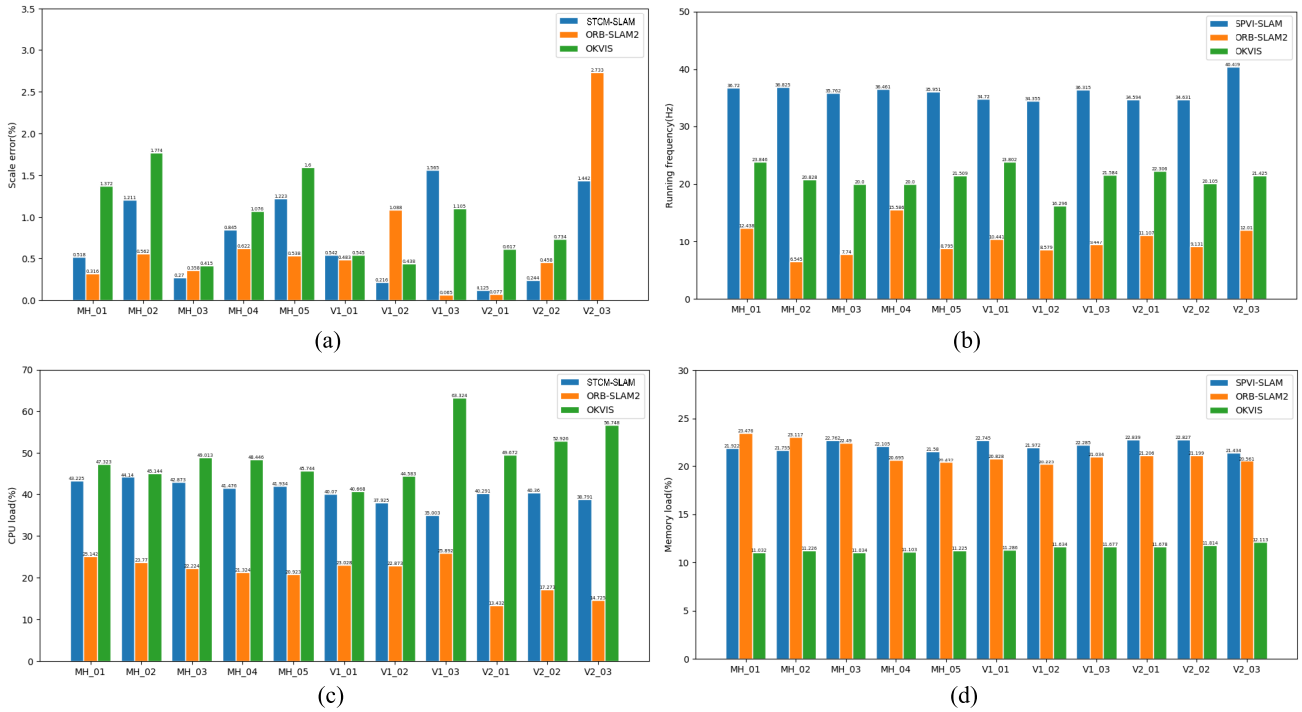


FIGURE 11. Performance comparisons between STCM-SLAM, ORB-SLAM2, and OKVIS. (a) The comparison of scale error. (b) The comparison of running frequency. (c) The comparison of CPU load. (d) The comparison of memory load.

TABLE 6. Trajectory analysis of the mobile robot trajectory.

Velocity(m/s)	STCM-SLAM			ORB-SLAM2			OKVIS		
	RMSE(m)	mean(m)	STD(m)	RMSE(m)	mean(m)	STD(m)	RMSE(m)	mean(m)	STD(m)
0.45	**0.064$_1$**	**0.055$_1$**	**0.066$_1$**	0.708$_3$	0.522$_3$	0.475$_3$	0.353$_2$	0.123$_2$	0.111$_2$
0.60	**0.054$_1$**	**0.048$_1$**	**0.057$_1$**	0.552$_3$	0.337$_3$	0.435$_3$	0.344$_2$	0.123$_2$	0.093$_2$
0.80	**0.052$_1$**	**0.038$_1$**	**0.053$_1$**	0.597$_3$	0.378$_3$	0.456$_3$	0.280$_2$	0.083$_2$	0.086$_2$
Average	**0.057$_1$**	**0.046$_1$**	**0.059$_1$**	0.619$_3$	0.413$_3$	0.456$_3$	0.326$_2$	0.110$_2$	0.097$_2$

ORB-SLAM and OKVIS. Although ORB-SLAM obtains the best absolute trajectory estimation and scale estimation, it cannot meet the real-time requirement of experiments. In contrast, STCM-SLAM and OKVIS are able to meet the real-time requirement but STCM-SLAM has obvious advantages over the OKVIS in terms of scale error, running frequency, and CPU load.

In this investigation, we compare the inter-frame processing times of STCM-SLAM, ORB-SLAM2, and OKVIS in order to better assess the real-time performance of SLAM systems. These results are presented in Fig. 12. The time variation for the inter-frame calculations across ORB-SLAM, OKVIS, and STCM-SLAM are indicated by blue, yellow, and green lines, respectively. As shown across Fig. 12(a) to

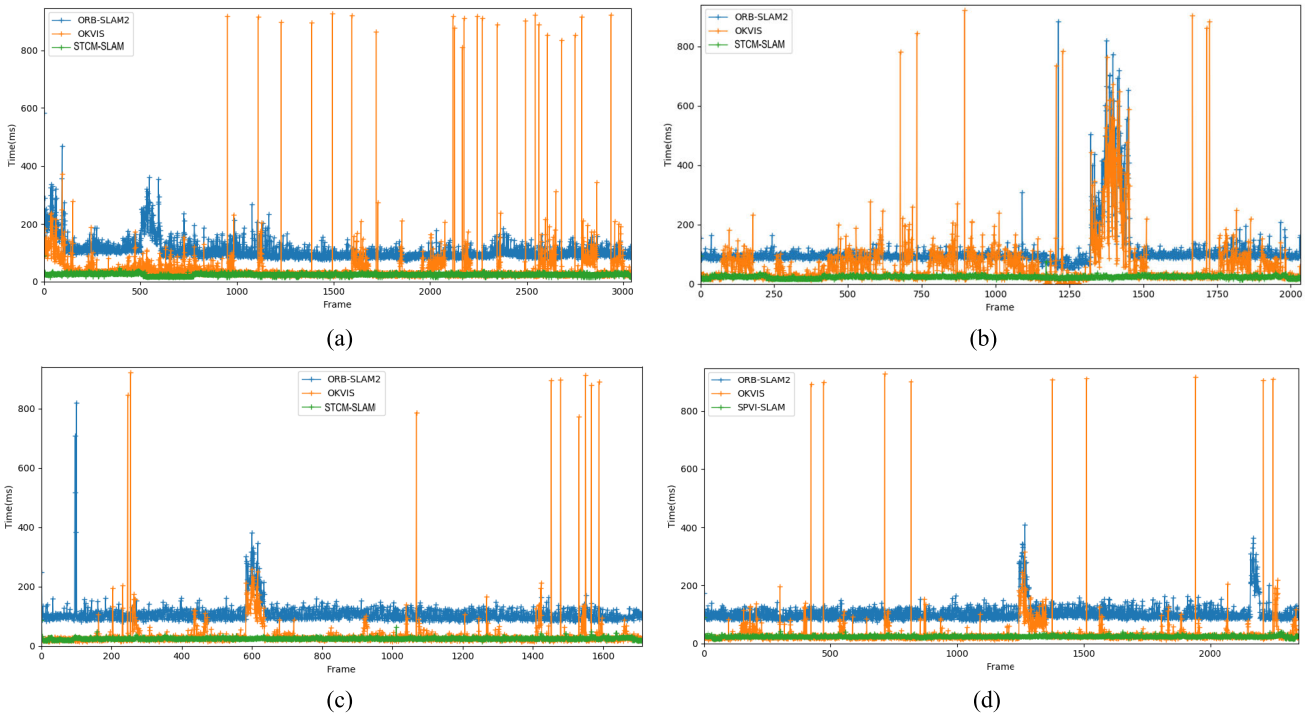


FIGURE 12. Inter-frame processing time comparisons between STCM-SLAM, ORB-SLAM2, and OKVIS. (a) MH_02 dataset. (b) MH_04 dataset. (c) V1_02 dataset. (d) V2_02 dataset.

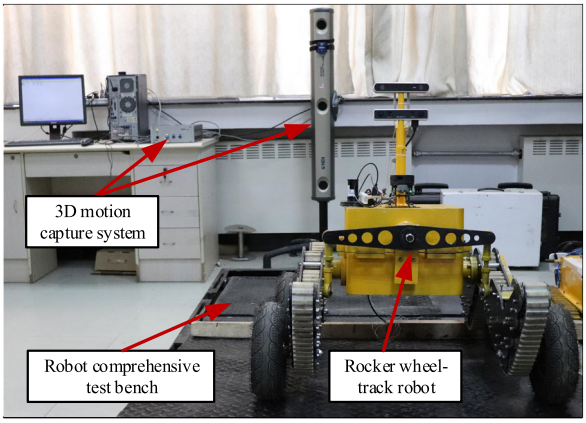


FIGURE 13. The layout of the mobile robot motion capture system.

Fig. 12(d), the inter-frame processing time of STCM-SLAM is minimal and relatively stable. Overall, the inter-frame processing time for OKVIS is superior to ORB-SLAM2, but OKVIS also produces some extremely long inter-frame processing times, due to the effect of global optimization, which affects the system's robustness.

C. INDOOR EXPERIMENT

When the mobile robot is working in an environment similar to a roadway or promenade, the robot's movement is mainly dominated by straight actions. Consequently, this paper simulates a laboratory environment as the mobile robot's operation environment, and uses the 3D motion capture system to evaluate the localization accuracy of the STCM-SLAM system. The arrangement of the equipment used in the mobile robot motion capture system is shown in Fig. 13.

In this experiment, we use a rocker wheel-track robot [3] equipped with a MYNT stereo camera. This stereo camera consists of two global shutters and a 6-axis IMU. The robot's comprehensive test bench serves as the platform for the mobile robot's movements. The bench surface is both convex and concave, thus simulating a road surface. In addition, the inclination angle is set to 10°, and baffles are placed at the start and end positions of the bench, serving as obstacles to increase the running collision effect.

During the experiment, the stereo camera's frequency is set to 20 Hz and the IMU is set to 200 Hz, which allows for time synchronization. The stereo camera and IMU have been previously rectified. The comparison between the trajectories calculated by STCM-SLAM, ORB-SLAM2 and OKVIS, and the ground-truth at different velocities is shown in Fig. 14.

As can be seen in Fig. 14, the trajectories of STCM-SLAM most closely match ground-truth trajectories. Furthermore, the localization accuracy of STCM-SLAM is better than either ORB-SLAM2 or OKVIS. As ORB-SLAM2 uses a pure visual method for localization and does not use IMU data, it is not able to effectively estimate poses during the pure translation of the mobile robot. The scale accuracy of ORB-SLAM2 has a large deviation, and the trajectories of the three methods are more deviated than ground truth. OKVIS performs better than ORB-SLAM2; however, the localization accuracy OKVIS achieved is lower than STCM-SLAM's.

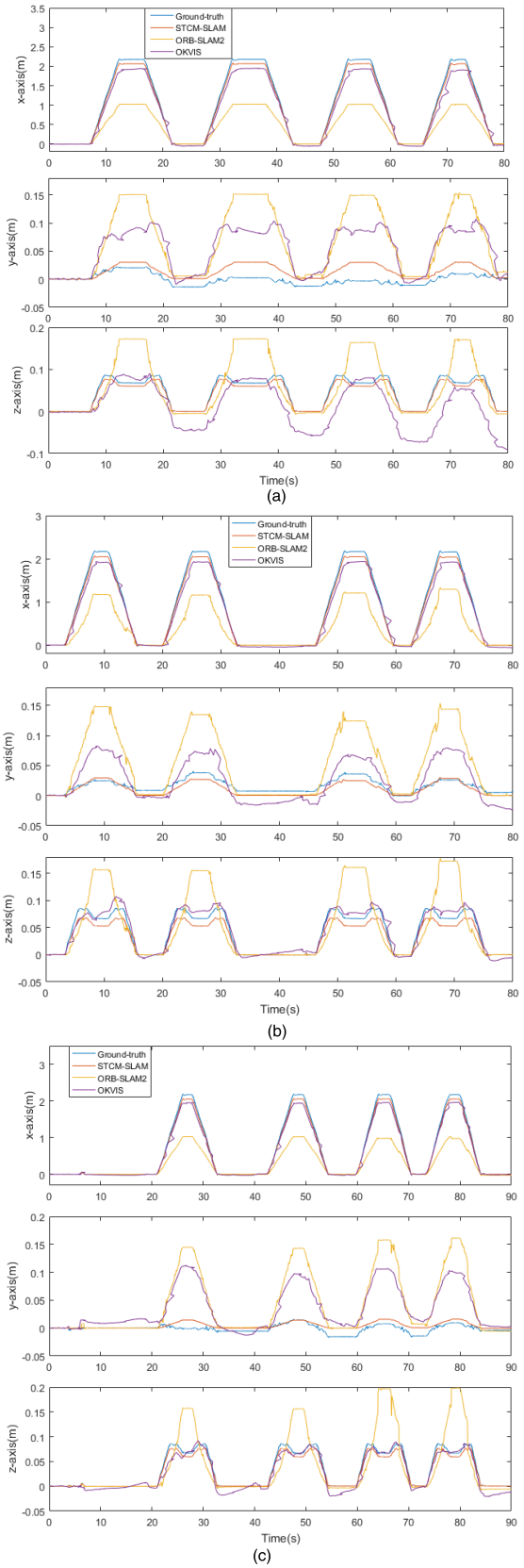


FIGURE 14. Comparison of the trajectories for ground-truth, STCM-SLAM, ORB-SLAM2, and OKVIS. (a) The trajectories at 0.45 m/s. (b) The trajectories at 0.60 m/s. (c) The trajectories at 0.80 m/s.

The trajectory of the y-axis and z-axis of the mobile robot is far less than the x-axis. To this end, we mainly evaluate the trajectory of the x-axis. Table 6 presents the mean error, RMSE, and STD of the x-axis trajectories for these three methods, with the mobile robot running at a range of different velocities. In terms of RMSE, the accuracy of STCM-SLAM is 980.702% and 475.439% over ORB-SLAM2 and OKVIS, respectively. In terms of mean error, the accuracy of STCM-SLAM is an increase of 795.652% and 136.957% over ORB-SLAM2 and OKVIS. In terms of mean error, the accuracy of STCM-SLAM represents an increase of 691.525% and 62.712% over ORB-SLAM2 and OKVIS. Based on RMSE, mean error, and STD, the accuracy of STCM-SLAM is superior to that of either ORB-SLAM2 or OKVIS.

VII. CONCLUSION AND FUTURE WORK

This paper investigates the effect of tightly coupling the stereo camera and IMU in order to better estimate the position of mobile robots in unknown environments without occlusions. Forward backward optical flow is used to track features, and STCM is proposed to manage features. In terms of relative pose error, the STCM-SLAM results are an order of magnitude greater than ORB-SLAM2 and two orders of magnitude greater than OKVIS. Our experiments indicate that STCM-SLAM has obvious advantages over OKVIS in terms of scale error, running frequency, and CPU load. In the indoor experiments, STCM-SLAM is able to accurately estimate the trajectory of the mobile robot, and it outperforms OKVIS and ORB-SLAM2 in terms of RMSE, mean error, and STD.

In future work, it would be useful to fuse light sources into the SLAM system, allowing us to solve problems related to the localization and mapping of robots in dark and narrow environments. In addition, using upward and forward cameras, or panoramic cameras, would further improve localization accuracy.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[3] C. Chen and H. Zhu, "Visual-inertial SLAM method based on optical flow in a GPS-denied environment," *Ind. Robot, Int. J.*, vol. 45, no. 3, pp. 401–406, May 2018.

[4] S. Yang, S. A. Scherer, X. Yi, and A. Zell, "Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles," *Robot. Auton. Syst.*, vol. 93, pp. 116–134, Jul. 2017.

[5] S. Bu, Y. Zhao, G. Wan, and Z. Liu, "Map2DFusion: Real-time incremental UAV image mosaicing based on monocular SLAM," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4564–4571.

[6] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Seattle, WA, USA, May 2015, pp. 2174–2181.

[7] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Merida, Mexico, Sep. 2016, pp. 1–10.

[8] J. Gui, D. Gu, S. Wang, and H. Hu, "A review of visual inertial odometry from filtering and optimisation perspectives," *Adv. Robot.*, vol. 29, no. 20, pp. 1289–1301, Sep. 2015.
[9] C. Chen, H. Zhu, M. Li, and S. You, "A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives," *Robotics*, vol. 7, no. 3, p. 45, Aug. 2018.
[10] J. Bai, J. Gao, Y. Lin, Z. Liu, D. Liu, and S. Lian, "A novel feedback mechanism-based stereo visual-inertial SLAM," *IEEE Access*, vol. 7, pp. 147721–147731, Oct. 2019.
[11] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Cambridge, U.K., Sep. 2007, pp. 1–10.
[13] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.
[14] R. Mur-Artal and J. D. Tardós, "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.
[15] X. Lin, F. Wang, L. Guo, and W. Zhang, "An automatic key-frame selection method for monocular visual odometry of ground vehicle," *IEEE Access*, vol. 7, pp. 70742–70754, 2019.
[16] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 834–849.
[17] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, May/Jun. 2014, pp. 15–22.
[18] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 2320–2327.
[19] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[20] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, Nov. 2013, pp. 2100–2106.
[21] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 343–352.
[22] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.
[23] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, 2018.
[24] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, May 2011, pp. 3400–3407.
[25] K. Shabalina, A. Sagitov, M. Svinin, and E. Magid, "Comparing fiducial markers performance for a task of a humanoid robot self-calibration of manipulators: A pilot experimental study," in *Proc. Int. Conf. Interact. Collaborative Robot.*, Greenville, SC, USA, Aug. 2018, pp. 249–258.
[26] R. Muñoz-Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, "Mapping and localization from planar markers," *Pattern Recognit.*, vol. 73, pp. 158–171, Jan. 2018.
[27] J. DeGol, T. Bretl, and D. Hoiem, "Improved structure from motion using fiducial marker matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Greenville, SC, USA, Sep. 2018, pp. 273–288.
[28] H. Sarmadi, R. Muñoz-Salinas, M. Berbís, and R. J. I. A. Medina-Carnicer, "Simultaneous multi-view camera pose estimation and object tracking with squared planar markers," *IEEE Access*, vol. 7, pp. 22927–22940, 2019.
[29] B. Pfrommer and K. Daniilidis, "TagSLAM: Robust SLAM with fiducial markers," 2019, *arXiv:1910.00679*. [Online]. Available: https://arxiv.xilesou.top/abs/1910.00679
[30] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, May 2011, pp. 3281–3288.
[31] R. Muñoz-Salinas, M. J. Marín-Jimenez, and R. Medina-Carnicer, "SPM-SLAM: Simultaneous localization and mapping with squared planar markers," *Pattern Recognit.*, vol. 86, pp. 156–171, Feb. 2019.
[32] R. Muñoz-Salinas and R. Medina-Carnicer, "UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," 2019, *arXiv:1902.03729*. [Online]. Available: https://arxiv.xilesou.top/abs/1902.03729
[33] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Rome, Italy, Apr. 2007, pp. 3565–3572.
[34] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, Jun. 2013.
[35] M. Li and A. I. Mourikis, "Improving the accuracy of EKF-based visual-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Saint Paul, MN, USA, May 2012, pp. 828–835.
[36] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, "Semi-direct EKF-based monocular visual-inertial odometry," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 6073–6078.
[37] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 298–304.
[38] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Graz, Austria, May 2006, pp. 430–443.
[39] 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, Dec. 2015.
[40] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *J. Field Robot.*, vol. 35, pp. 578–595, Jun. 2018.
[41] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
[42] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, Oct. 2004.
[43] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 1974–1982.
[44] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.
[45] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 2510–2517.
[46] Y. Liu, D. Yang, J. Li, Y. Gu, J. Pi, and X. Zhang, "Stereo visual-inertial SLAM with points and lines," *IEEE Access*, vol. 6, pp. 69381–69392, 2018.
[47] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, "Learning-based image enhancement for visual odometry in challenging HDR environments," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 805–811.
[48] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 7286–7291.
[49] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Salt Lake, UT, USA, Jun. 2018, pp. 224–236.
[50] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph slam," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1371–1385, Dec. 2014.
[51] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[52] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[53] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2015.
[54] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1680–1687.
[55] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 7244–7251.
[56] C. Wang, T. Ji, T.-M. Nguyen, and L. Xie, "Correlation flow: Robust optical flow using kernel cross-correlators," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 836–841.
[57] 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, Jan. 2016.

CHANG CHEN received the master's degree with the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, China, in 2019. He is currently a Researcher with SenseTime. His research interests include SLAM/VIO, robotics, deep learning, and computer vision.

HUA ZHU received the Ph.D. degree from the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, China. He is currently a Professor and the Supervisor of Ph.D. students with the China University of Mining and Technology. His research interests include robotics, computer vision, SLAM, the tribology theory and application, fractal and chaos theory, noise, and vibration control.

LEI WANG received the bachelor's degree in mechanical engineering from Jiangsu Normal University, Xuzhou, China, in 2017. He is currently pursuing the master's degree with the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou. His research interests include computer vision and intelligent robot.

YU LIU received the bachelor's degree in mechanical design, manufacturing and automation from Heilongjiang University, Harbin, China, in 2018. He is currently pursuing the master's degree with the School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou. His research interests include path planning and intelligent robot.