IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 9, 2021, accepted March 28, 2021, date of publication April 8, 2021, date of current version June 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071921

Semi-Direct Monocular SLAM With Three Levels of Parallel Optimizations

SHOUYI LU, YONGSHUAI ZHI, SUMIN ZHANG, RUI HE, AND ZHIPENG BAO
State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China

Corresponding author: Sumin Zhang (zhangsumin@jlu.edu.cn)

This work was supported in part by the Research on Construction and Simulation Technology of Hardware in Loop Testing Scenario for Self-driving Electric Vehicle in China under Grant 2018YFB0105103, in part by the National Science Foundation of China under Grant U1564211, and in part by the Graduate Innovation Fund of Jilin University.

ABSTRACT In practical applications, how to use the complementary strengths of the direct and the feature-based methods for effective fusion may be the main challenge of simultaneous localization and mapping (SLAM). To solve this challenge, we propose the DO-SLAM, a novel fast and accurate semi-direct visual SLAM framework, which can maintain the direct method's fast performance and the high precision and loop closure capability of the feature-based method. The direct method is used as the first half of the DO-SLAM to track the camera pose rapidly and robustly. The feature-based method is used as the second half of the DO-SLAM to refine the keyframe poses, perform loop closures, and build a globally consistent, long-term, sparse feature map that can be reused. The proposed pipeline fuses direct odometry and feature-based SLAM to perform three levels of parallel optimizations: (1) In the direct method module, the keyframe poses are estimated by minimizing the photometric error, (2) In the feature-based module, using the poses calculated by the inter-frame matching to correct and fuse the poses calculated by the direct method module as the initial poses, and the initial poses are optimized by the motion-only bundle adjustment, and (3) A pose graph optimization is used to achieve global map consistency in the presence of loop closures. Experimental evaluation on two benchmark datasets demonstrates that the proposed approach achieves higher accuracy and robustness on motion estimation compared to the other state-of-the-art methods.

INDEX TERMS Simultaneous localization and mapping (SLAM), semi-direct SLAM, three levels of parallel optimizations.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) plays an essential role in self-driving cars, virtual and augmented reality, unmanned aerial vehicles (UAV), artificial intelligence [1], [2]. This technology can provide reliable state estimation for UAV and self-driving cars in GPS-denied environments by relying on its sensors. Various sensors can be utilized in SLAM, such as stereo camera, lidar, inertial measurement units (IMU), and monocular camera. In different sensor modes, visual sensors, especially monocular cameras, provide a cheap solution with great potential [3].

Traditional visual SLAM can be divided into two classes: feature-based and direct methods. Feature-based methods extract salient image features in each image, match them in successive frames using invariant feature descriptors, robustly recover camera poses and structure using epipolar geometry, and refine poses and structure by minimizing projection errors [4]. Despite the good performances in the past several years, these feature-based approaches are still very sensitive to noise and outliers, time-consuming during the process of feature extraction and matching. In addition, the feature-based methods positively ignore the global cues except local features, making them unable to solve some challenges such as missing features [5].

The direct methods have been proposed to tackle the above drawbacks, which directly recover the camera poses and structure through photometric error without features extraction. Therefore, in the low-texture environments and repeated texture environments, the direct method's performance is better than the feature-based method. In addition, without feature extraction and matching, the direct method's calculation speed is faster than the feature-based method. However, the direct method is based on the assumption of the ideal

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

situation that the gray value is constant, so it is susceptible to camera internal parameters and light changes. Moreover, the photometric error function is highly non-convex. It is difficult for the direct method to converge when large baseline motion and image blur occurs.

In order to effectively combine the advantages of the direct method and the feature-based method to achieve a more accurate estimation of the camera poses, a novel semi-direct approach is proposed in this study to maintain the fast performance of a direct method and the high precision and loop closure capability of a feature-based method. This approach uses the DSO [6] as the first half to track the camera poses rapidly and robustly and uses the nonlinear optimization based on sliding window to solve the keyframe poses and the coordinates of map-points. The improved ORB-SLAM2 [7] is used as the second half of the approach to refine the keyframe poses, perform loop closures, and build a globally consistent, long-term, sparse feature map can be reused. Accordingly, this system is called DO-SLAM. Furthermore, the results of the proposed monocular SLAM systems are compared with that of state-of-the-art approaches as demonstrated on open datasets.

The main contributions of this work are summarized as follows:

- We present DO-SLAM, a novel fast and accurate semi-direct visual SLAM framework, that combines the exactness of the feature-based method and the quickness of the direct method.
- We propose the three levels of parallel optimization structure to optimize the keyframes poses.
- We propose a map-points fusion strategy based on direct method and feature-based method to optimize map-points' coordinates.
- We validate the proposed algorithm on two benchmark datasets, and results show that our system outperforms state-of-the-art methods, e.g., DSO [6], ORB-SLAM2 [7], and OpenVSLAM[8].

The rest of this paper is summarized as follows: Section II provides an overview of the current direct method, feature-based method, and semi-direct method. Section III shows the overview of DO-SLAM. Section IV and V demonstrate the direct module and feature-based module, respectively. The results of the open datasets are presented in Section VI. Finally, the conclusion is drawn in Section VII.

II. RELATED WORKS

There is a significant number of research works related to visual-based localization over the last decades. According to the implementation, they can be classified into the following categories.

(1) **Feature-based:** Feature-based methods leverage salient image features (like the point or the line features) to recover and refine camera motion by minimizing reprojection errors of the features correspondences [9]–[11]. The first monocular approach MonoSLAM was proposed in 2003 by Davison *et al.* [12], Davison [13]. MonoSLAM used EKF as the back end to track the sparse feature points acquired by the front end and used the camera poses and the landmark points as the state variables to update its mean and covariance. PTAM [11] was proposed in 2007, the first real-time feature indirect SLAM method, from the University of Oxford, which split the poses and map estimation into different threads and proposed to use BA(Bundle Adjustment). After that, most feature-based methods were improved versions of PTAM, one of which is ORB-SLAM2 [7]. ORB-SLAM2 is the most successful feature-based SLAM, which uses ORB features in tracking, mapping, re-location, and loop closure detection [7]. OpenVSLAM was proposed in 2019, which created maps that can be stored and loaded, then OpenVSLAM can localize new images using prebuilt maps [8]. Newly, ORB-SLAM3 was released on arXiv in August 2020 [14]. It focused on the integration of ORB-SLAM2 and IMU information and multi-map system.

(2) **Direct:** In contrast to feature-based methods, direct methods aim at using the whole image to estimate the structure and motion. DTAM is a monocular slam method based on the direct method proposed in 2011. Compared with the traditional slam method that extracts sparse features, this method extracts the inverse depth of each pixel and constructs a dense map through optimization. The camera poses are calculated by using the depth map through direct image matching [15]. LSD-SLAM [16] is the first direct visual SLAM approach for monocular cameras that is capable of mapping large scale environments in real-time. It tracks the camera motion, produces a semi-dense map, and performs pose graph optimization to obtain a consistent global map. On the basis of LSD-SLAM, direct sparse odometry (DSO) [6] samples pixels evenly throughout the images and integrates a fully photometric calibration, which accounts for exposure time, lens vignetting, and nonlinear response functions. Based on DSO, LDSO adds loop closure detection, ensuring tracking accuracy during long navigation [17]. Stereo DSO is an improved version of DSO, in which depth value is estimated by multiple view geometry [18]. Lee authors in [19] presents a new implementation method for efficient simultaneous localization and mapping using a forward-viewing monocular vision sensor. The method is developed to be applicable in real time on a low-cost embedded system for indoor service robots. Finally, with the development of deep learning, some SLAM applications emerge to imitate the previously proposed approaches [20], [21].

(3) **Semi-Direct:** Semi-direct methods estimate camera poses using both direct and feature-based methods. In [22], a semi-direct monocular VO (SVO) was implemented on the onboard computer of a multirotor, showing precise and fast state estimation results by combining the advantages of feature-based and direct methods,

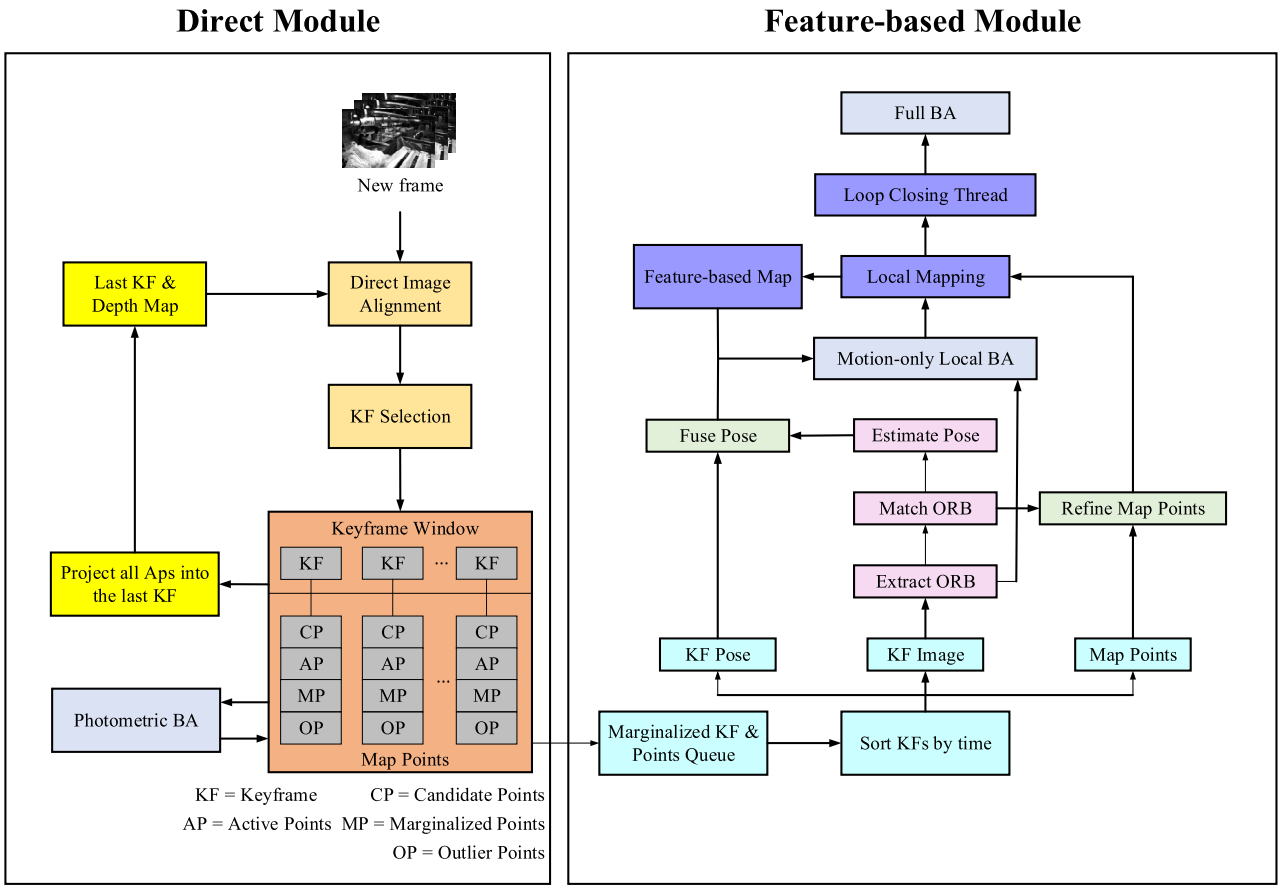

FIGURE 1. Overall system architecture of DO-SLAM.

and it was extended to multi-camera systems in [23]. However, SVO does not have back-end optimization and loop-closing capability. Gomez-Ojeda proposed an improved SVO by combining points and line segments (PL-SVO) [24] to solve the problem that SVO is still a strong dependence on the initial value of the poses. Nicola Krombach, based on the monocular version of LSD-SLAM, proposed a semi-direct approach for stereo odometry [25]. Lee and Civera [26] proposed a loose-coupled method by combining ORB-SLAM and DSO to improve the positioning accuracy. However, its front-end and back-end are almost independent, which cannot share estimation information to improve the poses precision further. In [27], a semi-direct approach was proposed for stereo odometry. This method uses the feature-based method to obtain a motion estimation, and then perform direct semi-dense to refine the camera pose. SVL [28] can be considered a combination of ORB-SLAM and SVO. The method for ORB-SLAM is adopted in keyframes, and SVO is adopted in non-keyframes.

III. DO-SLAM SYSTEM

A. THREE OPTIMIZED STRUCTURES OF DO-SLAM

Figure 1 demonstrates the framework of the DO-SLAM system. DO-SLAM includes two parts: direct module and feature-based module. As shown in Figure 2, the system applies DSO to quickly track each frame and provide an initial keyframe poses and the coordinates of map-points, and a modified version of ORB-SLAM2 to refine keyframe poses and coordinates of map-points, build a globally consistent map and detect loop with marginalized keyframes from the direct method. Therefore, extracting features and matching descriptors are no longer required in a non-keyframe. The system selects ORB as features, which are oriented multiscale FAST corners with an associated 256-bit descriptor. These features are very fast to be extracted and matched with good invariance in viewpoint.

The system includes three different optimized structures. In the direct module, this system uses DSO to achieve real-time camera tracking, uses the nonlinear optimization method based on the sliding window, and projects the map-points tracked on each keyframe in the sliding window onto the current frame to construct a photometric error based on the camera poses. The initial estimation of the keyframe poses is obtained by minimizing the photometric error.

In the feature-based module, this system uses the modified version of ORB-SLAM2 to refine keyframe poses and coordinates of map-points. When a keyframe is marginalized from the direct module, its image and poses information are sent to the feature-based module, along with the map points within

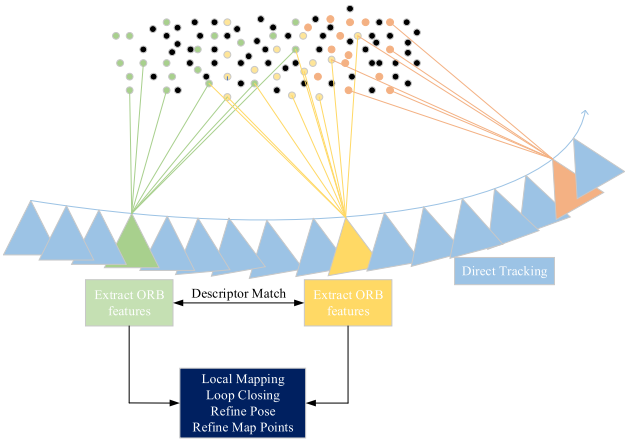

FIGURE 2. DO-SLAM schematic.

its FOV. The feature-based module extracts ORB descriptors from the marginalized image and then obtains the keyframe poses and coordinates of map-points by matching feature points between frames. These poses are used to verify the poses obtained by the direct module, and then the two poses are fused by the weighted method to obtain the best initial poses. Finally, refining the initial poses concerning the local feature map using motion-only BA.

In addition, when the loop is detected, the pose graph optimization is performed over Sim (3) constraints [29]. All keyframes and map-points in the map are optimized using the full bundle adjustment to achieve global consistency.

B. THE ALGORITHM FLOW OF DO-SLAM

The images from the monocular camera are input to the direct module. New images are tracked using direct image alignment [30] with respect to the last keyframe and its depth map created by projecting active points in the sliding window. If the image should be created as a keyframe, the frame is inserted into the sliding window. At the same time, an old keyframe is removed with the marginalization method. Finally, the keyframes poses and map-points in the sliding window are optimized.

The marginalized keyframe is sent to the feature-based module. Since the direct module does not necessarily marginalize the oldest keyframe in the sliding window, the system stores the marginalized keyframe information in the queue and wait for the oldest keyframe to be marginalized. When the oldest keyframe is received, the ORB features are extracted from this keyframe, and then the keyframe poses are optimized using the optimization method described in Section A to obtain the accurate poses. The map-points corresponding to the ORB features are solved by the triangulation method, and the map-points are fused with those obtained by the direct method. Finally, the keyframe and map-points are inserted into the local map. Simultaneously, the loop closing thread detects loops using the bag-of-words place recognizer built on DBoW2 with ORB. The accumulated error is corrected via pose graph optimization, which distributes loop closing errors along the graph.

IV. DIRECT MODULE

The system uses the original implementation of DSO [6] as the direct module. DSO is a keyframe-based sliding window approach, where 5-7 keyframes are maintained, and their parameters are jointly optimized in the current window. In this section, we mainly describe the windowed optimization and marginalization strategy of DSO. For other parts, readers can refer to the original work [6].

A. WINDOWED PHOTOMETRIC BUNDLE ADJUSTMENT

When a point p in a reference frame I_i is observed in current frame I_j. The photometric error E_{pj} is defined as the weighted SSD over the 8-point neighborhood pixels N_p as proposed in [6].

$$E_{pj} = \sum_{p \in N_p} \omega_p \left\| (I_j[p'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[p] - b_i) \right\|_\gamma \tag{1}$$

where $\|\cdot\|_\gamma$ is the Huber norm; t_i, t_j is the exposure times of the images I_i, I_j; a_i, a_j, b_i, b_j is the brightness transfer function parameters. ω_p is a weighting that down-weights high image gradients; p' stands for the projected point position of p with inverse depth d_p. ω_p and p' are calculated as follows:

$$\omega_p = \frac{c^2}{c^2 + \|\nabla I_i(p)\|_2^2} \tag{2}$$

$$p' = \prod\nolimits_K (T_{ji} \prod\nolimits_K^{-1} (p, d_p)) \tag{3}$$

where c is camera intrinsic parameters; T_{ji} is the pose transformation from frame i to frame j; $\prod_K : \mathbb{R}^3 \to \Omega$ and $\prod_K^{-1} : \mathbb{R}^3 \to \Omega$ is corresponding camera projection and back-projection functions.

The full photometric error over all frames and points is given by

$$E_{\text{photo}} = \sum\nolimits_{m \in \mathcal{F}} \sum\nolimits_{p \in P_m} \sum\nolimits_{j \in \text{obs}(p)} E_{pj} \tag{4}$$

where $\mathcal{F}$ is the set of all frames in the window; P_i is the set of all points in the frame I_m; obs(p) is the set of frames that can observe the point p.

If exposure times are known, we further add a prior pulling the affine brightness transfer function to zero. The total energy function is given by

$$E_{\text{end}} = E_{\text{photo}} + \sum\nolimits_{i \in \mathcal{F}(\lambda_a a_i^2 + \lambda_b b_i^2)} \tag{5}$$

where λ_a and λ_b is the specified constant values. If the exposure times are unknown, we set $\lambda_a = \lambda_b = 0$ and $t_i = t_j = 1$ in (1).

Finally, the Gauss-Newton optimization algorithm is used to iteratively solve the total energy function in the sliding window. The update equation is given by:

$$\delta\xi = -(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \mathbf{r} \tag{6}$$

$$\xi^{\text{new}} \leftarrow \delta\xi \odot \xi \tag{7}$$

where $\mathbf{r} \in \mathbb{R}$ is the stacked residual vector; $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the diagonal matrix containing the weights; $\mathbf{J} \in \mathbb{R}^{n\times d}$ is the Jacobian of $\mathbf{r}$; $\xi \in \mathrm{SE}(3)^n \times \mathbb{R}^m$ to denote all optimized variables, including camera poses, affine brightness parameters, inverse depth values, and camera intrinsics.

B. MARGINALIZATION STRATEGY

When the set of active variables in the sliding window is too large, the least useful keyframes and points are marginalized by using the Schur complement. In the point marginalization strategy, if a point has not been continuously observed in the latest two keyframes or its host keyframe is marginalized, the point will be marginalized. In the keyframe marginalization strategy, the latest two keyframes (I_1 and I_2) are always kept. For other keyframes in the sliding window, if less than 5 percent of their points are visible in I_1, the keyframe is marginalized. When the active keyframes in the sliding window are greater than the maximum number of keyframes that the window can contain, the largest distance score's keyframes are marginalized. The distance score is computed as:

$$s(I_i) = \sqrt{d(i,1)} \sum\nolimits_{j\in[3,n]\setminus\{i\}} (d(i,j)+\varepsilon)^{-1} \tag{8}$$

where $d(i,j)$ is the euclidean distance between keyframes I_i and I_j, and ε is a small constant. Through Equation (8), the keyframes' information in the sliding window is approximate to the newly inserted keyframe.

V. FEATURE-BASED MODULE

When the keyframe is marginalized from the direct module, the feature-based module receives the keyframe information, including its image, poses, and the map-points observed by the keyframe. This information is then used for feature-based poses and map-points refinement, mapping, and loop closing. In this section, we will describe the optimization algorithm for keyframe poses and map-points. For other parts, the system uses the same algorithm as ORB-SLAM2.

A. INTER-FRAME MATCHING POSES ESTIMATION

After matching the ORB features in the current frame and the previous frame, the matching result can be used for poses estimation. If the map-points corresponding to the ORB features in the previous frame is known, the correspondence between the ORB features in the current frame and the map-points can be obtained. The system then performs RANSAC iterations alternatively for the current frame and tries to find camera poses using the PnP algorithm [31].

B. INITIAL POSES FUSION STRATEGY

The keyframe poses ξ_D by the direct module are the poses when the photometric error reaches the minimum value. Since the image is usually a non-convex function, the poses are likely to fall into a local minimum. The inter-frame matching poses ξ_F obtained by the multi-view geometry have strong accuracy. Therefore, the poses calculated by the inter-frame matching can be used to verify the poses calculated by the direct module, and the fusion can be conducted according to the test results. The absolute trajectory error is used to calculate the difference between the two poses. The transformation matrix of poses ξ_D is T_D, and the transformation matrix of poses ξ_F is T_F. The calculation method of poses difference e is as follows:

$$e = (\left\| trans(T_D^{-1} T_F) \right\|^2)^{\frac{1}{2}} \tag{9}$$

where $trans(T_D^{-1} T_F)$ denotes the translation part in the absolute trajectory error.

If e is greater than the given threshold, it means that ξ_D falls into the local minimum, then ξ_F is taken as the initial pose ξ_B.

If e is less than the given threshold, it means that ξ_D is accurate, and the average weighted method is used to fuse the two poses as the initial pose ξ_B.

C. KEYFRAME POSES REFINEMENT

Once the initial pose ξ_B is obtained, the system refines it using motion-only geometric BA with respect to the local feature map. The local feature map's keyframes are composed of the keyframes I_1 sharing the map-points with the current frame and the keyframes I_2 connected to the keyframes I_1 in the covisibility graph. The map-points in the local feature map are composed of all map-points of the local feature map's keyframes. According to the geometric relationship of projection, the map-points matched with ORB features in the current frame are selected. According to the selected map-points' projection position in the current frame, a set of candidate ORB features is determined for each map-point. Then the best matching ORB feature of each map-point is determined in the candidate ORB features. Finally, the total energy function is composed of the variance-normalized reprojection errors of the local map points:

$$E_{\text{reproj}} = \sum_{i\in\mathcal{F}_{\text{local}}} \sum_{x\in P_i} \sum_{j\in\text{obs}(x)} \left\| \frac{p_{j,x} - \prod_c (\mathbf{T}_{jw} x_w)}{\sigma_x^2} \right\|_\gamma \tag{10}$$

$$\sigma_x^2 := (\lambda_{\text{pyr}})^{2L_{\text{pyr},x}} \tag{11}$$

where $\mathcal{F}_{\text{local}}$ denotes the set of all local keyframes, $p_{j,x} \in \mathbb{R}^2$ the match to the ORB feature x in frame I_j, and σ_x^2 the variance of the feature location in frame I_i. This variance depends on the constant scale factor of the image pyramid λ_{pyr} and the pyramid level $L_{\text{pyr},x}$ at which the ORB feature was detected.

D. MAP-POINTS REFINEMENT

When there are no map-points corresponding to the ORB feature in the feature-based module's global map, new map-points are created by triangulating the ORB feature from connected keyframes I_c in the covisibility graph. For each unmatched ORB feature in I_i, the system searches a match with other un-matched points in other keyframes. ORB feature pairs are triangulated, and to accept the new points, positive depth in both cameras, parallax, reprojection error,

TABLE 1. The mean error on the TUM mono VO dataset.

Dataset	MEAN			
	DO-SLAM	DSO	ORB-SLAM2	OpenVSLAM
01	0.242	0.249	0.345	0.318
03	0.469	1.579	1.569	1.637
05	0.725	0.790	×	×
07	0.178	0.371	0.542	×

and scale consistency are checked. If the map-points have also been generated in the direct module, the average weighted method will be used to fuse the two map-points as the final map-points.

VI. EXPERIMENTS

In this section, we will extensively evaluate our algorithm (DO-SLAM). The DO-SLAM method is compared with the state-of-the-art vision SLAM methods, such as ORB-SLAM2, DSO, and OpenVSLAM. We use two datasets for evaluation:

1) The TUM mono VO dataset [32], which provides 50 photometrically calibrated sequences, comprising 105 minutes of video recorded in dozens of different environments, indoors and outdoors. Since the dataset does not provide all the ground truth, we only use part of the dataset's ground truth to evaluate the positioning accuracy.
2) The EuRoC MAV dataset [33], which contains 11 stereo-inertial sequences comprising 19 minutes of video, recorded in three different indoor environments. For this dataset, we only use the left video for evaluation. For this dataset, no photometric calibration or exposure times are available. Hence we omit photometric image correction and set ($\lambda_a = \lambda_b = 0$). For this dataset, we crop the beginning and end of each sequence to disregard large occlusions due to the ground and aggressive motions meant for IMU initialization.

A. POSITIONING ACCURACY ANALYSIS ON THE TUM MONO VO DATASET

Since DSO does not have the loop-closing capability, we close the loop closing thread of ORB-SLAM2, DO-SLAM, and OpenVSLAM. We adopt the open-source tool EVO [34] to evaluate the performance of DO-SLAM. By comparing the estimated value with the actual value, we calculate the absolute pose error (APE) as an index of the evaluation algorithm [35]. Table 1, 2 show the mean and root mean square error (RMSE) of the translation on the TUM mono VO dataset.

As shown in Tables 1, 2, DO-SLAM is better than DSO, ORB-SLAM2, and OpenVSLAM in terms of mean and root mean square error. Because the dataset is recorded in the indoor environment, there are a lot of low-texture environments, so ORB-SLAM2 and OpenVSLAM based on the feature-based method are easy to fail (such as sequence 05), but DO-SLAM combines DSO and ORB-SLAM2, even in the low-texture environment can run stably, and positioning accuracy is better than DSO.

TABLE 2. The root mean square error (RMSE) on the TUM mono VO dataset.

Dataset	RMSE			
	DO-SLAM	DSO	ORB-SLAM2	OpenVSLAM
01	0.253	0.260	0.379	0.438
03	0.483	1.690	1.715	1.942
05	0.834	0.919	×	×
07	0.182	0.381	0.553	×

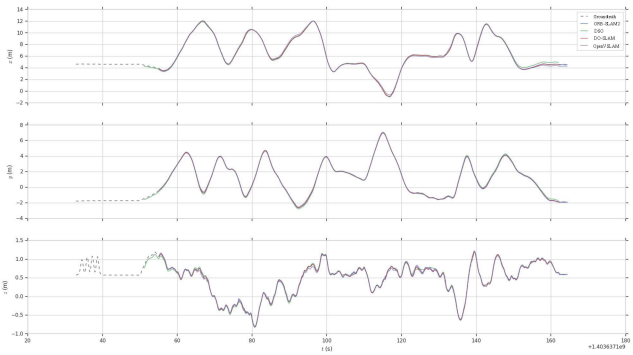

FIGURE 3. Comparison of position estimation.

B. POSITIONING ACCURACY ANALYSIS ON THE EUROC DATASET

In the experiments on the EuRoC dataset, we also adopt the open-source tool EVO [34] to evaluate DO-SLAM performance. For fairness, the following algorithms do not use the loop closure detection module. Take the test in the MH_03_medium dataset as an example to illustrate. Figure 3 shows the position comparison results of DO-SLAM, ORB-SLAM2, DSO, and OpenVSLAM in MH_03_medium. From the figure, we can see that the trajectory of DO-SLAM is closer to the real trajectory, followed by ORB-SLAM2, OpenVSLAM, and finally DSO.

Figure 4 shows more intuitively the trajectory heat map estimated by DO-SLAM, ORB-SLAM2, OpenVSLAM, and DSO in MH_03_medium. From the experiment, we can get that the overall RMSE of DO-SLAM in MH_03_medium is 0.027, the overall RMSE of ORB-SLAM2 in MH_03_medium is 0.035, the overall RMSE of OpenVSLAM in MH_03_medium is 0.039, and

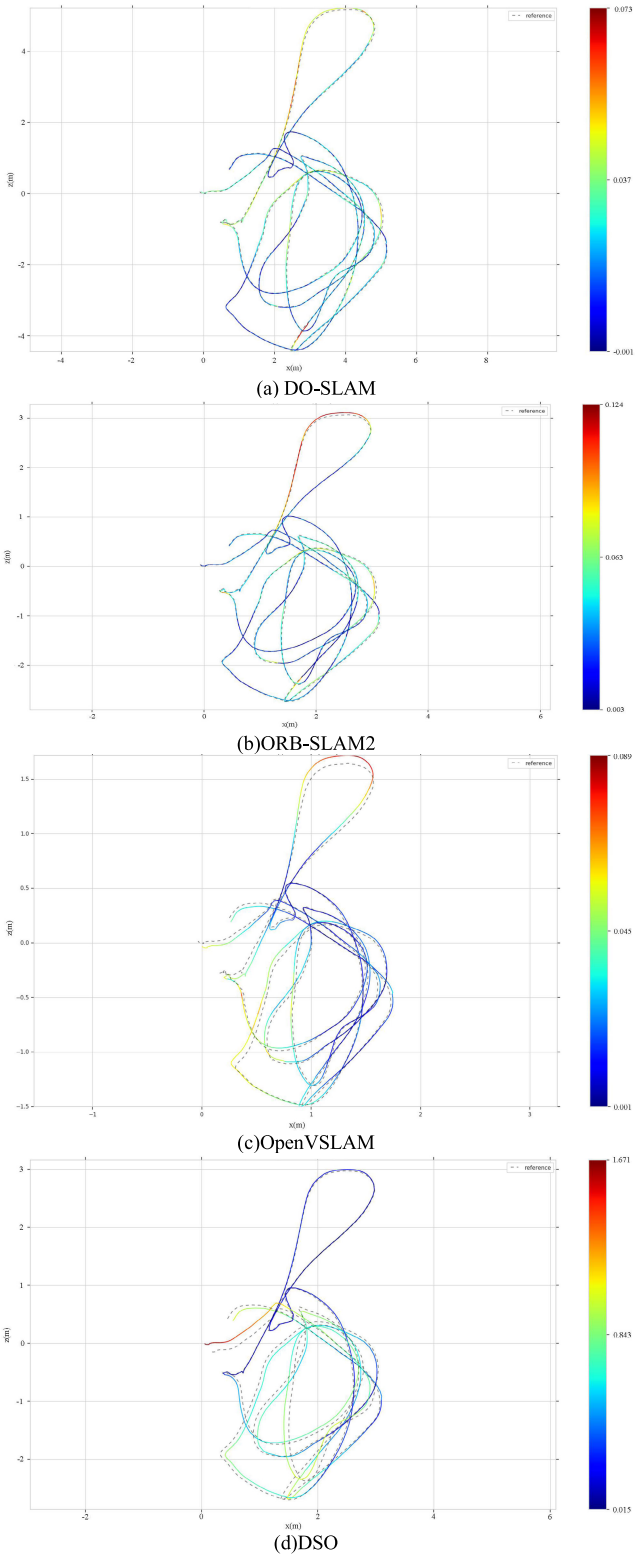

FIGURE 4. The trajectory heat map estimated by DO-SLAM, ORB-SLAM2, OpenVSLAM, and DSO in MH_03_medium.

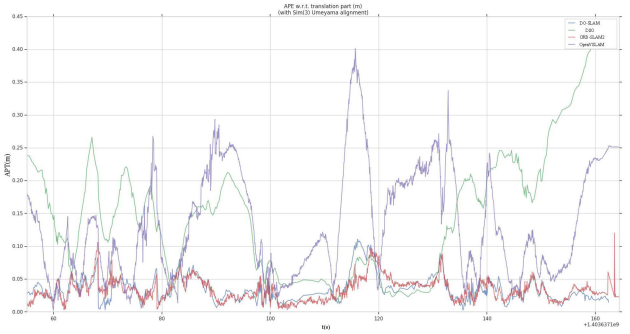

FIGURE 5. The change of translation absolute pose error with time in MH_03_medium.

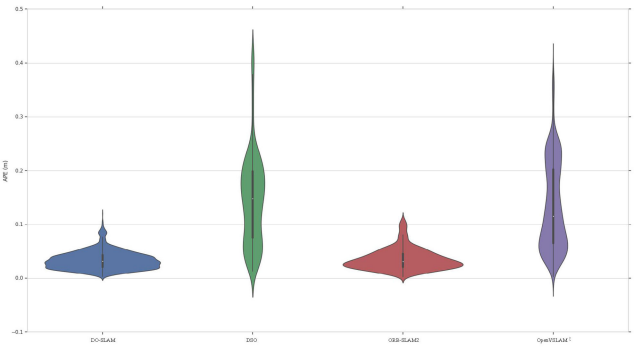

FIGURE 6. The overall distribution of absolute pose error in MH_03_medium.

TABLE 3. The mean error on the EuRoC dataset.

Dataset	MEAN			
	DO-SLAM	DSO	OpenVSLAM	ORB-SLAM2
MH_01_easy	0.026	0.053	0.034	0.031
MH_02_easy	0.025	0.042	0.039	0.027
MH_03_medium	0.025	0.600	0.033	0.031
MH_04_medium	0.051	0.064	0.054	0.060
MH_05_difficult	0.047	0.117	0.077	0.054
V1_01_easy	0.084	0.093	0.044	0.093
V1_02_ medium	0.090	0.226	0.175	0.158
V1_03_ difficult	1.136	1.274	×	1.202
V2_01_easy	0.063	0.078	0.032	0.077
V2_02_ medium	0.164	0.196	0.165	0.175
V2_03_ difficult	0.274	1.334	×	0.288

the overall RMSE of DSO in MH_03_medium is 0.639. Figures 5 and 6 show the change of translation absolute pose error with time and the overall distribution of absolute pose error in MH_03_medium. Through Figures 4, 5, and 6, we conclude that the accuracy and robustness of our algorithm have reached the level of state-of-the-art algorithm.

TABLE 4. The root mean square error (RMSE) on the EuRoC dataset.

Dataset	RMSE			
	DO-SLAM	DSO	OpenVSLAM	ORB-SLAM2
MH_01_easy	0.033	0.060	0.037	0.037
MH_02_easy	0.031	0.052	0.049	0.033
MH_03_medium	0.027	0.639	0.039	0.035
MH_04_medium	0.058	0.070	0.062	0.064
MH_05_difficult	0.054	0.129	0.084	0.063
V1_01_easy	0.088	0.099	0.046	0.096
V1_02_ medium	0.294	0.275	0.186	0.179
V1_03_ difficult	0.090	1.423	×	1.350
V2_01_easy	0.070	0.084	0.036	0.083
V2_02_ medium	0.178	0.215	0.196	0.189
V2_03_ difficult	0.294	1.456	×	0.328

TABLE 5. Average time (ms) spent tracking an image.

Dataset	DO-SLAM	DSO	OpenVSLAM	ORB-SLAM2
MH_01_easy	14.97	10.09	31.61	35.89
MH_02_easy	12.07	9.63	31.53	33.87
MH_03_medium	16.99	9.64	30.27	32.63
MH_04_medium	12.46	9.58	28.59	29.46
MH_05_difficult	11.79	9.59	29.53	29.67
V1_01_easy	11.74	9.83	31.49	36.08
V1_02_ medium	13.51	9.77	28.83	27.13
V1_03_ difficult	15.09	9.58	×	25.99
V2_01_easy	11.96	9.83	30.06	30.64
V2_02_ medium	14.49	9.49	29.57	29.33
V2_03_ difficult	13.70	9.52	×	26.52

Tables 3, 4 show our test results on other sequences on the EuRoC dataset. It can be seen that the DO-SLAM, which effectively integrates ORB-SLAM2 and DSO, significantly improves the positioning accuracy and positioning robustness compared to the other three algorithms.

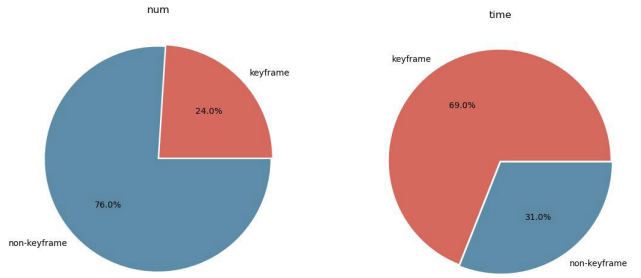

FIGURE 7. The left picture shows a comparison of the number of keyframes and non-keyframes. The right picture shows the comparison of time consumption between tracking keyframes and non-keyframes.

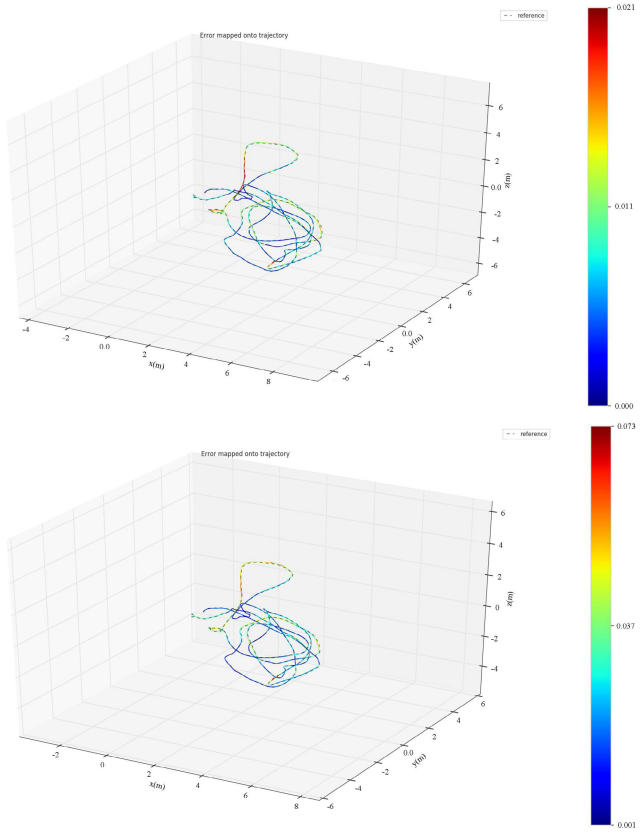

FIGURE 8. The trajectory heat map estimated by DO-SLAM-LOOP and DO-SLAM.

In addition, we evaluate the real-time performance of DO-SLAM. We compared the average time required to track an image (Table 5).

As shown in Table 5, the image tracking of ORB-SLAM2 and OpenVSLAM uses the feature-based method to extract and match each frame's ORB features, which takes a long time. However, DSO uses the direct method to track features, saving the calculation of feature descriptors, so the time consumption is less than ORB-SLAM2 and OpenVSLAM. In DO-SLAM, non-keyframes are used for fast-tracking and localization by the direct method, and keyframes are also tracked by feature-based methods and used for three levels of parallel optimizations and loop closure detection. This algorithm saves a lot of time and minimizes the average time

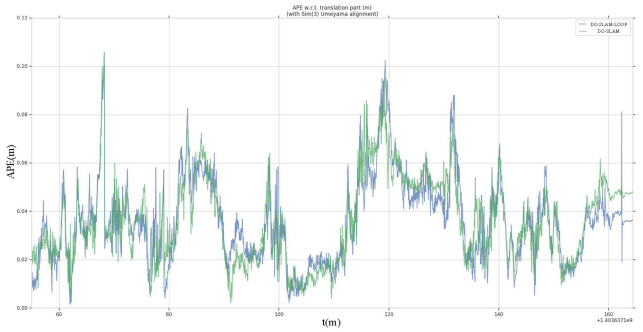

FIGURE 9. The trajectory heat map estimated by DO-SLAM-LOOP and DO-SLAM.

of DO-SLAM tracking images. Although DO-SLAM is not as fast as DSO, its real-time performance is much better than feature-based methods.

Compared with the feature-based method, we use the direct method to track non-keyframes and accelerate the algorithm without reducing the accuracy and robustness. As shown in Figure 7, in MH_03_medium, 24% of the frames are determined to be keyframes, while 76% of the frames are determined to be non-keyframes. The time consumption of tracking keyframes is 69%, while that of non-keyframes are only 31%. Combined with the above, we can conclude that we can achieve a better balance between quickness and exactness compared with the state-of-the-art SLAM systems.

Finally, in order to verify the integrity and feasibility of the proposed algorithm, we evaluate the loop closure detection capability of DO-SLAM. As can be seen from Figures 8 and 9, the accuracy of DO-SLAM with loop detection is improved obviously. Compared with the direct method, DO-SLAM exhibits the function of loop closure detection and solves the problem of drift in long-term operation.

VII. CONCLUSION

We present DO-SLAM, a novel fast and accurate semi-direct visual SLAM framework, that combines the exactness of the feature-based method and quickness of the direct method. Compared with the state-of-the-art feature-based method, we use the direct method to track non-keyframes and accelerate the algorithm without reducing the accuracy and robustness. Compared with the direct method, DO-SLAM exhibits the function of loop closure detection and solves the problem of drift in long-term operation. In the future, we will extend the algorithm to support more types of multi-sensor fusion to increase its robustness in complex environments.

REFERENCES

[1] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 55–81, Jan. 2015.

[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] R. Mur-Artal and J. D. Tardos, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[4] Q. Liu, Z. Wang, and H. Wang, "SD-VIS: A fast and accurate semi-direct monocular visual-inertial simultaneous localization and mapping (SLAM)," *Sensors*, vol. 20, no. 5, p. 1511, Mar. 2020.

[5] H. Li, L. Hao, Q. Zhang, X. Hu, and J. Cheng, "A lifted semi-direct monocular visual odometry," in *Proc. IEEE Int. Conferences Ubiquitous Comput. Commun. (IUCC) Data Sci. Comput. Intell. (DSCI) Smart Comput., Netw. Services (SmartCNS)*, Oct. 2019, pp. 422–426.

[6] 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.

[7] 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.

[8] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A versatile visual SLAM framework," in *Proc. 27th ACM Int. Conf. Multimedia*, Nice, France, Oct. 2019, pp. 21–25.

[9] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[10] Q. Fu, H. Yu, L. Lai, J. Wang, X. Peng, W. Sun, and M. Sun, "A robust RGB-D SLAM system with points and lines for low texture indoor environments," *IEEE Sensors J.*, vol. 19, no. 21, pp. 9908–9920, Nov. 2019.

[11] 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.

[12] 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.

[13] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 1403–1410.

[14] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," 2020, *arXiv:2007.11898*. [Online]. Available: http://arxiv.org/abs/2007.11898

[15] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2320–2327.

[16] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 834–849.

[17] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 2198–2204.

[18] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 3903–3911.

[19] T.-J. Lee, C.-H. Kim, and D.-I.-D. Cho, "A monocular vision sensor-based efficient SLAM method for indoor service robots," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 318–328, Jan. 2019.

[20] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6565–6574.

[21] 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.

[22] 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.

[23] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[24] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct monocular visual odometry by combining points and line segments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4211–4216.

[25] N. Krombach, D. Droeschel, S. Houben, and S. Behnke, "Feature-based visual odometry prior for real-time semi-dense stereo SLAM," *Robot. Auto. Syst.*, vol. 109, pp. 38–58, Nov. 2018.

[26] S. H. Lee and J. Civera, "Loosely-coupled semi-direct monocular SLAM," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 399–406, Apr. 2019.

[27] P. Kim, H. Lee, and H. J. Kim, "Autonomous flight with robust visual odometry under dynamic lighting conditions," *Auto. Robots*, vol. 43, no. 6, pp. 1605–1622, Aug. 2019.
[28] S.-P. Li, T. Zhang, X. Gao, D. Wang, and Y. Xian, "Semi-direct monocular visual and visual-inertial SLAM with loop closure detection," *Robot. Auto. Syst.*, vol. 112, pp. 201–210, Feb. 2019.
[29] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," in *Robotics: Science and Systems*. Zaragoza, Spain, Jun. 2010.
[30] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb. 2004.
[31] 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, 2009.
[32] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," *CoRR*, 2016.
[33] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.
[34] M. Grupp, EVO. [Online]. Available: https://github.com/MichaelGrupp/evo
[35] 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, Portugal, Oct. 2012, pp. 573–580.

SHOUYI LU is currently pursuing the M.S. degree in automotive engineering with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, China. His research interests include visual SLAM, multi-sensor fusion, computer vision, and deep learning.

YONGSHUAI ZHI is currently pursuing the Ph.D. degree in vehicle engineering with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, China. His research interests include vehicle behavior, trajectory prediction, vehicle detection, and deep learning.

SUMIN ZHANG received the Ph.D. degree from Jilin University, Changchun, China, in 2011. He is currently an Associate Professor with the College of Automotive Engineering, Jilin University. He is the author or coauthor of numerous publications in the areas of vehicle controls and power management systems, and has been an in charge of several nationally funded government projects on electric vehicles, modeling and simulation.

RUI HE received the B.S. and Ph.D. degrees from Jilin University, Changchun, China, in 2007 and 2012, respectively. He is currently an Associate Professor with the State Key Laboratory of Automotive Simulation and Control, Jilin University. He is the authors of over 40 peer-reviewed articles in international journals and conferences, and has been in charge of numerous projects funded by national government and institutional organizations on vehicles. His research interests include vehicle control systems, electric vehicles, and autonomous driving.

ZHIPENG BAO is currently pursuing the M.S. degree in automotive engineering with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, China. His current research interests include vehicle trajectory prediction, vehicle trajectory planning, and deep learning.