CamMap: Extrinsic Calibration of Non-Overlapping Cameras Based on SLAM Map Alignment

Jie Xu, Ruifeng Li, Lijun Zhao, Wenlu Yu, Zhiheng Liu, Bo Zhang, and Yuchen Li

***Abstract*—Multiple cameras have emerged as a promising technology for robots and vehicles due to their broad fields of view (FoV) and high resolution. However, there are often limited or no overlapping FoVs among cameras, bringing challenges to estimating extrinsic camera parameters. To overcome this problem, we propose CamMap: a novel 6-degree-of-freedom (DoF) extrinsic calibration pipeline. Following three operating rules, we make a multi-camera rig capture some similar image sequences individually to create sparse feature-based maps with a SLAM system. A two-stage optimization problem is formulated to align the maps and obtain the transformations between them based on bidirectional reprojection. The transformations are exactly the extrinsic parameters. Supporting diverse camera types, the pipeline is available in any texture-rich environment. It can calibrate any number of cameras simultaneously without requiring calibration patterns, synchronization, same resolution and frequency. The pipeline is evaluated on cameras with limited and no overlapping FoVs. In the experiments, we demonstrate our method's accuracy and efficiency. The absolute pose error (APE) between Kalibr and CamMap is less than 0.025.**

***Index Terms*—Extrinsic calibration, multiple cameras, non-overlapping FoVs, SLAM.**

I. Introduction

CAMERAS are cost-effective sensors for mobile robots, micro air vehicles (MAVs) [1], automatic pilots, and augmented reality (AR) [2] to perceive the environment. Based on the simultaneous location and mapping (SLAM) algorithm, they can estimate the motion and recover detailed information about the scene. To improve the robustness of the system, it has become a trend for robots and autonomous driving to install multiple cameras [3], [4], which can provide wider FoV and higher resolution.

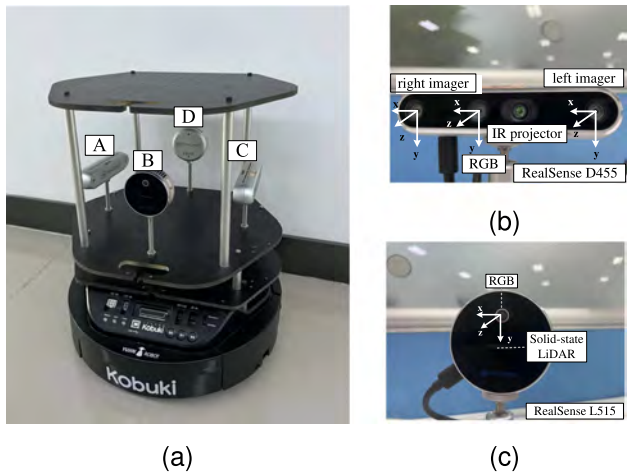

Fig. 1. An example of a rig with non-overlapping cameras. (a) A TurtleBot equipped with two RealSense D455 and two L515. We removed the connecting wire for clear indication. (b) Composition of RealSense d455. The right imager and left imager can be combined into a stereo camera, and the coordinate system of the left imager is used as a reference. The middle RGB module can be used as a monocular camera individually. (c) Structure of RealSense L515. It can be employed as an RGB-D or a monocular camera.

Meanwhile, accurate estimation of extrinsic camera parameters is crucial for associating the information of multiple cameras. However, as shown in Fig. 1(a), the overlapping FoV is often small or nonexistent, which brings significant challenges to the extrinsic calibration. The pattern should be placed far from the cameras in order to be captured by them at the same time. The calibration accuracy will decrease due to the increase of pixel error in corner extraction. For the case with non-overlapping FoVs, a calibration room full of patterns with known relative positions is needed, which is feature expensive and inconvenient.

Focusing on solving the aforementioned problems and making full use of ORB-SLAM3 [5], this letter proposes CamMap: a 6-DoF extrinsic calibration pipeline. It can perform extrinsic calibration of non-overlapping cameras with high accuracy and be used in several types such as monocular, stereo and RGB-D cameras. The setting of reference for each camera is given as follows.

For a monocular camera, the reference is the camera coordinate system, whose origin is on the image plane, the Z-axis is perpendicular to the image plane, the X-axis and Y-axis are parallel to the ones on the image. A stereo camera consists of two monocular ones. It is common to use the coordinate system of the left monocular camera as a reference. What is more, an RGB-D camera consists of an RGB monocular one

Manuscript received 30 May 2022; accepted 12 September 2022. Date of publication 19 September 2022; date of current version 27 September 2022. This letter was recommended for publication by Associate Editor D. Fontanelli and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62073101 and in part by the Self-planned Task of State Key Laboratory of Robotics and System (HIT) under Grant SKLRS202007B. *(Corresponding author: Lijun Zhao.)*

Jie Xu and Lijun Zhao are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China, and also with Wuhu Robot Industry Technology Research Institute, Harbin Institute of Technology, Wuhu 241000, China (e-mail: 498530205@qq.com; zhaolj@hit.edu.cn).

Ruifeng Li, Wenlu Yu, Zhiheng Liu, and Bo Zhang are with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: lrf100@hit.edu.cn; yuwenlu0720@qq.com; liuzhiheng@hit.edu.cn; 1094134448@qq.com).

Yuchen Li is with the Wuhu Robot Industry Technology Research Institute, Harbin Institute of Technology, Wuhu 241000, China, and also with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: yuchenli@hit.edu.cn).

We make the source codes public at github.com/jiejie567/SlamForCalib.

Digital Object Identifier 10.1109/LRA.2022.3207793

TABLE I
APPLICATION REQUIREMENTS OF THE PROPOSED METHOD

Requirements	Whether need
Intrinsic pre-calibration	✓
Time synchronization	×
Same resolution	×
Same frequency	×
Types	**Whether support**
Monocular	✓
Stereo	✓
RGB-D	✓
Models	**Whether support**
Pin-hole	✓
Fish-eye lens	✓
Omnidirectional	×

and a depth measurement module, the RGB coordinate system is considered as a reference. The extrinsic camera parameter means the transformation between two reference coordinate systems. In stereo or RGB-D cameras, there are also easy-to-calibrate extrinsic parameters among their internal modules, which are not considered in this letter. For the case with multiple cameras, we set one of them as a master, others as slaves. The calibration aims to estimate all the transformations between the master camera and the slave ones, the same as the calibration with two cameras. So we only introduce the calibration with two cameras below.

Currently, compared with other visual feature-based SLAM, ORB-SLAM3 surpasses other similar open-source algorithms in positioning accuracy and robustness. With Oriented FAST and Rotated BRIEF (ORB) [6] feature points, it can capture the features evenly and stably in the scenes and has strong robustness indoors and outdoors. It applies to monocular, stereo and RGB-D cameras, using pinhole and fish-eye lens models. Moreover, it is a multiple map system that relies on a new place recognition method with high recall.

Our calibration method, CamMap, is performed by aligning the maps created by the ORB-SLAM3. In other words, the natural scene can be used as a calibration pattern. After building two similar maps with two cameras respectively and finding all the matching map points, the extrinsic parameters is precisely the transformation between maps. The process of calibration only costs tens of seconds. Application requirements of the proposed method can be found in Table I. The main contributions of this letter are listed as follows:

- We propose a camera extrinsic calibration pipeline that integrates ORB-SLAM3 system for various camera types with non-overlapping FoVs, which is applicative in any texture-rich natural environment without calibration patterns and can calibrate any number of cameras at one time. We make the proposed method open-sourced.
- We propose three operating rules for different placement of multi-camera. It is aimed to eliminate theoretical error when the cameras are not synchronized and reduce the error caused by SLAM drifts.
- We introduce cost functions based on bidirectional reprojection in two-stage optimization problem to calculate the extrinsic parameters. After that, an evaluation method for SLAM drifts is provided to determine whether the calibration is successful.

This letter is organized as follows: Section II reviews the related works on existing muti-camera calibration approaches. Section III describes the details of the proposed pipeline, including sequence capture and map creation, similar keyframe detection, extrinsic parameters refinement. Section IV shows experimental results in different natural scenes, followed by discussion in Section V and conclusion in Section VI.

II. RELATED WORK

There are many excellent camera calibration methods with overlapping FoVs. Ince et al. [7] associated measurements from cameras with their relative poses by SLAM. They proposed an iterative optimization method to refine the map, keyframe poses and relative poses between cameras simultaneously. Li et al. [8] proposed an online calibration method for only a limited common FoV by SLAM initialization. They took the inaccurate extrinsic poses as soft constraints to accommodate the calibration errors and performed calibration by matching map points across different cameras between two keyframes.

As the multi-camera application becomes more and more extensive, camera calibration with no overlapping FoV has received more attention. Li and Heng et al. [9] presented a novel feature descriptor-based calibration pattern and a Matlab toolbox. They used the specially designed calibration pattern to easily calibrate the intrinsic and extrinsic parameters of a multiple-camera system. The method only requires that neighbouring cameras observe parts of the calibration pattern simultaneously; the observed parts may not overlap. Kumar et al. [10] used a planar mirror to generate a family of mirrored camera poses that described the real camera pose and estimated the extrinsic parameters from their mirrored poses. However, the entire pattern has to be visible in the camera, which is non-straightforward.

The emerging of SLAM and struct from motion (SFM) brings convenience to the extrinsic calibration. Heng et al. [11] built a highly accurate map with SLAM so that other cameras can use it as a calibration pattern. But it belonged to the "frame to frame" matching method, which did not fully use of the map information. [12] belonged to the hand-eye calibration. The authors performed an extrinsic calibration which found all camera-odometry transforms, inferring metric scale from odometry data. Esquivel et al. [13] applied a structure and motion (SAM) algorithm to estimate relative transformations between the rigidly coupled cameras. Since [12], [13] relied on trajectory matching rather than map alignment, it is deficient in robustness and accuracy. The estimation of rotation is inaccurate due to the tire skidding when the vehicle turns. Besides, some DoFs of calibration parameters may be unobservable when the moving trajectory only has simple changes. Pollok et al. [14] tackled the extrinsic calibration for distributed, non-overlapping multi-camera networks. They reconstructed the scene from a video including areas visible by each network camera. The pose estimation of each surveillance camera is performed individually by assigning 3 dimensions (3D) to 3D correspondences of the map points with SLAM.

The works that we have found which are most similar to our approach are that by Carrera et al. [15] and by Nishiguchi et al. [16], who also used the matching information of SLAM

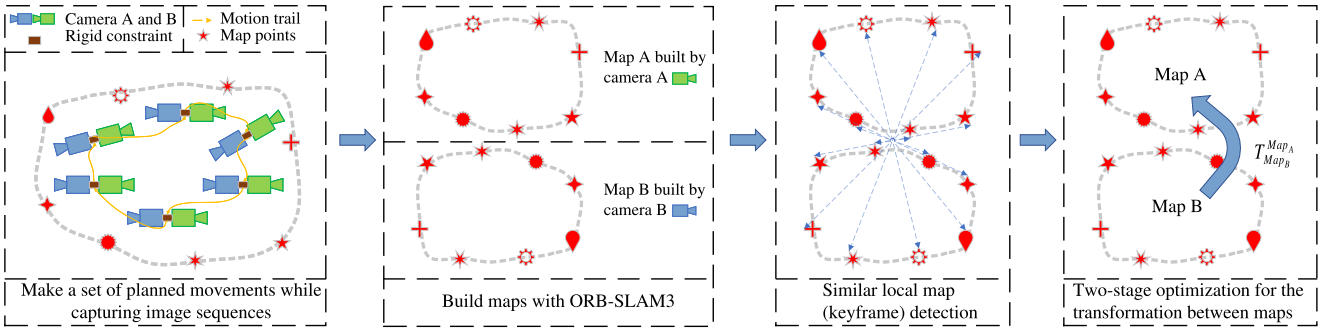

Fig. 2. System overview of the proposed calibration pipeline. Note that the map coordinate system coincides with the camera coordinate system of the first frame, so $T_{Map_B}^{Map_A} = T_{B_1}^{A_1}$.

map. Because of the use of mono visual SLAM rather than a stereo one, they could only calibrate the 3-DoF rotation. Besides, they did not propose different calibration operating rules according to the specific placement of cameras of non-overlapping FoV to reduce the SLAM drifts. What's more, they require the cameras to shoot at the same frequency and time.

III. Methodology

In this section, the proposed method is described in details. As shown in Fig. 2, a rig is installed with two rigid joint cameras A and B. After the rig makes series of planned motions, the cameras capture image streams, which ORB-SLAM3 will process to create maps, including keyframes A_i, B_j for $i = 1, \ldots, m, j = 1, \ldots, n$ and map points P. It is worth noting that the map coordinate system coincides with the first keyframe coordinate system of the camera, which means the transformation $T_{Map_B}^{Map_A}$ between maps is exactly the transformation $T_{B_1}^{A_1}$ between A_1 and B_1, as well as the extrinsic parameters. Then, a similar keyframe detection is performed to find matching keyframes in two maps. Finally, we use a two-stage optimization to estimate the extrinsic parameters by aligning all the pairs of matching map points $(P_k^{A_1}, P_k^{B_1})$ for $k \in N, N = \{1, \ldots, l\}$. The pipeline of CamMap consists of the following steps:

1) Capture some image sequences by cameras as the muti-camera rig makes a set of programmed movements, such as rotating on a small circle, which should be planned according to the relative position of the cameras. At the beginning and end of calibration, the rig should remain stationary.
2) Process image sequences with the ORB-SLAM3 system to create ORB feature-based maps.
3) With the bag of words (BoW) module, similarity detection is performed on all keyframes between two maps to find similar ones and match map points.
4) Align the scales and local maps captured by pairs of similar keyframes to estimate the extrinsic parameters. This process is "frame to frame" alignment, which is the first optimization stage. Meanwhile, use the chi-square test to remove the wrong matching pairs of map points.
5) For the second stage of optimization, use all the correct matching pairs of map points to refine the extrinsic parameters, which is "map to map" alignment. The chi-square test is applied to get the number of inliers. Finally, judge whether the calibration is successful according to the number of inliers and the difference between $T_{B_1}^{A_1}$ and $T_{B_n}^{A_m}$.

The details of these steps will be explained in the following subsections.

A. Sequences Capture and Map Creation

Before performing SLAM in the texture-rich scene, the intrinsic parameters should be calibrated in advance, which will profoundly affect the accuracy of both SLAM and calibration. In the method, it is sufficient that the map coordinate system coincides with the camera coordinate system of the first image. Thus, the cameras can shoot at different frequencies. ORB-SLAM3 system refuses to create a keyframe if there are few feature points in the first image. Therefore, it is supposed to place the cameras in the direction with relatively abundant features at the beginning.

Cameras may not be synchronized during the extrinsic calibration. As a result, if the cameras move at the beginning of the SLAM, assuming that t_0 and t_1 are the moment of two cameras capturing the first image, there will be an error $T_{B_{t1}}^{B_{t0}}$ in the extrinsic parameters $T_{B_1}^{A_1}$:

$$T_{B_1}^{A_1} = T_{B_{t1}}^{A_{t0}} = T_{B_{t0}}^{A_{t0}} \cdot T_{B_{t1}}^{B_{t0}}, \tag{1}$$

where $T_{B_{t0}}^{A_{t0}}$ is the actual extrinsic parameters and $T_{B_{t1}}^{B_{t0}}$ is the transformation of camera B from time t_1 to time t_0. So if camera B moves fast and captures images at a low frequency, the error will be greater. Accordingly, if the cameras keep stationary at the beginning, $T_{B_{t1}}^{B_{t0}}$ will be an identity matrix, and the theoretical error will be avoided.

Theoretically, if the camera keeps stationary at the end of SLAM, $T_{B_n}^{A_m}$ will be equivalent to extrinsic parameters as well. But in the SLAM system, pose estimation is a recursive process. The accumulating error increases gradually, as Fig. 3 shows. Consequently, when finishing calibration, we can judge whether the calibration is successful by comparing $T_{B_1}^{A_1}$ with $T_{B_n}^{A_m}$. An effective way to decrease the error is loop detection. Loop detection determines whether the robot has returned to the previously passed position. If a loop is detected, it will optimize all the SLAM poses and map points to improve the accuracy. What's more, ORB-SLAM3 optimizes the local map based on

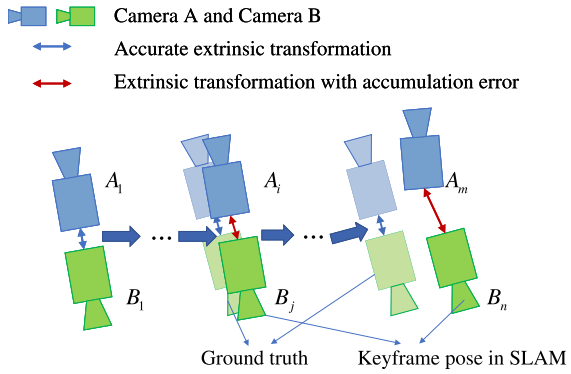

Fig. 3. Influence of accumulating error on calibration results. Because of the SLAM drifts, there will be accumulating error at the end of the SLAM, so $T_{B_1}^{A_1}$ is a more accurate estimation of the camera extrinsic matrix than $T_{B_n}^{A_m}$.

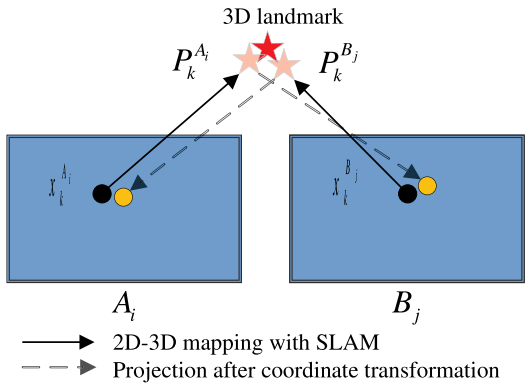

Fig. 4. Bidirectional reprojection of a pair of matching map points on the corresponding keyframes.

the connected keyframe relationship, so if there are overlapping views among them, the accumulating error will be small.

In summary, three operating rules are proposed:

1) The rig should keep stationary at the begin and the end of the SLAM.
2) If there are overlapping FoVs among cameras, it will be suggested to move the rig for a short distance without turning too much, such as the motion trail in Fig. 7(a).
3) Otherwise, it will be better to make the rig move in a loop in order to make the system detect loop closure, such as the motion trail in Fig. 8(b).

To detect loop easier, We modify the loop detection module of ORB-SLAM3 such as increasing the number of keyframes. In summary, the operating rules not only aim to make the rig capture enough similar images, but also keep the accumulating error small. It should be noted that if there is a monocular one among the cameras, we must move the rig rather than only rotate it. Otherwise, monocular SLAM cannot estimate the pose in this instance.

Since the accuracy of the camera changes with distance, we remove map points 10 times deeper than the camera baseline. At the end of SLAM, global Bundle Adjustment (BA) is performed to optimize the position of map points and the pose of keyframes.

B. Similar Keyframe Detection

Similar to loop detection, it is needed to find all the pairs of similar keyframes (A_i, B_j) for $(i, j) \in M$ and matching map points $(P_k^{A_i}, P_k^{B_j})$ in two maps. We use the bag-of-words (BoW) [17] module to find matching keyframes and map points, which can transform feature points into a vector, so we can find similar images by comparing the distance between vectors. Then, we can match the map points by their ORB descriptor. Used to verify the matching relationship, the similarity transformation $S_{B_j}^{A_i}$ between A_i and B_j can be obtained by:

$$P_k^{A_i} = S_{B_j}^{A_i} \cdot P_k^{B_j}, \quad S = \begin{pmatrix} \lambda R & t \\ 0^T & 1 \end{pmatrix}, \tag{2}$$

where R is an orthogonal rotation matrix, describing the relative orientation. t is the translation between the two reference frames. λ accounts for the scale factor between two coordinate systems. When all the cameras are stereo or RGB-D, $\lambda = 1$.

The similarity transformation $S_{B_j}^{A_i}$ can be optimized by the bidirectional reprojection method in (3). As Fig. 4 shows, it projects the pair of matching map points $P_k^{A_i}, P_k^{B_j}$ together into the pixel points on keyframes A_i, B_j. Pixel error is described by Mahalanobis distance in A_i and B_j respectively. If the cameras have different FoVs and resolution, the weight of reprojection error will not be the same. Mahalanobis distance with covariance is more accurate than Euclidean distance to describe the error.

$$\begin{aligned} S_{B_j}^{A_i} = \arg\min_{S_{B_j}^{A_i}} \Bigg\{ & \sum_{k \in N} \\ & \rho \left(\left\| x_k^{A_i} - \pi_A \left(S_{B_j}^{A_i} \cdot P_k^{B_j} \right) \right\|_{\Sigma_A}^2 \right. \\ & \left. + \left\| x_k^{B_j} - \pi_B \left(S_{B_j}^{A_i{}^{-1}} \cdot P_k^{A_i} \right) \right\|_{\Sigma_B}^2 \right) \Bigg\}, \end{aligned} \tag{3}$$

where $x_k^{A_i}$, $x_k^{B_j}$ are the 2D pixel coordinates corresponding to the map point $P_k^{A_i}$ and $P_k^{B_j}$ in the keyframe A_i and B_j respectively. $\pi_A(\cdot)$ and $\pi_B(\cdot)$ are the projection functions of camera A and B, which can map a 3-dimensional vector map point to a 2-dimensional pixel point in the image. We use the Mahalanobis distance $\|\cdot\|_{\Sigma_A}$ and $\|\cdot\|_{\Sigma_B}$ instead of L2 norm. $\rho(\cdot)$ is the robust kernel function to reduce the impact of outliers. The chi-square test are used to eliminate outliers. If the number of correct matching map points exceeds a threshold, we consider that similar keyframes match successfully. This is the "frame to frame" alignment, the first stage of optimization. The initial value is important in nonlinear optimization problems, so we choose $\widetilde{T}_{B_1}^{A_1}$ as the initial value for second stage optimization:

$$\widetilde{T}_{B_1}^{A_1} = T_{A_i}^{A_1} \cdot S_{B_j}^{A_i} \cdot T_{B_j}^{B_1{}^{-1}} (\lambda = 1), \tag{4}$$

where $S_{B_j}^{A_i}$ is calculated by most matching map points.

C. Extrinsic Parameters Refinement

For the second stage of the optimization, transform all the pairs of matching map points $(P_k^{A_i}, P_k^{B_j})$ into $(P_k^{A_1}, P_k^{B_1})$.

Then, the extrinsic calibration can be converted into a nonlinear optimization problem by bidirectional reprojection:

$$T_{B_1}^{A_1} = \arg\min_{T_{B_1}^{A_1}} \left\{ \sum_{(i,j)\in M} \sum_{k\in N} \rho \left(\left\| x_k^{A_i} - \pi_A \left(T_{A_1}^{A_i} \cdot T_{B_1}^{A_1} \cdot P_k^{B_1} \right) \right\|_{\Sigma_A}^2 + \left\| x_k^{B_j} - \pi_B \left(T_{B_1}^{B_j} \cdot T_{B_1}^{A_1 -1} \cdot P_k^{A_1} \right) \right\|_{\Sigma_B}^2 \right) \right\}. \quad (5)$$

After several iterations of nonlinear optimization and chi-square tests for outlier removal, we can estimate the extrinsic parameters $T_{B_1}^{A_1}$ and the number of matching map points that pass the chi-square test. Finally, we can get the transformation $T_{B_n}^{A_m}$ by:

$$T_{B_n}^{A_m} = T_{A_m}^{A_1\ -1} \cdot T_{B_1}^{A_1} \cdot T_{B_n}^{B_1}. \quad (6)$$

The optimization method with "map to map" alignment information is better than the one with the "frame to frame" alignment information. The primary factor affecting the calibration accuracy is the drift in the SLAM system, so we judge whether the calibration is successful from the following two aspects. On the one hand, both $T_{B_1}^{A_1}$ and $T_{B_n}^{A_m}$ are extrinsic parameters theoretically; due to the SLAM drift, $T_{B_n}^{A_m}$ has some deviations from the ground truth. Therefore, it is advised to calibrate again if the difference between $T_{B_1}^{A_1}$ and $T_{B_n}^{A_m}$ is large. On the other hand, if only a few map points pass the chi-square test, the result is the same as the calibration of "frame to frame" alignment, which is not accurate enough.

In particular, for the case where camera A is a stereo or RGB-D, and camera B is a monocula, $T_{B_1}^{A_1}$ is the similarity transformation with scale factor λ shown as:

$$T_{B_1}^{A_1} = \begin{pmatrix} \lambda R_{B_1}^{A_1} & t_{B_1}^{A_1} \\ 0^T & 1 \end{pmatrix}. \quad (7)$$

Because of the orthogonality of the rotation matrix $R_{B_1}^{A_1}$, we can easily get $R_{B_1}^{A_1}$ and $t_{B_1}^{A_1}$, which are actual extrinsic parameters. For the other case where the two cameras both are monocular, considering that the real scale information cannot be obtained in the process of monocular SLAM, we can only calculate the 3-DoF rotation $R_{B_1}^{A_1}$.

IV. Experiments Evaluation

Our ROS calibration code is run on Ubuntu 20.04 with an Intel Core i7-1165G7@2.8 GHz CPU. Two types of cameras, RealSense D455 and L515, are used in the experiment as either RGB-D or monocular cameras. D455 can also be used as a stereo one. We can calibrate the extrinsic parameters among multiple cameras simultaneously by performing the planned movement on the rig only once, and the image sequences are recorded in the rosbag. In order to prove the accuracy of the method and the positive effect of the proposed operating rules, the experimental process and results are shown as follows.

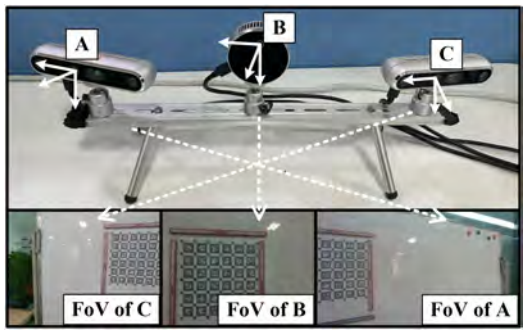

Fig. 5. Three cameras with overlapping FoVs and the AprilTag calibration pattern seen by each camera.

Fig. 6. Three experimental environment: (a) Display board; (b) Office; (c) Parking lot. The green boxes indicate the ORB feature points extracted by ORB-SLAM3.

A. Three Cameras With Overlapping FoVs

The placement of three cameras and the corresponding FoVs are shown as Fig. 5. We apply the proposed pipeline for calibration and use the results by Kalibr [18] as the ground truth. The classical open-source calibration tool Kalibr requires that calibration patterns should appear at all angles and corners in the FoV. But in Fig. 5, camera A and C only have a small overlapping FoV. Therefore, Kalibr cannot be used for calibration directly in this situation. Luckily, the overlapping FoV between camera A and B and the one between B and C are large. Thus, the extrinsic parameters between A and C can be obtained indirectly by $T_C^A = T_B^A \cdot T_C^B$. We conducted experiments on the display board, in the office and parking lot as shown in Fig. 6. The resolution of camera A is 848×480, while both B and C are 640×480. The shooting frequencies of the three cameras are 15 Hz, 30 Hz and 5 Hz respectively. They are all employed as RGB-D cameras here. We use absolute pose error (APE) for the accuracy evaluation:

$$APE = \sqrt{\|log(T_{gt}^{-1} T_{esti})^\vee\|_2^2}, \quad (8)$$

where $(\cdot)^\vee$ transforms Lie groups into Lie algebras, T_{esti} is the estimation of extrinsic parameters while T_{gt} is the ground truth.

In calibration, we assume that the intrinsic parameters are known. The rig should remain stationary at the beginning and the end of the calibration process (operating rule (1)). It should only move slowly for a small distance during the calibration (operating rule (2)), except for the following special cases: in the experiment of display board 2, cameras move fast when ORB-SLAM3 starts; in the experiment of display board 3, we move cameras for a long distance. The experimental results are shown in Table II. Note that in all of the experiments, $T_{B_n}^{A_m}$ is close to $T_{B_1}^{A_1}$, indicating the success of calibration.

TABLE II
ESTIMATED TRANSLATIONS (X, Y, Z), ANGLES (ROLL, PITCH, YAW) BETWEEN TWO CAMERAS AND APE FOR EVALUATION

			Camera$_{AB}$	Camera$_{BC}$	Camera$_{AC}$
Kalibr		Angle(°)	[-0.180, -9.573, -0.712]	[-1.261, -16.265, -3.027]	[-2.872,-25.71,-4.736]
		Translation(m)	[-0.203, -0.023, -0.045]	[-0.171, -0.038, 0.006]	[-0.372, 0.016, -0.067]
CamMap	Display board 1	Angle(°)	[-0.204, -9.634, -0.922]	[-1.502, -15.990, -2.762]	[-3.113, -24.95, -4.896]
		Translation(m)	[-0.211, -0.021, -0.038]	[-0.191, -0.041, 0.010]	[-0.374, 0.019, -0.071]
		APE	0.0121	0.02378	0.0225
	Display board 2	Angle(°)	[-0.888, -7.446, -1.323]	[-1.254, -16.887, -1.984]	[-3.724, -26.024, -1.442]
		Translation(m)	[-0.256, -0.016, -0.030]	[-0.231, -0.081, -0.011]	[-0.394, 0.015, -0.055]
		APE	0.0768	0.0800	0.0982
	Display board 3	Angle(°)	[-1.015, -7.672, -2.721]	[-0.213, -14.005, -1.532]	[-3.577, -27.058, -2.454]
		Translation(m)	[-0.235, -0.027, -0.055]	[-0.251, -0.051, -0.003]	[-0.430, 0.038, -0.032]
		APE	0.0779	0.1075	0.1100
	Office	Angle(°)	[-0.391, -8.890, -1.243]	[-1.436, -18.562, -2.557]	[-3.423, -24.513, -4.964]
		Translation(m)	[-0.237, -0.025, -0.047]	[-0.222, -0.021, 0.012]	[-0.358, 0.010, -0.077]
		APE	0.0400	0.0808	0.0399
	Parking lot	Angle(°)	-	-	[-6.782, -25.816, -6.933]
		Translation(m)	-	-	[-0.311, 0.025, -0.0654]
		APE	-	-	0.1065

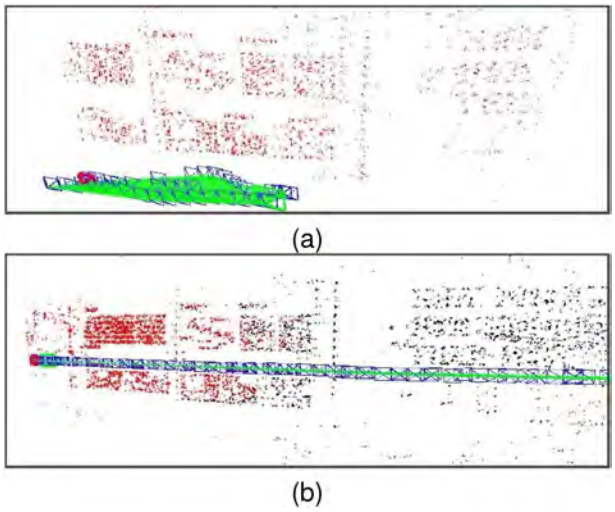

Fig. 7. Maps built by CamMap on display board in Fig. 6(a). Red points are current active map points, their locations are optimized in local BA with ORB-SLAM3, and the accumulating error among map points is small. Black points are historical map points, the accumulating error is large. The blue boxes indicate the keyframe, and the green lines indicate the connected relationship between keyframes. (a) represents experiment on the display board 1 in Table II. (b) represents experiment on the display board 2 and the rig moves for a long distance.

From the first two experiments, it can be inferred that moving at the beginning of calibration will produce the extra error, and the impact will be more significant to the cameras with low frequency. The maps of display board 1 and 3 are shown as Fig 7. In ORB-SLAM3, if several keyframes share overlapping views, these keyframes and corresponding map points are active. Coincidentally, it can be seen that all map points are active in Fig 7(a), which will be optimized in the local BA module to decrease the drifts. Therefore, its APE is smaller than the result on display board 3.

Since the depth module of the camera L515 B is easily affected by the light and cannot build a map in the parking lot, we fail to calibrate the extrinsic parameters outdoors. The feature quality means the accuracy of map point. When a map point is located on the edge of an object or in a far-field, its depth value is not accurate according to the ranging principle. We find that calibration accuracy positively correlates with the quality of indoor features, so it is highly recommended to apply CamMap indoors with high quality features like a display board.

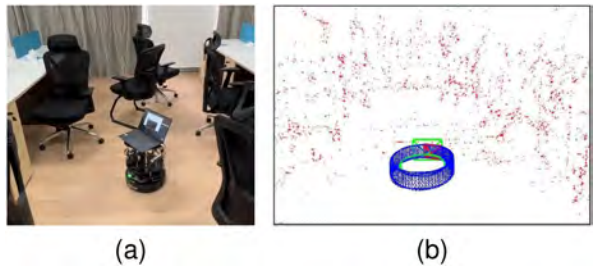

Fig. 8. Experiments in the office. (a) Experiment setup of a TurtleBot rotating automatically. (b)The map built by four cameras.

B. Four Cameras With Non-Overlapping FoVs

We perform calibration in the office shown in Fig. 8(a). In order to reduce the accumulating error, we use the operating rule (3) to make the TurtleBot in Fig. 1(a) rotate one circle automatically. In this experiment, camera A and D are used as monocular ones, B is RGB-D and C is stereo. We consider camera A as the master and others as slaves. One of the maps built by ORB-SLAM3 is shown in Fig. 8(b). The camera's trajectory constitutes a loop. ORB-SLAM3 detects the loop closure, and corrects the pose of keyframes and the location of map points. It should be noted that the movement cannot be a pure rotation, due to the SLAM cannot estimate the translation with monocular camera A and D. CamMap cannot estimate the actual translation between two monocular cameras, A and D, but it can be obtained indirectly by calibrating camera A, C and C, D first. The comparison of calibration results between our method and the manual setting is shown in Table III. APE is not calculated here due to a large manual setting error.

TABLE III
RESULTS OF CALIBRATION

Extrinsic		Manual setting	CamMap
Camera_{AB}	Angle(°)	[0, 90, 0]	[1.982, 88.895, -0.223]
	Translation(m)	[-0.120, -0.035, -0.120]	[-0.123, -0.038, 0112]
Camera_{AC}	Angle(°)	[0, 180, 0]	[0.525, -179.237, 0.0414]
	Translation(m)	[-0.010, 0.000, -0.220]	[-0.008, 0.002, -0.238]
Camera_{AD}	Angle(°)	[0, -90, 0]	[-1.253, -90.462, 0.555]
	Translation(m)	[0.100, -0.035, 0.120]	[-]

V. DISCUSSION

When there are limited or no overlapping FoVs, the traditional method such as Kalibr can not work under these situations. The proposed method CamMap is specially designed for the problems above and can achieve similar accuracy as Kalibr. In addition, our method is easy-to-use and consumes less time. The calibration process can be automated, which is suitable for industrial applications. Our method can be applicable to a variety of scenarios and has no constraints on synchronization. Also, the cameras are not required to have the same frequency and resolution.

However, there are also some limitations. First, due to the limitation of equipment, we did not conduct calibration experiments on the fish-eye model camera. Second, this method is not suitable for the situation where the two cameras cannot capture the same picture. For example, if one of the cameras is installed on the top of the car, and the other is on the underside, it will be impossible to make them take the same picture. Third, when two cameras are all monocular and cannot be formed into a stereo one, our method can only calibrate 3-DoF rotation. Luckily, in ORB-SLAM3, we can use monocular cameras with an inertial measurement unit (IMU) to create maps with absolute scale and optimize 3-DoF translation. It will be included in future work.

VI. CONCLUSION

When there are limited or non-overlapping FoVs among cameras, we cannot use the calibration pattern to calibrate them directly and conveniently. To overcome these challenges, we present a SLAM-based extrinsic calibration pipeline. Features from the natural environment extracted by the modified ORB-SLAM3 are used in calibration instead of the calibration pattern. Three operating rules are proposed to reduce the accumulating error when creating maps. The extrinsic camera parameters are obtained by solving a two-stage optimization problem which further improve the accuracy. Our method is convenient and efficient with no requirements for same resolution, same frequency and time synchronization. It works for RGB-D, stereo and monocular cameras. The experiment results have demonstrated the accuracy of the method as well as the effectiveness of the operating rules. The APE between CamMap and Kalibr is less than 0.025, proving that our method has high accuracy.

It can be eagerly-awaited that the calibration accuracy will be enhanced with the improvement of visual SLAM in the future. Besides, estimating intrinsic parameters can be a part of mapping and map alignment to achieve a more comprehensive calibration pipeline.

REFERENCES

[1] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

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

[3] L. Heng et al., "Project AutoVision: Localization and 3D scene perception for an autonomous vehicle with a multi-camera system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 4695–4702.

[4] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza, "Redesigning SLAM for arbitrary multi-camera systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2116–2122.

[5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[7] O. F. Ince and J. S. Kim, "Accurate on-line extrinsic calibration for a multi-camera SLAM system," in *Proc. IEEE 17th Int. Conf. Ubiquitous Robots*, 2020, pp. 540–545.

[8] A. Li, D. Zou, and W. Yu, "Robust initialization of multi-camera SLAM with limited view overlaps and inaccurate extrinsic calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 3361–3367.

[9] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1301–1307.

[10] R. K. Kumar, A. Ilie, J.-M. Frahm, and M. Pollefeys, "Simple calibration of non-overlapping cameras with a mirror," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.

[11] L. Heng, P. Furgale, and M. Pollefeys, "Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig," *J. Field Robot.*, vol. 32, pp. 775–802, 2015.

[12] L. Heng, L. Bo, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1793–1800.

[13] S. Esquivel, F. Woelk, and R. Koch, "Calibration of a multi-camera rig from non-overlapping views," in *Proc. 29th DAGM Symp. Pattern Recognit.*, 2007, pp. 82–91.

[14] T. Pollok and E. Monari, "A visual SLAM-based approach for calibration of distributed camera networks," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2016, pp. 429–437.

[15] G. Carrera, A. Angeli, and A. J. Davison, "SLAM-based automatic extrinsic calibration of a multi-camera rig," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2652–2659.

[16] K. Nishiguchi, H. Uchiyama, K. Hayakawa, J. Adachi, and R. I. Taniguchi, "On-the-fly extrinsic calibration of non-overlapping in-vehicle cameras based on visual SLAM under 90-degree backing-up parking," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 2021–2028.

[17] D. Galvez-Lpez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[18] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.