Robotics and Autonomous Systems 128 (2020) 103505

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Multi-camera visual SLAM for off-road navigation

Yi Yang [a,*], Di Tang [a], Dongsheng Wang [a], Wenjie Song [a], Junbo Wang [a], Mengyin Fu [a,b]

[a] School of Automation, Beijing Institute of Technology, Beijing 100081, China
[b] Nanjing University of Science and Technology, Nanjing 210094, China

ARTICLE INFO

Article history:
Received 22 October 2019
Received in revised form 29 February 2020
Accepted 16 March 2020
Available online 26 March 2020

Keywords:
Multi-camera
Panorama
Off-road
Simultaneous localization and mapping

ABSTRACT

With the rapid development of computer vision, vision-based simultaneous localization and mapping (vSLAM) plays an increasingly important role in the field of unmanned driving. However, traditional SLAM methods based on a monocular camera only perform well in simple indoor environments or urban environments with obvious structural features. In off-road environments, the situation that SLAM encounters could be complicated by problems such as direct sunlight, leaf occlusion, rough roads, sensor failure, sparsity of stably trackable texture. Traditional methods are highly susceptible to these factors, which lead to compromised stability and reliability. To counter such problems, we propose a panoramic vision SLAM method based on multi-camera collaboration, aiming at utilizing the characters of panoramic vision and stereo perception to improve the localization precision in off-road environments. At the same time, the independence and information sharing of each camera in multi-camera system can improve its ability to resist bumps, illumination, occlusion and sparse texture in an off-road environment, and enable our method to recover the metric scale. These characters ensure unmanned ground vehicles (UGVs) to locate and navigate safely and reliably in complex off-road environments.

1. Introduction

Simultaneous Localization and Mapping (SLAM) technology solves the problem of incrementally constructing a consistent map of environment while positioning the robot at an unknown location in an unknown environment [1,2]. Over the past decade, due to the increasingly prominent performance of visual sensors in terms of image richness, price and data volume, vision-based SLAM has gained more attention in the field of unmanned driving.

There are many widely recognized vSLAM works, such as PTAM [3], LSD-SLAM [4] and DSO [5] based on a monocular camera, and ORB-SLAM2 [6,7] supporting various types of cameras. However, traditional SLAM based on narrow angle camera can only perform well in indoor environments or urban environments with obvious structural features and cannot perform well in environments with more complicated topographical features for a long time. In off-road environments, environmental elements are in a weak motion state, a few examples are the vegetation moving with the wind, the drifting clouds, and the texture of the sand roads after the passage of the car. These factors lead to the lack of stably trackable points in off-road environments. In addition, factors such as direct sunlight, vegetation occlusion, rough roads and sensor failure could further complicate the environment. In such a harsh environment, a single sensor would be very vulnerable to environmental interference, resulting in increased risk of localization failure.

In addition to these problems in motion estimation, complicated off-road environments also pose serious challenges to the loop closure detection of SLAM systems. Loop closure detection enables the SLAM system to correct accumulated errors and maintain globally consistent results. Its essence is place recognition with visual features. Currently, it is common to use the bag-of-words(BoW) model [8] to cluster designed features, such as SIFT [9], SURF [10], ORB [11], to transform features of images into vectors, and then use these vectors to match the scene. This method cannot effectively adapt to various complex situations, especially weak dynamic off-road environments. In order to make place recognition adapt to complex environments and achieve higher accuracy, researchers use CNN to extract deep-seated features of images, and have achieved considerable results [12–17]. Among these methods, NetVLAD [17] is an outstanding feature extractor. Referring to the vector representation of locally aggregated descriptors [18], NetVLAD adds a general VLAD layer to CNN to represent all CNN features of aggregated images as a compact vector with fixed length, which is used to match the scene.

To make the system perform more robustly in such complicated environments, more visual information about the surrounding is a necessity, which prompts us to design a SLAM system based on panoramic cameras. In order to improve the

* Corresponding author.
E-mail address: yang_yi@bit.edu.cn (Y. Yang).

https://doi.org/10.1016/j.robot.2020.103505

adaptability of visual SLAM in off-road environments, this paper proposes a vSLAM method based on multi-camera collaboration, which utilizes the large and overlapping field of view of cameras. The consequential robustness of positioning and mapping enables our method to outperform existing methods in complex off-road environments. In addition, to meet the requirements of loop closure detection in off-road environments, the method of detecting repeated scenes with panorama based on NetVLAD is used in our system to improve the accuracy of loop closure detection in weak dynamic environments where stably detectable texture is very sparse.

Combining various modules based on multi-camera system, we propose a novel vSLAM system. Its main contributions are as follows.

(1) Establish a spatial sensing model of multi-camera system according to the imaging principle of the system. This enables us to quickly obtain the relationship between 3D points and pixels based on captured images without distortion correction.
(2) Design and build a multi-camera collaborative SLAM system framework that combines the advantages of large field of view and overlapping field of view of the multi-camera system. While sensing the environment, this framework enables scale recovery of the results.
(3) Use a deep learning method to detect repeated scenes with panoramas to improve the accuracy of loop closure detection in weak dynamic environments and in the case of sparse detectable textures.

We use panoramic images collected in off-road environment to compare the results of the method based on BoW and our method in loop closure detection. Our method can find the corresponding images of the same scene in the dataset more accurately, and distinguish other images. We use panorama datasets collected in a semi-off-road environment and off-road environment to evaluate our SLAM system, and use the output of a high-precision integrated navigation system collected at the same time as the ground truth to evaluate the results. We also compare other methods with our system on these datasets, including an outstanding method based on monocular cameras and a famous method based on multi-camera systems, and our system achieve more robust tracking and more precise localization result.

2. Related work

Since the last century, researchers began to design panoramic vision systems to obtain more visual information for use in various environments. Up to now, panoramic vision systems can be divided into three categories: catadioptric vision system, fisheye camera and multi-camera vision system. Corresponding SLAM systems are then divided into three categories. The first two types of panoramic systems appeared earlier, and there are many corresponding researches about SLAM. As an example of each category, Scaramuzza et al. used a catadioptric system to estimate 2D motion of a vehicle [19]; Cremer and coworkers proposed a real-time direct monocular SLAM algorithm [20] for an omnidirectional fisheye camera.

In recent years, multi-camera panoramic systems are more and more popular because of their low degree of environmental information compression and relatively small distortion. The study of space geometry for multi-camera systems has begun more than a decade ago. Researchers try to unify the imaging process of cameras with different poses. Pless et al. proposed the generalized camera model, corresponding generalized epipolar constraint and generalized point reconstruction [21]. On this basis, there are many related works on localization, and corresponding algorithms are gradually improved. Stewénnius et al. proposed a minimal solution for the generalized essential matrix [22], which can get a full 6 degrees of freedom pose of a system with six pairs of matching points. To further reduce the demand for the number of matching points when calculating the pose of a system, and to get a more suitable method combined with RANSAC algorithm, Gim et al. proposed a 2-point algorithm based on Ackermann constraint of ground vehicles [23]. They also designed a SLAM system based on it, and implemented the SLAM system for an unmanned ground vehicle.

At present, estimators used in SLAM systems based on nonlinear optimization is the state-of-the-art, and the corresponding algorithm for indirect SLAM system is bundle adjustment (BA) [24]. However, as far as we know, researchers pay little attention to the combination of generalized camera model and BA. Currently, BA is based on the conventional camera models, which incorporate the calibrated relative poses between cameras into the derivation process. Therefore, many schemes do not use the generalized camera model, instead they mainly rely on the corresponding BA algorithm to estimate the trajectory of the system and the positions of map points.

Sharf group used multiple super wide-angle cameras to assist visual positioning of the drone in the air [25] to make it adapt to complex environments with sparse texture. They modified PTAM for multi-camera systems and proposed MCPTAM. They introduced the Taylor camera model in their system, designed a multikeyframe structure and corresponding graph constructions for bundle adjustment, and improved the details of some parts of PTAM algorithm framework. MCPTAM can be used to locate the multi-camera system without overlapping field of view. If the multi-camera system is triggered synchronously and there are overlapping fields of view between the cameras, MCPTAM can also realize localization and map construction with metric scale. With the help of the priori information like a chessboard, MCPTAM can also calibrate the internal and external parameters of a multi-camera system. In their follow-up work, the framework of MCPTAM was further improved [26]. Afterwards, they improved MCPTAM for scale recovery [27]. Even if the cameras have non-overlapping fields of view, improved MCPTAM can estimate the state of non-degenerate motion with metric scale. However, the initialization of MCPTAM makes some strong assumptions about the environment, which strictly limits its application.

Considering the payload capability of a drone, Yang et al. proposed a highly efficient panoramic SLAM scheme to reduce the computer performance requirements [28]. They installed a forward camera and a downward camera on a drone. In order to reduce the computational burden of matching, the maps of cameras are totally separated. The system does not match image points from different cameras, and the only relationship between cameras in this SLAM system is their pose transformation relationship. Compared with MCPTAM, they made more improvements in all aspects of PTAM. They used the combination of FAST corner detector and BRIEF descriptor to improve the accuracy of point matching, which makes the system more robust. In [29], they introduced loop detection based on BoW in SLAM system to make the estimation of system states globally consistent. Because of the separate processing of information from cameras, the SLAM system cannot estimate the states of the drone and the surroundings with metric scale in an unknown environment. In practice, the SLAM system initializes the metric map with the help of a standard helicopter landing pad.

Seok et al. proposed an omnidirectional visual odometry system using a multi-camera system with a large field of view and wide baseline [30]. The multi-camera system was mounted on the rooftop of a vehicle. They proposed a hybrid projection

model to make the warped images continuous and smooth, and they combined the hybrid model, ORB detector and matcher to match the features on different images captured by adjacent cameras. They used the KLT tracking algorithm to perform the intra-view feature processing and it is more efficient than the descriptor matching method. They also proposed a multi-view P3P RANSAC algorithm for robust pose estimation. In order to deal with the deformation and motion of the camera caused by shocks, vibrations and the heat, they performed an online extrinsic calibration in the optimization. Because of the wide baseline, their system can estimate system pose with high precision in large-scale environments. They evaluated their system with synthetic datasets and real datasets collected in urban environments and the performance of their system in dynamic urban environments was demonstrated.

Urban et al. extended ORB-SLAM to a multi-camera system version and proposed Multicol-SLAM [31]. The system can reconstruct part of the scene with metric scale with the overlapping field of view between cameras during initialization. Compared with MCPTAM and SLAM system proposed by Yang, Multicol-SLAM is more robust and has fewer assumptions about the environment and the motion of the vehicle. However, to be used in off-road environments, the system still needs some upgrades. Since BoW is used for loop detection, the effect of loop detection is not good in off-road environments, which affects the global consistency and robustness of the system. In addition, the system does not use overlapping field of view for scale recovery in the tracking process, which can cause scale drift. These problems are taken into consideration in our SLAM system and we make improvements accordingly.

3. Multi-camera system space perception model

By modeling the imaging process of multi-camera system, the projection relationship between 3D points in space and 2D points in imaging plane can be obtained. A suitable model for imaging process is important for the precision of SLAM system. In this section, we describe the camera imaging model and the multi-camera space perception model we choose.

3.1. Camera imaging model

To make our multi-camera system space perception model applicable to different type of cameras, we utilize the general camera model proposed by Scaramuzza to model the imaging process [32]. The principle of model is shown in Fig. 1.

The model can represent not only the imaging process of narrow angle camera, but also the wide-angle camera with field of view larger than 180°. The mathematical expression of camera model is shown in (1).

$$\begin{aligned}\lambda g(\mathbf{m}) &= \frac{\lambda[x, y, f(x, y)]^T}{\left\|[x, y, f(x, y)]^T\right\|} \\ &= \frac{\lambda[x, y, f(\rho)]^T}{\left\|[x, y, f(\rho)]^T\right\|} \\ &= \mathbf{P}\end{aligned} \tag{1}$$

In (1), $g(\mathbf{m})$ represent the direction vector from the optical center to point $\mathbf{P}$, $\mathbf{m} = [x, y]^T$ is the point on the imaging plane corresponding to $\mathbf{P}$, λ is the distance between $\mathbf{P}$ and the optical center, $\rho = \sqrt{x^2 + y^2}$ is the distance between $\mathbf{m}$ and the principle point of the imaging plane, function f is related to the distance between the lens and the imaging plane. We express the inverse perspective transformation as $\mathbf{m} = \psi_c^g(\mathbf{P})$. To make

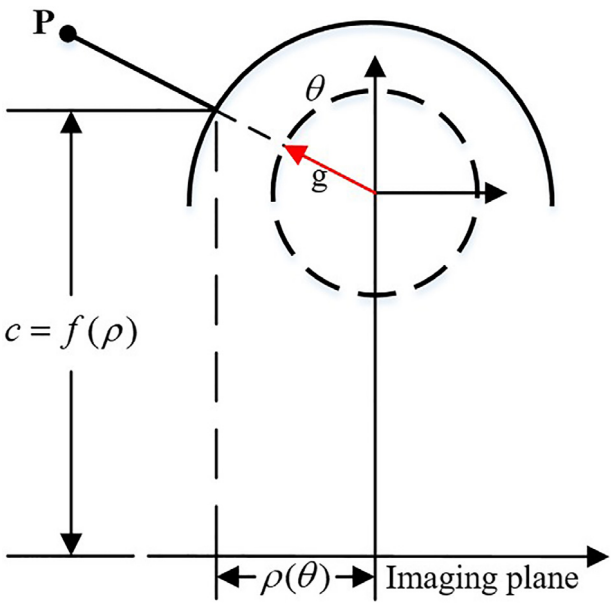


Fig. 1. The general camera model. The projection process from points in space to the imaging plane is modeled through the mapping relationship between points in space and points on a sphere centered by optical center. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the model suitable for any camera, a general Taylor polynomial representation of function f is adopted.

$$f(\rho) = a_0 + a_2\rho^2 + a_3\rho^3 + \cdots + a_n\rho^n \tag{2}$$

Since it is desirable that the z axis of camera coordinate system intersect with the extremum of f, a_1 is 0 and therefore omitted in (2). Due to the limitation of camera processing technology, the imaging plane will not align with the sensor plane. It is thus necessary to consider the relationship between the points on imaging plane and the points on pixel plane. The relationship is shown in (3).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix} \tag{3}$$

$[u, v]^T$ denotes the pixel on the pixel plane, represented by vector $\mathbf{m}'$. The origin of the pixel coordinate system is on the upper left vertex of image. The matrix composed by c, d, e and unity, denoted as $\mathbf{A}$, is used to represent the digital processing under the case of improper alignment between imaging plane and pixel plane. $\mathbf{o}_c = [o_x, o_y]^T$ is the coordinate of principle point of imaging plane under the influence of distortion. With these notations, (3) can be expressed as $\mathbf{m}' = r(\mathbf{m}) = \mathbf{A}\mathbf{m} + \mathbf{o}_c$.

3.2. Multi-camera system space perception model

There are four coordinate systems related to a monocular camera, namely world coordinate system, camera coordinate system, imaging coordinate system and pixel coordinate system. In multi-camera system, data association needs to be considered. Hence, we define the body coordinate system. The transformation relationship between coordinate systems is shown in Fig. 2.

Based on (1) and (3), we can derive the multi-camera system space perception model which shows the relationship between 3D point $\mathbf{P}_i$ in space and 2D point $\mathbf{m}_i'$ in pixel coordinate system.

$$\begin{aligned}\lambda\mathbf{m}_i' &= \mathrm{r}_c(\mathbf{m}_i) \\ &= \mathrm{r}_c\psi_c^g(\mathbf{P}_{ic})\end{aligned}$$

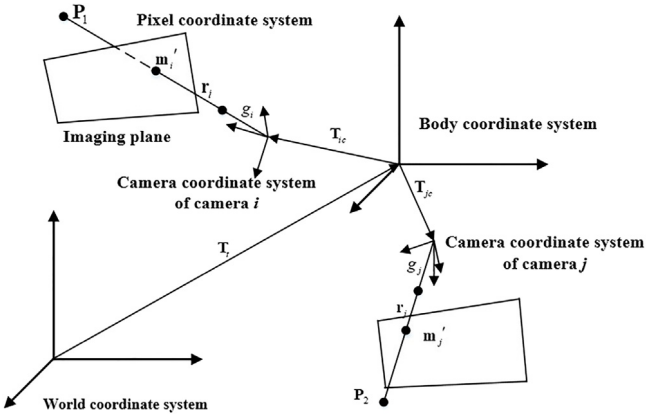


Fig. 2. Multi-camera system space perception model. $\mathbf{T}_t$ is the transformation matrix from world coordinate system to body coordinate system and $\mathbf{T}_{ic}$ is the transformation matrix from camera coordinate system of camera i in the multi-camera system to the body coordinate system. The projection function from camera coordinate system to the imaging coordinate system is denoted by g_i, and we use r_i to represent transformation from the imaging coordinate system to the pixel coordinate system.

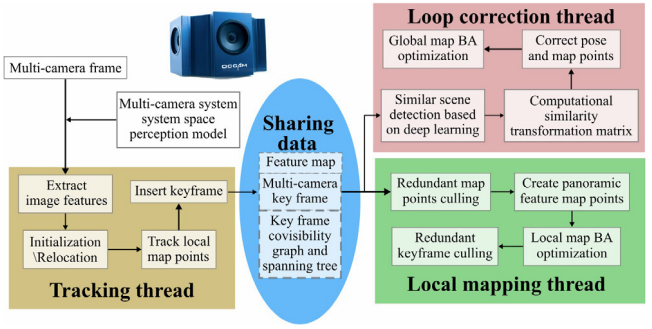


Fig. 3. Complete framework of Multi-camera SLAM system.

$$= \mathrm{r}_c \psi_c^g(\mathbf{T}_t \mathbf{T}_{ic} \mathbf{P}_i) \tag{4}$$

As the cameras in the multi-camera system are rigid connected, $\mathbf{T}_{ic}$ can be obtained by pre-calibration. More specifically, the transformation matrix contains a rotation matrix and a translation vector.

$$\mathbf{T}_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ \mathbf{0} & 1 \end{bmatrix} \tag{5}$$

In (5), $\mathbf{t}_c \in \mathcal{R}^{3\times1}$ is the translation vector representing the displacement along x, y and z axes of body coordinate system and $\mathbf{R}_c \in \mathcal{R}^{3\times3}$ is the rotation matrix used to represent the rotation angle along those axes.

4. Modules of multi-camera SLAM system

The multi-camera SLAM system includes three modules: multi-camera visual localization, panoramic mapping and loop correction. We use multi-threaded programming to achieve parallel processing of these modules. The framework of our SLAM system is shown in Fig. 3. In this section, we introduce the implementation of these modules.

4.1. Multi-camera visual localization

The goal of multi-camera visual localization is to get the 6D pose of UGV in real time. We can estimate the pose quickly with synchronized video frames of multiple cameras based on multi-camera space perception model described in Section 3. The process of localization could be dissected into three states: initialization, tracking and relocalization. We now introduce how each state is carried out in our system.

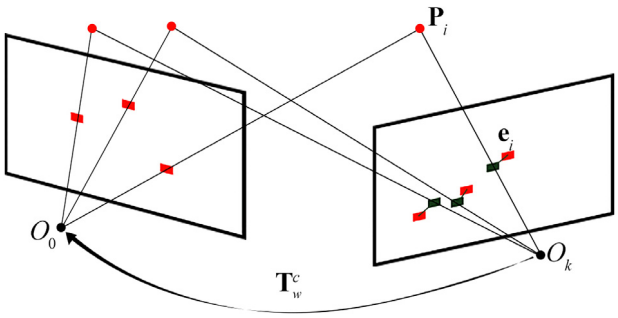


Fig. 4. Schematic diagram of bundle adjustment. Red points are matched pixels and space points, and green points represent the projection points.

4.1.1. Initialization

Our system uses frames captured by multiple cameras with the overlapping fields of view as input. When the first frame is acquired, ORB features in the overlapping fields of view between the cameras will be extracted, and the matching range is reduced according to extrinsic camera parameters. When the number of matched points reaches a threshold, initialization will start, otherwise the current frame will be aborted until the number of matched points meets the requirement.

The depth of 3D points in overlap area can be estimated based on triangulation during system initialization, after which the initial point cloud is inserted into local maps. At the same time, the body coordinate system corresponding to the reference frame is set as the world coordinate system. This process makes full use of the advantages of multi-camera system, which has great field of view and overlapping field of view. The proposed SLAM system can generate initial point cloud with metric scale quickly and extensively, and increase the adaptability to environments with sparse features based on it.

4.1.2. Tracking

After initialization finishes, the pose of the multi-camera system needs to be updated steadily. We reduce the time consumption of features matching process based on uniform motion model. Then we can estimate system pose based on the matching relationship between 3D points and 2D points. Considering possible mismatches and the measurement noise of map points, we first use MLPnP algorithm [33] which can effectively utilize the perception model mentioned in Section 3 to roughly solve the pose, and then we optimize the result based on BA.

The principle of BA is shown in Fig. 4. Suppose that a 3D point $\mathbf{P}$ is the matching object of 2D point $\mathbf{m}$ and it has a projection point $\mathbf{p}$ on the pixel plane. According to (4), the matching relationship between 3D points and 2D points is shown in (6).

$$\mathbf{p}_i = \mathrm{r}_c \psi_c^g(\mathbf{T}_w^c \mathbf{T}_{ic} \mathbf{P}_i) \tag{6}$$

Due to the measurement noise and the errors of poses, there are projection errors $\mathbf{e}_i$ between the projection points and corresponding pixels. Such errors are called reprojection errors, and the relationship between a reprojection error and matching point pair is shown in (7).

$$\begin{aligned} \mathbf{e}_i &= \mathbf{m}_i' - \mathbf{p}_i \\ &= \mathbf{m}_i' - \mathrm{r}_c \psi_c^g(\mathbf{T}_w^c \mathbf{T}_{ic} \mathbf{P}_i) \end{aligned} \tag{7}$$

By adjusting estimated pose $\mathbf{T}_w^c$ to reduce the reprojection errors of matched points, the estimated pose is closer to its actual value. We can then give the objective function of pose optimization problem as shown in (8).

$$\mathbf{T}_w^{c\,*} = \underset{\mathbf{T}_w^c}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \left\| \mathbf{m}_i' - \mathrm{r}_c \psi_c^g (\mathbf{T}_w^c \mathbf{T}_{ic} \mathbf{P}_i) \right\|_2^2 \tag{8}$$

It is supposed that all map points are fixed. The object function can be minimized iteratively, and the local minimum which satisfy certain conditions is regarded as the result of optimization. The initial value given by MLPnP draws the result close to its actual value. In our system, Levenberg–Marquart algorithm with trust region is used to optimize the pose of multi-camera system.

4.1.3. Relocalization

In weak dynamic off-road environments, bumps and leaf occlusion can easily lead to the failure of tracking. It is necessary to relocalize once the number of matching features is too few or the pose estimation totally fails during tracking process. The relocalization means matching points in the current image with points in the local map. If enough matching points are obtained, the relocalization can be considered successful and the tracking process will be continued, otherwise the system will turn back to relocalization until enough matching points are obtained after expanding search scope. The multi-camera SLAM system can search the matching points in all directions, which makes the relocalization process easier.

4.2. Mapping based on multi-camera panoramic vision

Compared with SLAM systems based on monocular camera, our SLAM system has more information to process. We choose to construct a sparse point cloud from feature points with simple data structure as our map, so that the system can perform well in large-scale map construction, and each map point has accurate description which make the map reusable. To improve the real time performance, all the implementations of reconstruction are carried out in mapping thread.

4.2.1. Select criterion of key frames

To avoid the excessive increase of map size and waste of system resources, we only construct map for frames satisfying specific conditions. Considering advantages of multi-camera systems and characters of off-road environments, we set five criterions for key frames:

(1) The interval after initialization or relocalization needs to be longer than 15 frames.
(2) The mapping thread should be idle and the interval from current frame to the last key frame needs to be more than 10 frames.
(3) The number of points matched with map points in current frame should be less than 80% of the map points associated with the last key frame.
(4) The number of points matched with map points in current frame which are not in covisibility graph needs to be more than 100.
(5) The distance between system positions corresponding to current frame and last key frame needs to be more than 10 times the baseline length.

Since panoramic systems can perceive 360° environment information, there is no requirement for rotation in these criterions. When all the criterions are satisfied at the same time, the current frame is regarded as a key frame.

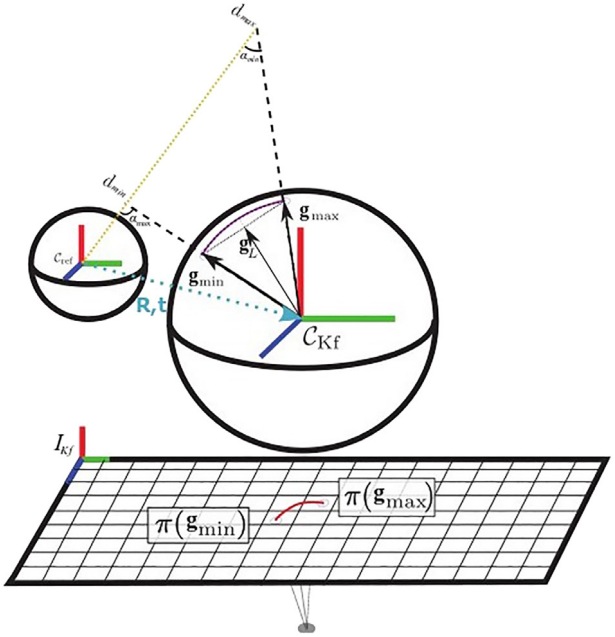


Fig. 5. Depth measurement of map points based on general camera model.

4.2.2. Mapping

Key frames consist of the features extracted from multiple camera images. To represent the covisibility information between key frames, we use key frames as nodes and the number of sharing map points of two frames as the weight of edges to construct a covisibility graph. Greater weight means more observations shared by frames. When the number of covisible points of two frames is less than 15, it is considered that there is no correlation between frames.

The mapping process includes synchronous mapping and asynchronous mapping. Synchronous mapping uses an arbitrary pair of cameras to engage the 3D construction process. As overlapping fields of view only exist between adjacent cameras, there are in total 5 combinations for this process of synchronous mapping since our multi-camera system consists of 5 cameras. Asynchronous mapping exploits the current key frame and previous key frames in covisibility graph to generate map points. Here we use 10 key frames with highest correlation with current key frame in covisibility graph to match with the current key frame, and the total number of asynchronous mapping combinations can reach up to 100 groups. To quickly process tens of thousands or even hundreds of thousands of feature points, we use BoW [8] for matching feature points. The computational complexity of O(*logn*) can be achieved by constructing a vocabulary tree. The mapping process is shown in Fig. 5.

The π_c represents the mapping function from camera coordinate system to the pixel coordinate system of camera c. According to the multi-camera space perception model we have $\pi_c = \mathrm{r}_c \psi_c^g$. The rotation matrix and the translation vector for combination of camera i and camera j is shown in (9).

$$\mathbf{p}_{im} = (\mathbf{T}_{c_i} \mathbf{T}_{t_m}) \mathbf{T}_{mn} (\mathbf{T}_{c_j} \mathbf{T}_{t_n})^{-1} \mathbf{p}_{jn} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{p}_{jn} \tag{9}$$

In (9), $\mathbf{T}_{t_n}$ and $\mathbf{T}_{t_m}$ represent the pose of key frame n and key frame m, $\mathbf{T}_{mn}$ is the transformation matrix to transform the 3D point from camera coordinate system of key frame n to key frame m. $\mathbf{T}_{c_i}$ and $\mathbf{T}_{c_j}$ are the transformation matrixes from the body coordinate system to camera coordinate system i and j respectively. $\mathbf{p}_{im}$ is one of the 3D points observed in camera i in key frame m, and

$\mathbf{p}_{jn}$ is the corresponding point observed in camera j in key frame n.

Resolution of images is limited, and the descriptors of feature points cannot be used in matching in weak dynamic off-road environments when the difference of angles of view is large. Therefore, we need to set limitations for the angle of directional vector of matching points. The limit is set to 3° to 60°, and we denote the corresponding directional vector to be $\mathbf{g}_{max}$ and $\mathbf{g}_{min}$. If the directional vector of a 3D point is fixed, the depth estimation is performed only when the directional vector of another 3D point satisfies the condition shown in (10).

$$\mathbf{g}_L = \beta \mathbf{g}_{max} + (1-\beta)\mathbf{g}_{min} \tag{10}$$

In (10), we have $0 \leq \beta \leq 1$. The depth of the point can be recovered through (11).

$$\lambda_{ref}\mathbf{p}_{ref} = \mathbf{R}\lambda_L \mathbf{p}_L + \mathbf{t} \tag{11}$$

We can solve the equation quickly based on epipolar geometry constraint [34]. After obtaining the 3D coordinates of points, map points are projected to corresponding images, and the map points with negative depth or large reprojection error are deleted from map to help reduce the influence of the measurement noise and model error.

To avoid duplicate map points due to measurement noise, we match new map points with existing map points in a certain space range. If two points are matched they will be fused, and the fused point is located at the midpoint of the connection line of two origin points, and the descriptor of the fused point comes from new map point.

4.2.3. Optimization of local map

To improve the precision of map points, we need to optimize the local map. The local map consists of new map points generated with the current key frame and map points from the key frames with high relevance with the current key frame in covisibility graph. Optimization of map points is also based on bundle adjustment and its objective function is shown in

$$\mathbf{P}_i^* = \underset{\mathbf{P}_i}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{n} \left\| \mathbf{m}_i' - \mathrm{r}_c \psi_c^g(\mathbf{T}_w^c \mathbf{T}_{ic} \mathbf{P}_i) \right\|_2^2 \tag{12}$$

The notation of each term in (12) is the same with that in (8). The only difference here is of the fixed pose of the system.

4.3. Loop closure detection and correction

Loop closure detection is the ability of a system to detect whether it returns to a previous scene or not, and loop correction based on the detection can greatly improve the global consistency of a system. We design a loop closure detection algorithm based on panorama and NetVLAD, and implement it in an independent thread. Based on loop closure information, we correct the trajectory and map in the same thread.

4.3.1. Loop closure detection based on panorama and netvlad

We stitch the images captured by cameras together and use NetVLAD to extract features of stitched images and transform them into 4096-dimensional vectors. To judge the similarity of scenes, we calculate the Manhattan distance between two vectors as shown in (13).

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{13}$$

In (13), $\mathbf{x}$ and $\mathbf{y}$ are the VLAD vectors belonging to different scenes, $n = 4096$ is the dimension of vectors, $d(\mathbf{x}, \mathbf{y})$ is the Manhattan distance of two vectors. To limit the similarity information into [0, 1] interval, we process it further.

$$Sim(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + d(\mathbf{x}, \mathbf{y})} \tag{14}$$

In (14), $Sim(\mathbf{x}, \mathbf{y})$ is the similarity information of two scenes. The closer its value is to 1, the higher the similarity of scenes is, the more likely there will be a loop.

4.3.2. Correction of global trajectory and map

Based on information of similar key frames provided by loop closure detection, we match the map points corresponding to similar key frames and then obtain a Sim3 transform matrix [35] with matching points. First, we assume that initial scale of poses of all key frames except similar key frames in spanning tree are 1 and transform all poses $\mathbf{T}_t$ into Sim3 transform matrices $\mathbf{S}_t$. We also transform the relative pose transformation between key frames $\Delta\mathbf{T}_{ij}$ into similar transformation $\Delta\mathbf{S}_{ij}$. Then we define the transform error between two key frames as follows. In (15), we represent logarithmic mapping as *log* [36].

$$r_{i,j} = log_{sim(3)}(\Delta\mathbf{S}_{ij}\mathbf{S}_i\mathbf{S}_j^{-1}) \tag{15}$$

We regard the transformation between key frames $\Delta\mathbf{S}_{ij}$ as measurement, and minimize the error by adjusting $\mathbf{S}_i$ and $\mathbf{S}_j$ to keep the transformation between key frames unchanged. After several iterations, the optimal values are obtained, and the similar transformation errors of loop are propagated to the poses of all key frames to correct trajectory. Then we can obtain the corresponding transform matrices of poses.

$$\hat{\mathbf{T}}_t = \begin{bmatrix} \mathbf{R} & \mathbf{t}/s \\ \mathbf{0} & 1 \end{bmatrix} \tag{16}$$

We need to correct the map points corresponding to all key frames based on the above results.

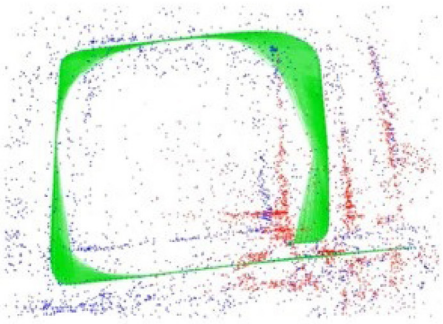
$$\hat{\mathbf{p}} = \hat{\mathbf{S}}_t^{-1}(\mathbf{T}_t \mathbf{p}) \tag{17}$$

An example of the effect of loop correction is shown in Fig. 6.

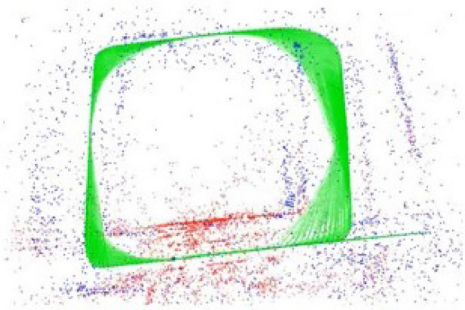
5. Experimental result

We collect test data with a panoramic camera and an integrated navigation system installed on the vehicle as shown in Fig. 7, and perform some experimental validation of the SLAM algorithm with our sequences. The panoramic camera consists of five cameras, each with a resolution of 752 * 480 and a frame rate of 20 Hz. The integrated navigation system outputs centimeter-level differential positioning result at 20 Hz, and its output is used as the ground truth in our experiments.

Environments in which we collect data are shown in Figs. 8 and 10a, and the satellite images of these environmental areas are shown in Figs. 9 and 11 respectively. We choose a route with a shape of "8" and a length of 1400 m in the off-road area and a 300-meter-long circular route in the semi-off-road area to drive the vehicle to collect data. From the satellite image of the off-road area, it can be seen that there are only gravel roads and irregular vegetation in a large area, and there are no structured roads and buildings. Therefore, features that can be tracked stably in captured images are very sparse, and the bumpy road condition causes violent jitters when driving. There are more features that might be used in the semi-off-road area since there are structured roads, a large number of buildings and a parking lot around the area. However, we only extract features in the green mask area shown in Fig. 10b when we run our system to test the effectiveness of the algorithm in complex environment, and most of the features on these objects are not used. Both routes



(a) The spanning tree and the map points before loop correction



(b) The spanning tree and the map points after loop correction

Fig. 6. An example of loop correction when our system is used in the experiment.

Fig. 7. The vehicle used for collecting data with a panoramic camera and an integrated navigation system.

Fig. 8. One of the images captured on the off-road experimental area. The off-road experimental area has undulating slopes with a slope angle up to 20° and the surrounding environment is mainly trees, weeds, sand and stones of different sizes.

Fig. 9. The satellite imagery of the off-road experimental area.

(a) One of the images captured on the semi-off-road experimental area.

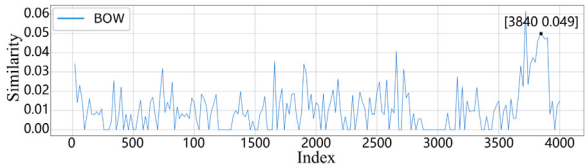
(b) The mask used in semi-off-road experiment, which is represented as the green area in the figure.

Fig. 10. One of images captured on semi-off-road area and corresponding. The semi-off-road experimental area has some characteristics of off-road environment and some characteristics of structured environment. The system only use the area represented as mask when extracting features, so that we can test the effect of system in complex environment.

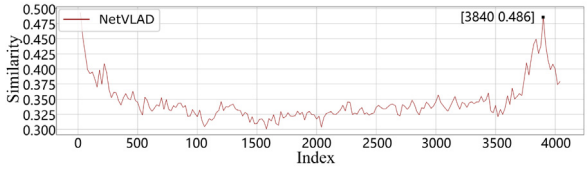
Fig. 11. The satellite imagery of the semi-off-road experimental area.

contain one or more loops so that they can also be used to verify the validity of loop closure detection algorithm.

The experiments are performed on a laptop with a 2.8 GHz CPU, 16 GB RAM and GTX-1050Ti. All algorithms mentioned below run in the ROS environment configured in Linux operating system, and they are implemented in C++ except for NetVLAD algorithm which is based on TensorFlow architecture [37] and implemented in python.



(a) Similarity information between each selected frame and reference frame calculated based on BoW method



(b) Similarity information between each selected frame and reference frame calculated based on our method

Fig. 12. Comparison of experimental results of different loop closure detection algorithms.

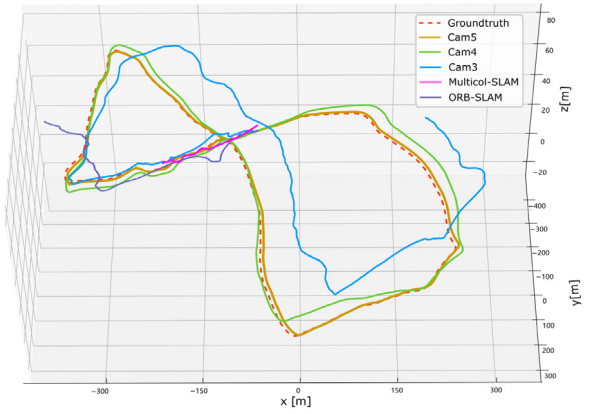
5.1. Loop closure detection accuracy

We use the sequence collected on the off-road area to test the performance of loop closure detection. We choose the first stitched image as the reference and evaluate similarity between the reference image and each stitched image in the sequence. When we run our SLAM system in the sequence, we use NetVLAD and BoW based on ORB features to evaluate the similarity between stitched images from all key frames and the first frame respectively. In the experiment, the 3840 key frame is the loop closure frame corresponding to the first frame.

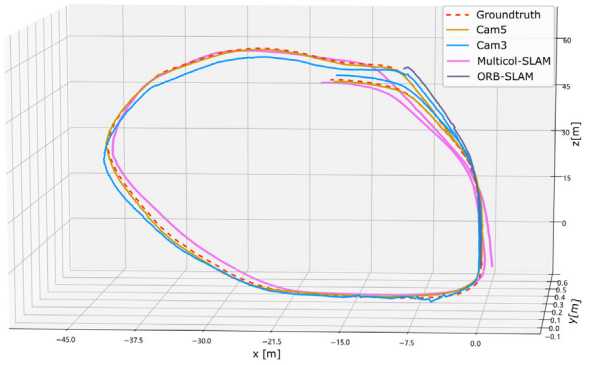
We collect similarity information provided by different algorithms every 20 key frames, as illustrated in Fig. 12. The similarity information calculated based on BoW method at that frame is higher than average, but it is not the highest one and the curve contains many peaks which will lead to high false positive rate. The reason for its poor performance is that the stable recognizable features in the experimental environment are sparse and the discrimination is not high. The similarity information calculated based on our method performs much better, as shown in Fig. 12b. The curve only contains one peak at the 3840 frame and the value of this frame is much higher than any other frame. In the experiment, our method can run at 33 Hz which meets the real-time requirement of SLAM system for loop closure detection.

5.2. Performance of our SLAM system on sequences

We evaluate several modes of our SLAM system on the datasets collected in a semi-off-road environment and an off-road environment, respectively. These modes differ by the numbers of cameras that are used to collect images. When we run our SLAM system, image data are processed appropriately so that the system could process images captured by different numbers of cameras according to the mode we choose in experiments. For comparison, we also run Multicol-SLAM in 5-camera mode and the monocular vision of ORB-SLAM. To highlight advantages of large field of view, we limit the number of feature points extracted by each camera to 2000 when using the off-road sequence, or 1500 when using the semi-off-road sequence. The comparison of trajectories and ground truth provided by integrated navigation system is shown in Fig. 13, in which cam(n) represents the number of cameras corresponding to images used in the multi-camera SLAM system. As we can see, in the off-road environment, our system can correct the trajectory based on loop closure detection in 4-camera mode and 5-camera mode



(a) Localization trajectories in the off-road environment



(b) Localization trajectories in the semi-off-road environment

Fig. 13. Comparison results of localization experiments in different environments.

and locate the vehicle effectively, but the system fails to locate in 3-camera mode, and neither Multicol-SLAM nor ORB-SLAM succeed; in the semi-off-road environment, our system performs well in 3-camera mode and 5-camera mode, and Multicol-SLAM also performs well, while ORB-SLAM fails to locate. The reason why our system performs better than Multicol-SLAM in the off-road environment is that our system uses the overlapping field of view between cameras to generate additional map points by triangulating all key frames. This is not only conducive to scale recovery, but also makes the distribution of local map points around the multi-camera system more uniform. When we run Multicol-SLAM in an environment with sparse stably trackable features, the tracked map points gather behind the motion direction, which makes the number of the tracked points unstable. The number of tracked points are reduced rapidly, and eventually the tracking thread is lost. However, in the semi-off-road environment, more than half of the cameras have some stably trackable features in the field of view, which makes the tracking of Multicol-SLAM more robust. Since ORB-SLAM only uses information provided by one camera, this method often encounters failures of tracking in complicated environments. Moreover, when such a failure occurs, it is difficult for the system to recover by relocalization because the camera could not often capture images with obvious texture in such environments. The result shows that the larger the field of view is, the more features are tracked stably, and the closer the trajectory provided by our system are to the ground truth. It also shows that even if the functions of some cameras are damaged due to internal or environmental factors, the system can process information normally to a certain extent, and that enables our system to effectively cope with common problems

Table 1
Time consumed by algorithms under different conditions. The unit of time is millisecond.

Environment	Algorithm	The number of tracked map points	Max	Min	Median	Mean
Off-road	Proposed(Cam5)	264	81.633	57.214	69.745	69.938
	Proposed(Cam4)	225	77.899	55.969	67.021	66.891
	Proposed(Cam3)	173	70.974	52.121	63.269	63.104
	Multicol-SLAM	159	100.341	38.485	53.177	54.673
	ORB-SLAM	57	29.134	26.547	27.323	27.744
Semi-off-road	Proposed(Cam5)	239	77.571	53.285	66.832	67.276
	Proposed(Cam3)	161	63.254	47.853	58.579	60.017
	Multicol-SLAM	176	77.479	35.8324	49.3923	48.2649
	ORB-SLAM	71	31.615	28.796	29.659	29.587

Table 2
Localization precision of algorithms under different conditions.

Environment	Algorithm	Length of trajectory (m)	ATE (m)	ARE (deg)	RTE (%)	RRE (deg/m)
Off-road	Proposed(Cam5)	1390	1.7801	2.220	1.0558	0.00118
	Proposed(Cam4)	1390	8.1312	2.747	2.2391	0.00269
	Proposed(Cam3)	1183	30.0407	3.572	3.5557	0.01397
	Multicol-SLAM	269	31.0567	5.165	11.553	0.0192
	ORB-SLAM	321	40.2892	5.774	7.0497	0.04068
Semi-off-road	Proposed(Cam5)	283	0.4067	0.0177	0.4623	0.000961
	Proposed(Cam3)	283	1.5282	0.1408	1.0088	0.0012419
	Multicol-SLAM	283	1.6314	0.5253	1.1062	0.0044286
	ORB-SLAM	67	3.0794	1.0316	3.7212	0.073556

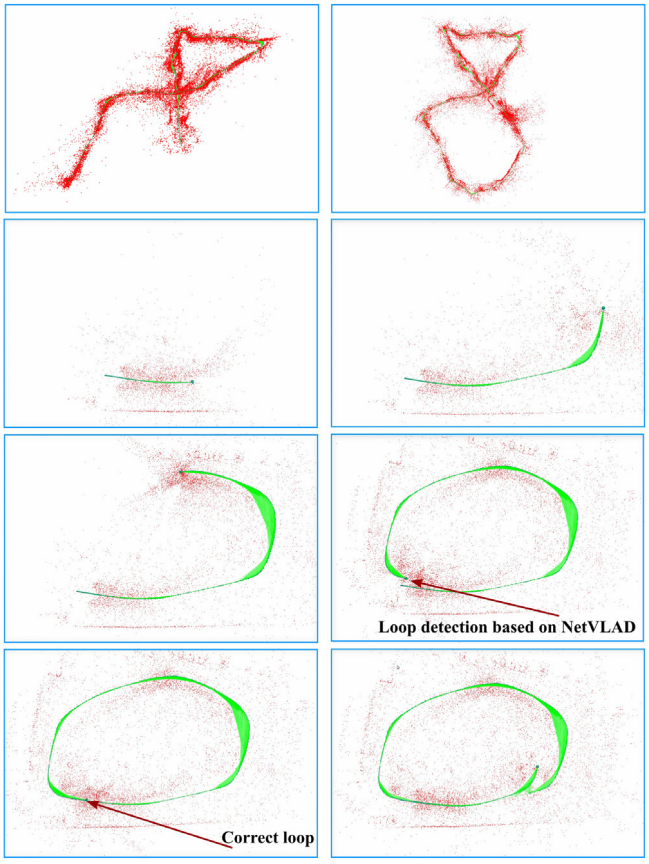


Fig. 14. Results of mapping when our system runs in 5-camera mode. The first two figures show experimental results in semi-off-road environment and others are experimental results in off-road environment.

one would expect to encounter in an off-road environment, such as severe illumination changes, leaf occlusion, loss of stable tracking features, steep roads, failure of some cameras, etc. Such robustness guarantees our system exceptional performance in complex environments.

We illustrate some figures in Fig. 14 to show the process of localization and mapping when our SLAM system runs in 5-camera mode. These figures show processes when our system runs with different sequences, and both groups of images show the effect of loop correction on localization and mapping. In these two processes, the pose estimated by our system are more or less deviated, resulting in errors in map. After loop correction, not only the trajectories are corrected, but also a map with consistency is constructed. With loop correction, our system has the ability to construct a global consistent map.

We evaluate the performance of SLAM system with a trajectory evaluation algorithm [38] and results are shown in Tables 1 and 2. In these tables, ATE represents absolute translation error, ARE represents absolute rotation error, RTE represents relative translation error, RRE represents relative rotation error. Tables show comparison of running time and localization precision between ORB-SLAM, Multicol-SLAM and our system in different modes. Since ORB-SLAM uses monocular images, its computation costs less, resulting in shortest time consumption. Our system adopts multi-thread parallel processing algorithm, so the time consumed does not increase linearly with the number of cameras, but more slowly, and the requirement of the UGV for running speed could be met under our experimental condition. The process of adding additional map points based on the overlapping field of view makes our system consume more time than Multicol-SLAM, but greatly improves the robustness of our system. Table 2 shows that the system localization precision will increase as the field of view expands, which proves the effectiveness and feasibility of our multi-camera SLAM system in off-road environments.

6. Conclusion

This paper proposes a panoramic vision SLAM system based on multi-camera collaboration with the ability to work in complex environments according to the characters of off-road environment. The system combines the advantages of large field of view

and overlapping fields of view of multi-camera system to construct a vision panoramic perception model, which can perceive environmental information with metric scale in a wide range, and improve the adaptability of our method to weak texture off-road environment. At the same time, multiple cameras are independent of each other, and the problem with any single camera will not affect normal functions of the system.

Our system is robust to object occlusion, direct sunlight and failure of some cameras, a feature that could be immensely useful for the navigation of UGV. The multi-thread implementation of the algorithm enhances the real-time performance of our system. Experiments in challenging off-road environment show validation of the system, and the robustness of our system is highlighted in comparison with the performance of ORB-SLAM and Multicol-SLAM.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC, 61973034, 61473042, U1913203 and 61903034).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2020.103505.

References

[1] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robot. Autom. Mag. 13 (2) (2006) 99–110.
[2] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): Part II, IEEE Robot. Autom. Mag. 13 (3) (2006) 108–117.
[3] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society, 2007, pp. 1–10.
[4] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: European Conference on Computer Vision, Springer, 2014, pp. 834–849.
[5] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, IEEE Trans. Pattern Anal. Mach. Intell. 40 (3) (2017) 611–625.
[6] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.
[7] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.
[8] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.
[9] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[10] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: European Conference on Computer Vision, Springer, 2006, pp. 404–417.
[11] E. Rublee, V. Rabaud, K. Konolige, G.R. Bradski, ORB: An efficient alternative to SIFT or SURF, in: ICCV, Vol. 11, Citeseer, 2011, p. 2, no. 1.
[12] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Advances in Neural Information Processing Systems, 2014, pp. 487–495.
[13] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1452–1464.
[14] Y. Hou, H. Zhang, S. Zhou, Convolutional neural network-based image representation for visual loop closure detection, in: 2015 IEEE International Conference on Information and Automation, IEEE, 2015, pp. 2238–2245.
[15] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, M. Milford, Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free, Proc. Robot.: Sci. Syst. XII (2015).
[16] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, J. Gonzalez-Jimenez, Training a convolutional neural network for appearance-invariant place recognition, 2015, arXiv preprint arXiv:1505.07428.
[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5297–5307.
[18] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition, IEEE Computer Society, 2010, pp. 3304–3311.
[19] D. Scaramuzza, R. Siegwart, Monocular omnidirectional visual odometry for outdoor ground vehicles, in: International Conference on Computer Vision Systems, Springer, 2008, pp. 206–215.
[20] D. Caruso, J. Engel, D. Cremers, Large-scale direct slam for omnidirectional cameras, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 141–148.
[21] R. Pless, Using many cameras as one, in: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, Vol. 2, IEEE, 2003, pp. II–587.
[22] M. HenrikStewénius, K. Aström, D. Nistér, Solutions to Minimal Generalized Relative Pose Problems, Citeseer, 2005.
[23] G. Hee Lee, F. Faundorfer, M. Pollefeys, Motion estimation for self-driving cars with a generalized camera, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2746–2753.
[24] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle adjustment-a modern synthesis, in: International Workshop on Vision Algorithms, Springer, 1999, pp. 298–372.
[25] A. Harmat, I. Sharf, M. Trentini, Parallel tracking and mapping with multiple cameras on an unmanned aerial vehicle, in: International Conference on Intelligent Robotics and Applications, Springer, 2012, pp. 421–432.
[26] A. Harmat, M. Trentini, I. Sharf, Multi-camera tracking and mapping for unmanned aerial vehicles in unstructured environments, J. Intell. Robot. Syst. 78 (2) (2015) 291–317.
[27] M.J. Tribou, A. Harmat, D.W. Wang, I. Sharf, S.L. Waslander, Multi-camera parallel tracking and mapping with non-overlapping fields of view, Int. J. Robot. Res. 34 (12) (2015) 1480–1500.
[28] S. Yang, S.A. Scherer, A. Zell, Visual SLAM for autonomous MAVs with dual cameras, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 5227–5232.
[29] S. Yang, S.A. Scherer, X. Yi, A. Zell, Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles, Robot. Auton. Syst. 93 (2017) 116–134.
[30] H. Seok, J. Lim, ROVO: Robust omnidirectional visual odometryfor wide-baseline wide-fov camera systems, 2019, arXiv preprint arXiv:1902.11154.
[31] S. Urban, S. Hinz, MultiCol-SLAM-A modular real-time multi-camera SLAM system, 2016, arXiv preprint arXiv:1610.07336.
[32] D. Scaramuzza, A. Martinelli, R. Siegwart, A flexible technique for accurate omnidirectional camera calibration and structure from motion, in: Fourth IEEE International Conference on Computer Vision Systems (ICVS'06), IEEE, 2006, p. 45.
[33] S. Urban, J. Leitloff, S. Hinz, Mlpnp-a real-time maximum likelihood solution to the perspective-n-point problem, 2016, arXiv preprint arXiv:1607.08112.
[34] G. Ben-Artzi, T. Halperin, M. Werman, S. Peleg, Epipolar geometry based on line similarity, in: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE, 2016, pp. 1864–1869.
[35] H. Strasdat, J. Montiel, A.J. Davison, Scale drift-aware large scale monocular SLAM, Robot.: Sci. Syst. VI 2 (3) (2010) 7.
[36] H. Strasdat, Local Accuracy and Global Consistency for Efficient Visual SLAM (Ph.D. dissertation), Department of Computing, Imperial College London, 2012.
[37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.
[38] Z. Zhang, D. Scaramuzza, A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 7244–7251.

Yi Yang received the Ph.D. degree in control science and engineering from the School of Automation, Beijing Institute of Technology, Beijing, China, in 2010.

He is currently a Professor with the School of Automation, Beijing Institute of Technology. His research interests include the area of mobile robots and unmanned ground vehicles, with focus on sensor fusion, localization and mapping, machine learning, collaborative perception, motion planning and control for autonomous navigation.

Di Tang received the B.S. degree in automation from Beijing Institute of Technology, China, in 2018. He is currently pursuing the M.S. degree in control science and engineering in the school of Automation, Beijing Institute of Technology.

His research interests include autonomous vehicle, computer vision and visual SLAM. He has taken part in ICRA DJI RoboMaster Artificial Intelligence Challenge in 2019 as the Captain of BIT.

Dongsheng Wang received the B.S. degree in automation and the M.S. degree in control science and engineering from the school of Automation, Beijing Institute of Technology, Beijing, China, in 2016 and 2019, respectively.

He is currently an algorithm engineer with China Intelligent and Connected Vehicles (Beijing) Research Institute, Beijing, China. His research interests include image stitching, machine learning and SLAM.

Wenjie Song received the B.S. degree and the Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2013 and 2019, respectively. He studied in Princeton University as a visiting scholar from 2016 to 2017.

He is currently an Assistant Professor with the School of Automation, Beijing Institute of Technology. His research interests include autonomous driving, environmental perception, SLAM and path planning.

Junbo Wang received the B.S. degree in automation from Beijing Institute of Technology, Beijing, China, in 2019. He is currently pursuing the M.S. degree in control science and engineering in the school of Automation, Beijing Institute of Technology.

His research interests include visual SLAM, machine learning and collaborative perception.

Mengyin Fu received the B.S. degree from Liaoning University, China, the M.S. degree from the Beijing Institute of Technology, China, and the Ph.D. degree from the Chinese Academy of Sciences.

He was elected as the Yangtze River Scholar Distinguished Professor in 2009. He was a recipient of the Guanghua Engineering Science and Technology Award for Youth Award in 2010 and the National Science and Technology Progress Award for several times in recent years.

Prof. Fu is the President of the Nanjing University of Science and Technology. His research interests cover integrated navigation, intelligent navigation, image processing, learning and recognition and their applications.