2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Robust and Efficient Vehicles Motion Estimation with Low-Cost Multi-Camera and Odometer-Gyroscope

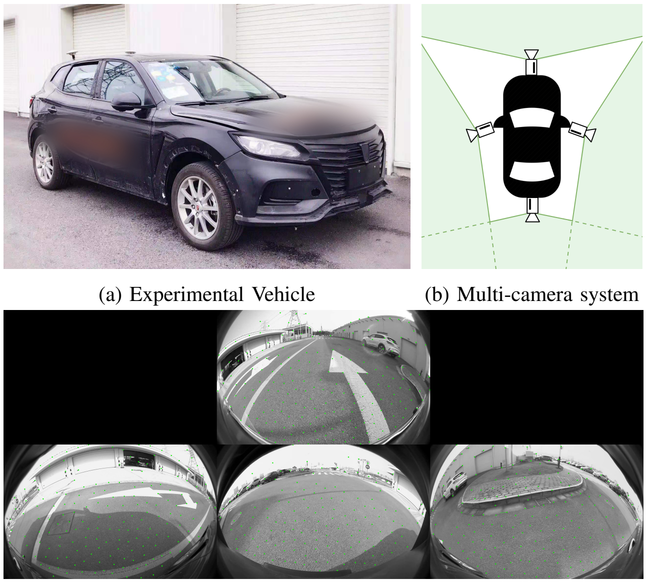
Wenlong Ye[1], Renjie Zheng[1], Fangqiang Zhang[2], Zizhou Ouyang[2] and Yong Liu[1,3]

Abstract— **In this paper, we present a robust and efficient estimation approach with multi-camera, odometer and gyroscope. Robust initialization, tightly-coupled optimization estimator and multi-camera loop-closure detection are utilized in the proposed approach. In initialization, the measurements of odometer and gyroscope are used to compute scale, and then estimate the bias of sensors. In estimator, the pre-integration of odometer and gyroscope is derived and combined with the measurements of multi-camera to estimate the motion in a tightly-coupled optimization framework. In loop-closure detection, a connection between different cameras of the vehicle can be built, which significantly improve the success rate of loop-closure detection. The proposed algorithm is validated in multiple real-world datasets collected in different places, time, weather and illumination. Experimental results show that the proposed approach can estimate the motion of vehicles robustly and efficiently.**

I. Introduction

Self-driving vehicle is a complex topic with many challenges, among which motion estimation is a crucial issue. Various sensors (e.g., camera, LiDAR, IMU and wheel odometer) are used to estimate motion. The vision-based method has become prevalent in motion estimation, as the advantage of lightweight, low cost and sufficient information. The low-cost camera configuration (see Fig. 1-(b)) is already commonly equipped in some commercial vehicles. Besides, multi-camera system covers a wider field-of-view, which can improve the performance of motion estimation, especially in poorly textured environments. Therefore, some motion estimation methods toward the multi-camera system are presented[1], [2], [3], [4]. However, vision-only approaches are usually not robust enough, which is fatal for self-driving vehicles.

To make the vision-based method more reliable in real-world applications, various sensors are introduced to improve the robustness. To this end, there is a growing trend of visual-inertial navigation system (VINS) recently. VINS, which combines visual observations from cameras and motion data from IMU to achieve 6-DOF localization, has the advantage of scale observability and being robust to fast motion. Considering the cost reduction of IMU and excellent performance of VINS in autonomous MAVs, we expect it to perform the same well on the self-driving vehicles moving on the ground. However, this is not the case. The main reason is that the movement of the vehicles is not a complete 6-DOF movement. The restricted motion that vehicles undergo on the ground is planar and has the constant velocity or acceleration in most case, which may lead to the unobservability of part of the state (e.g., metric scale)[5]. Since we are focusing on vehicles that are usually equipped with wheel odometer, we are considering the use of wheel odometer to solve the unobservability problem.



(a) Experimental Vehicle

(b) Multi-camera system

(c) Example of images from multi-camera system from outdoors

Fig. 1. The experimental vehicle equipped with multi-camera system, odometer, gyroscope and GPS/INS system. (a) The appearance of the experimental vehicle. (b) The multi-camera system configuration. (c) Example of images from multi-camera system from outdoors. The upper row is front camera, and the bottom row is left, back and right cameras.

Since the visual-only approach and VINS encounter significant challenges in the motion estimation for vehicles, a motion estimation approach is proposed, which combines images from multi-camera, wheel odometer and gyroscope in this paper. The proposed approach is validated using the experimental vehicle in Fig. 1. In summary, the main contributions of this work can be summarised as the following:

- We present a robust and efficient motion estimation method for vehicles equipped with multi-camera system, odometer and gyroscope.
- We derive the pre-integration of odometer-gyroscope and estimate the bias of sensors in a tightly-coupled sliding window optimization framework.

This work is supported by the National Natural Science Foundation of China under Grant U1509210, 61836015.

[1] Wenlong Ye and Renjie Zheng are with the Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China.

[2] Fangqiang Zhang and Zizhou Ouyang are with SAIC Motor Corporation Limited, Shanghai, 201804, China.

[3] Yong Liu is with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China (Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn).

978-1-7281-4004-9/19/$31.00 ©2019 IEEE

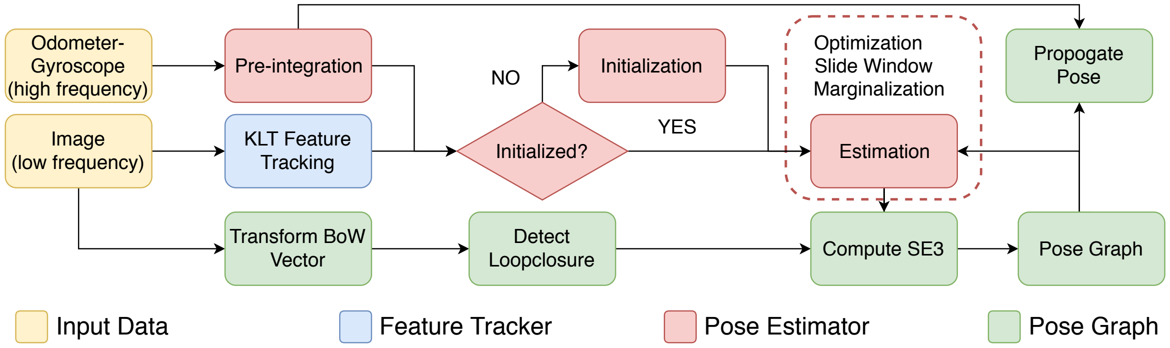


Fig. 2. System overview of the proposed motion estimation approach.

- We propose an initialization algorithm with odometer-gyroscope and multi-camera loop-closure detection method to improve the robustness and efficiency of motion estimation.
- We carry out sufficient experiments on real-world datasets and validate the performance of the proposed approach.

II. Related Work

A. Multi-Camera Based Approach

In recent years, many multi-camera-system-based robot localization algorithms have been proposed due to the emergence of various fisheye camera models. For example, the models proposed in[6], [7] show excellent performance in practice. In [8], an extrinsic calibration method that uses the specially designed pattern to extract features of varying scales is proposed. In [9][10], SLAM approach based calibration method that runs a bundle adjustment on vehicle' driving data to get refined extrinsic is proposed. Based on the above, multi-camera system is applied to the state-of-art visual SLAM methods. Urban et al.[11] and Liu et al.[2] extend ORB_SLAM2[12] and DSO[13] repeatedly to make them applicable to multi-camera system. However, these two approaches take a tremendous computational cost. The difference between the multi-camera system and pinhole camera that light rays do not meet at a single point results in that standard pose estimation solutions cannot work. To fix this problem, Pless et al.[14] propose the 17-points algorithm to compute relative pose and Li et al.[15] extend it to suit the degeneracies. Stewénius et al.[16] propose 6-points algorithm that needs fewer points correspondences, but it is difficult to identify the correct one from 64 solutions given by 6-points algorithm. The 2-points algorithm[3] with Ackermann motion model that gives up to 6 solutions enables fast localization. There are also some algorithms[17], [18] representing the light rays as Plücker lines that can solve the pose estimation efficiently without the need for SVD.

B. Sensors Fusion based Approach

Due to the poor robustness of vision-only localization approaches, motion measuring sensors(e.g., IMU and odometer) are used in visual estimation methods. There are two methods to deal with images and IMU data in VINS: loosely-coupled fusion and tightly-coupled fusion, where the latter is more effective. The tightly-coupled fusion approaches are either filter-based[19], [20] or optimization-based[21], [22], [23]. MSCKF[19], [20] update the camera poses in a window using the geometric constraints provided by the features in multiple cameras' field of view. The optimization-based approaches[21], [22], [23] usually maintain a fixed-size window of states consisting of feature and camera pose and then solve the estimation problem using nonlinear optimization. On the other hand, odometer measurements and planar motion constraints are introduced to VINS to fix the scale drift problem in the case of restricted motion in [5]. However, the implementation in [5] is not a complete motion estimation system.

III. Method

A. Overview

The proposed approach consists of feature tracker, pose estimator and pose graph module, as shown in Fig. 2. The input data consists of synchronized images acquired from the multi-camera system and motion measurements from odometer and gyroscope.

Feature Tracker: The tracking module processes each frame (four images). Extracted corner points are tracked between frames by the KLT sparse optical flow algorithm[24] and rejected by RANSAC. A unique ID is assigned to the matched point for distinguishing.

Pose Estimator: The primary functions of the estimating module are pre-integration of motion measuring sensors (odometer and gyroscope) and estimating the motion by graph optimization. Sliding window optimization is used to reduce computing complexity. Keyframes are selected by the average parallax and track quality of the feature. Keyframes remain active in sliding window optimization and are fed to the pose graph module.

Pose Graph: The pose graph module is a place recognition module. The descriptors of previous keyframes are transformed into bag-of-words vector and saved in a database that stores the description of the scene. The new frame detects the loop-closure by querying in the database. Once loop-closure is detected, 4-DOF global pose graph optimization[22] is performed to correct the drift of trajectory.

The notation used in this paper is defined. We use lower case letters for scalar variables, bold lower case for vectors and bold capital case for matrices. We use superscript to represent the coordinate system, where w represents world frame, b represents body frame that coincides odometer-gyroscope frame, c represents camera frame. Hamilton quaternions **q** represents rotation and **p** represents translation. **T** represents the transformation matrix between two frames. $\pi()$ represents and projection function of camera. $\mathbf{M}_{C_i}$ represents the estrinsic between i^{th} camera and body frame.

B. Pre-integration of Odometer-Gyroscope

In this paper, the pre-integration used in the proposed approach is derived based on the IMU pre-integration[22]. The raw gyroscope measurements $\hat{\boldsymbol{\omega}}$ is given by:

$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{w_t} + \mathbf{n}_w \tag{1}$$

The wheel odometers measure the linear and angular velocity. Considering the scale error due to the measurement error of wheel diameter, the odometer measurements are modelled as:

$$\hat{\boldsymbol{v}}_t = \boldsymbol{v}_t + \mathbf{b}_{v_t} + \mathbf{n}_v \tag{2}$$

$\hat{\boldsymbol{\omega}}_t$ and $\hat{\boldsymbol{v}}_t$ are the measurements of rotational velocity and linear velocity, $\boldsymbol{\omega}_t$ and $\boldsymbol{v}_t$ are the real value, $\mathbf{b}_{w_t}$ and $\mathbf{b}_{v_t}$ are random walk bias whose derivatives are Gaussian, $\mathbf{n}_w$ and $\mathbf{n}_v$ are additive noise that is Gaussian. Although odometers can only measure the velocity in *x* axis, the measurements are expanded to three dimensions for maintaining the uniformity with other measurements and introducing the constraint that vehicles move on the ground.

The state of the vehicle that consists of position and orientation is propagated by the motion measurements during time interval $[t_k, t_{k+1}]$. To avoid the re-propagation that takes the expensive cost, pre-integration in body frame is utilized as follows:

$$\begin{aligned} \mathbf{q}_w^{b_k}\mathbf{p}_{b_{k+1}}^w &= \mathbf{q}_w^{b_k}\mathbf{p}_{b_k}^w + \mathbf{p}_{b_{k+1}}^{b_k} \\ \mathbf{q}_w^{b_k} \otimes \mathbf{q}_{b_{k+1}}^w &= \mathbf{q}_{b_{k+1}}^{b_k}, \end{aligned} \tag{3}$$

As the measurements in practice are discrete, the pre-integration and discrete-time error state dynamics used in the proposed system is derived. Mid-point integration is used for deriving here. The mean of $\mathbf{p}_{b_{k+1}}^{b_k}$ and $\mathbf{q}_{b_{k+1}}^{b_k}$ can be propagated step by step as:

$$\begin{aligned} \hat{\mathbf{q}}_{i+1} &= \hat{\mathbf{q}}_i \otimes \begin{bmatrix} 1 \\ 0.5\boldsymbol{\omega}_i^{'} \end{bmatrix} \\ \hat{\mathbf{p}}_{i+1} &= \hat{\mathbf{p}}_i + \mathbf{v}_i^{'}dt \end{aligned} \tag{4}$$

where

$$\begin{aligned} \boldsymbol{\omega}_i^{'} &= \frac{\hat{\boldsymbol{\omega}}_{i+1} + \hat{\boldsymbol{\omega}}_i}{2} - \mathbf{b}_{\omega_i} \\ \mathbf{v}_i^{'} &= \frac{\hat{\mathbf{q}}_k(\hat{\mathbf{v}}_i - \mathbf{b}_{v_i}) + \hat{\mathbf{q}}_{k+1}(\hat{\mathbf{v}}_{i+1} - \mathbf{b}_{v_i})}{2} \end{aligned} \tag{5}$$

Next, the discrete-time linearized dynamics of error state can be derived as:

$$\begin{bmatrix} \delta\mathbf{p}_{k+1} \\ \delta\boldsymbol{\theta}_{k+1} \\ \delta\mathbf{b}_{v_{k+1}} \\ \delta\mathbf{b}_{\omega_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{f}_{01} & -\frac{\mathbf{q}_k+\mathbf{q}_{k+1}}{2}dt & \mathbf{f}_{03} \\ 0 & \mathbf{f}_{11} & 0 & -dt \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\mathbf{p}_k \\ \delta\boldsymbol{\theta}_k \\ \delta\mathbf{b}_{v_k} \\ \delta\mathbf{b}_{\omega_k} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\mathbf{q}_k dt & \mathbf{g}_{01} & \frac{1}{2}\mathbf{q}_{k+1}dt & \mathbf{g}_{03} & 0 & 0 \\ 0 & \frac{1}{2}dt & 0 & \frac{1}{2}dt & 0 & 0 \\ 0 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & dt \end{bmatrix} \begin{bmatrix} \mathbf{n}_{v_k} \\ \mathbf{n}_{\omega_k} \\ \mathbf{n}_{v_{k+1}} \\ \mathbf{n}_{\omega_{k+1}} \\ \mathbf{n}_{bv} \\ \mathbf{n}_{b\omega} \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} \mathbf{f}_{01} =& -\frac{1}{2}\mathbf{q}_k\lfloor\mathbf{v}_k - \mathbf{b}_{v_k}\rfloor_\times dt - \frac{1}{2}\mathbf{q}_{k+1}\lfloor\mathbf{v}_{k+1} - \mathbf{b}_{v_k}\rfloor_\times \\ & (\mathbf{I} - \lfloor\frac{\hat{\boldsymbol{\omega}}_{k+1} + \hat{\boldsymbol{\omega}}_k}{2} - \mathbf{b}_{\omega_\mathbf{k}}\rfloor_\times dt)dt \\ \mathbf{f}_{03} =& -\frac{1}{2}(-\mathbf{q}_{k+1}\lfloor\mathbf{v}_{k+1} - \mathbf{b}_{v_k}\rfloor_\times dt)dt \\ \mathbf{f}_{11} =& \mathbf{I} - \lfloor\frac{\hat{\boldsymbol{\omega}}_{k+1} + \hat{\boldsymbol{\omega}}_k}{2} - \mathbf{b}_{\omega_\mathbf{k}}\rfloor_\times dt \\ \mathbf{g}_{01} =& \mathbf{g}_{03} = \frac{1}{4}(-\mathbf{q}_{k+1}\lfloor\mathbf{v}_{k+1} - \mathbf{b}_{v_k}\rfloor_\times dt^2)dt \end{aligned} \tag{7}$$

The Eq.6 can be abbreviated as:

$$\delta\mathbf{z}_{k+1} = \mathbf{F}\delta\mathbf{z}_k + \mathbf{G}\mathbf{n} \tag{8}$$

During the pre-integration, the Jacobian $\mathbf{J}_{k+1}$ and covariance $\mathbf{P}_{k+1}$ can be propagated by:

$$\begin{aligned} \mathbf{J}_{k+1} &= \mathbf{F}\mathbf{J}_k \\ \mathbf{P}_{k+1} &= \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \end{aligned} \tag{9}$$

When the estimation of bias changes during optimization, the pre-integration can be corrected by:

$$\begin{aligned} \mathbf{p}_{b_{k+1}}^{b_k} &= \hat{\mathbf{p}}_{b_{k+1}}^{b_k} + \mathbf{J}_{b_v}^p\delta\mathbf{b}_{\mathbf{v_k}} + \mathbf{J}_{b_\omega}^p\delta\mathbf{b}_{\omega_\mathbf{k}} \\ \mathbf{q}_{b_{k+1}}^{b_k} &= \hat{\mathbf{q}}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}_{b_\omega}^q\delta\mathbf{b}_{\omega_k} \end{bmatrix} \end{aligned} \tag{10}$$

where $\mathbf{J}_{b_v}^p$,$\mathbf{J}_{b_\omega}^p$ and $\mathbf{J}_{b_\omega}^q$ are sub-blocks of $\mathbf{J}_{k+1}$.

C. Robust Initialization

Due to the difficulty in obtaining accurate metric scale directly, VINS has a complicated initialization procedure that is difficult to succeed, which often fails VINS. It is unacceptable for the case of self-driving cars that has high requirements for robustness. To this end, a simple but robust initialization is adopted in the proposed system. The motion of vehicles can be estimated directly by integrating the odometer and gyroscope based on the assumption that the bias is zero. The poses are used to triangulate the feature points. When the measurements reach the size of the window, the optimization (in Sec. III-D) that considers the visual factor and odometer-gyroscope factor is performed. At this point, the refined pose and bias are estimated. The initialization has completed. The odometer-aided initialization method shows robust performance in the experiment(see Sec. IV-B).

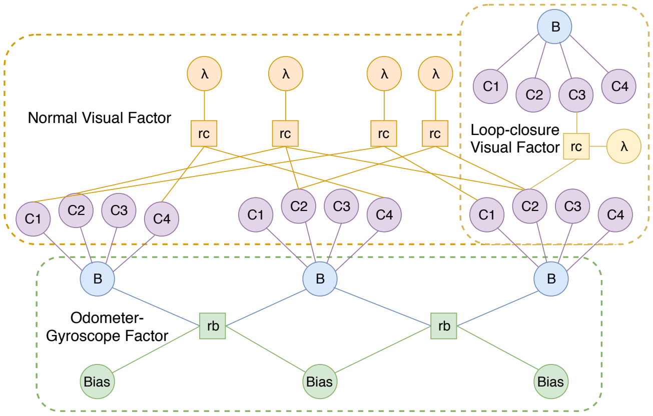


Fig. 3. Factor graph of optimization. Bias is bias of odometer-gyroscope. B is pose in body frame. C_n is pose of the n^th camera of multi-camera system. λ is inverse depth of feature in the host frame. rb connects two adjacent frames. rc connects two frames observing the same feature. In normal visual factor, rc only connect the same camera (e.g. C2 to C2). In loop-closure visual factor, rc may connect two different cameras (e.g. C2 to C3).

D. Graph Optimization

Consider the frame state vector consisting of odometer-gyroscope state and inverse depth of feature:

$$\mathbf{x}_k = [\mathbf{p}_k, \mathbf{q}_k, \mathbf{b}_v, \mathbf{b}_\omega, \lambda_1, \lambda_2, \cdots, \lambda_m] \tag{11}$$

Ceres Solver [25] is used to solve the optimization problem:

$$\mathbf{x} = argmin\{\sum \mathbf{r}_V + \sum \mathbf{r}_B + \mathbf{r}_M\} \tag{12}$$

where $\mathbf{r}_V$ is visual factor, $\mathbf{r}_B$ is odometer-gyroscope factor, $\mathbf{r}_M$ is marginalization factor.

The structure of the factor graph is shown in Fig. 3.

*1) **Visual Factor**:* Pinhole camera is modelled as generalized reprojection plane and pinhole. This model is not suitable for fisheye cameras. In [6], [7], fisheye camera is modelled as reprojection unit sphere. Therefore, the reprojection error computed on unit sphere in the proposed system. Considering the multi-camera model in the proposed system, four images can be regarded as a frame and calculate the reprojection error in a framework. Therefore, measurements from four cameras can be combined to optimize the motion of vehicle in body frame.

$$\begin{aligned}\mathbf{r}_V(\mathbf{p}_i, \mathbf{p}_j) &= \\ [\mathbf{b}_1, \mathbf{b}_2]^T(\pi^{-1}\mathbf{p}_i &- \frac{\mathbf{M}_{C_{pi}}^{-1}\mathbf{T}_{ij}^C\mathbf{M}_{C_{pj}}(\pi^{-1}\mathbf{p}_j)/\lambda_{p_j}}{\parallel \mathbf{M}_{C_{pi}}^{-1}\mathbf{T}_{ij}^C\mathbf{M}_{C_{pj}}(\pi^{-1}\mathbf{p}_j)/\lambda_{p_j} \parallel})\end{aligned} \tag{13}$$

where $\mathbf{M}_{C_p}$ is the extrinsic of the camera which observes $\mathbf{p}$ to body frame, $\mathbf{p}_i$ and $\mathbf{p}_j$ are pixel coordinates of the matched features in frame i and j, $\mathbf{b}_1$ and $\mathbf{b}_2$ are two orthogonal bases of the tangent plane of reprojection unit sphere.

*2) **Odometer-Gyroscope Factor**:* The state to be optimized of odometer-gyroscope consists of $\mathbf{p}$, $\boldsymbol{\theta}$, $\mathbf{b_v}$ and $\mathbf{b}_\omega$, therefore the residual factor of odometer-gyroscope can be defined as:

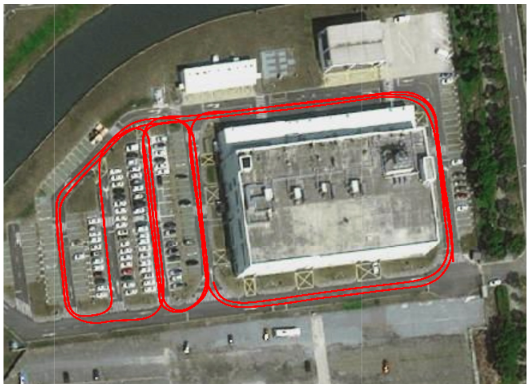


Fig. 4. A bird's-eye view of the outdoor scene

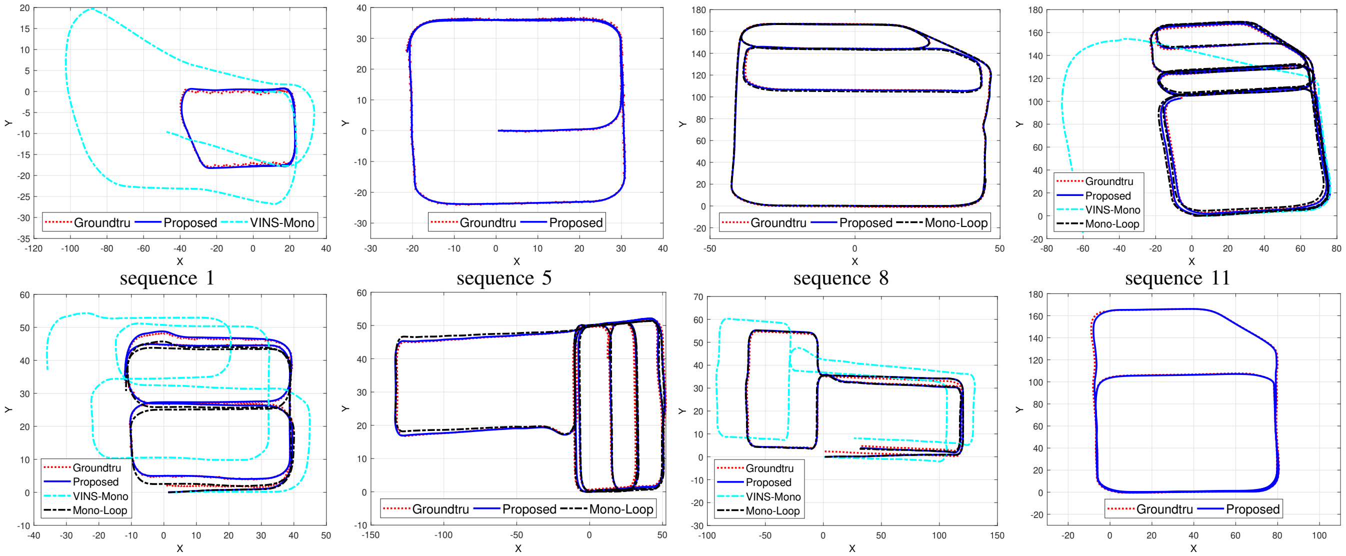
$$\mathbf{r}_B(\mathbf{x}_{k+1}^k) = \begin{bmatrix} \delta\mathbf{p}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\theta}_{b_{k+1}}^{b_k} \\ \delta\mathbf{b}_v \\ \delta\mathbf{b}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{q}_w^{b_k}(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w) - \hat{\mathbf{p}}_{b_{k+1}}^{b_k} \\ 2\mathbf{q}_w^{b_k} \otimes \mathbf{q}_{b_{k+1}}^w \otimes (\hat{\mathbf{q}}_{b_k}^{b_{k+1}}) \\ \mathbf{b}_{v_{k+1}} - \mathbf{b}_{v_k} \\ \mathbf{b}_{\omega_{k+1}} - \mathbf{b}_{\omega_k} \end{bmatrix} \tag{14}$$

As the state θ is three dimensional but $\mathbf{q}$ is four-dimensional, the imaginary part of the quaternion is extracted as error state.

*3) **Marginalization Factor**:* To maintain the fixed size of the sliding window, the old state should be thrown away. Marginalization is adopted to avoid destroying the constraints established by the previous measurements. When the frameset reaches the size of the window, marginalization is carried out at the time processing the next frame. There are two marginalization criteria according to whether this frame is a keyframe. When this frame is keyframe, all the factors of the oldest frame will be marginalized out. Otherwise, the visual factors of this frame will be marginalized out, and odometer-gyroscope factors will be kept to maintain the continuity of odometer-gyroscope measurements between frames.

E. Multi-Camera Loop-Closure

Multi-camera system is beneficial for loop-closure detection. The image captured by each camera is regarded as a separate camera frame. Corner points in each image are described by the BRIEF descriptor[26]. In addition to the points extracted in feature tracker, more keypoints are extracted here. DBoW2[27] is used to transform descriptors to bag-of-words vectors and store these vector in keyframes database. New keyframe detects loop-closure by querying the bag-of-word vector keyframes database. The relative pose between the current camera frame and loop-closure camera frame can be computed by EPnP[28]. Then the relative pose to loop-closure frame in body frame can be computed since the extrinsic is known. In particular, one keyframe containing four images can only detect one loop-closure in the proposed system. Once loop-closure is detected, 4-DOF global pose graph optimization[22] is performed to correct the drift of trajectory.



sequence 1 sequence 5 sequence 8 sequence 11

sequence 12 sequence 13 sequence 15 sequence on PX2

Fig. 5. The part of trajectories of the proposed system and groundtruth. Grounttru is the ground truth, Proposed is the proposed approach(full system), Mono-Loop is the proposed approach(with monocular camera loopclosure).

IV. Exprimental Results

Experiments are carried out in the park of SAIC Motor Corporation Limited. The vehicle equipped with multi-camera system, wheel odometer and a low-cost IMU (only gyroscope is used) is the experimental platform. The model in [6] is used to model and calibrate the intrinsic of the multi-camera system. To attain the extrinsic, the experimental vehicle is parked in an empty room with a big chessboard calibration pattern on the ground. Initial extrinsic is estimated by optimizing the reprojection error of chessboard corners Then the initial result is fed to CamOdoCal[9]. After the offline bundle adjustment on long-term outdoor travel data, refined extrinsic of multi-camera system and rigid transform between cameras and odometer-gyroscope are calibrated.

The proposed algorithm is implemented based on VINS-Mono[22]. The feature tracker and pose graph module of VINS-Mono are extended for multi-camera system. The estimation module pre-integrates the measurements of odometer-gyroscope and propagates the error. The nonlinear optimization problem (in Sec. III-D) is solved to estimate the motion of vehicle and bias of sensors. Four images with the close timestamp are regarded as one frame in the estimation module and as four frames in the pose graph module. 150 corners are extracted for each image in feature tracker. Images are resized to 960*604. All the parameter settings remain the same in all the tests.

Experiments run on an Intel Core i7-7700k desktop computer with 32GB RAM and Nvidia DRIVE PX2. GPU is not included in both platforms. In particular, only one Tegra in DRIVE PX2(consisting of Tegra A and Tegra B) is used. The performance metrics include the absolute trajectory error (ATE) to evaluate accuracy and CPU runtime to evaluate the computational cost. The ground truth of trajectory is provided by GPS/INS system on the vehicle. The driving data is collected in different places (outdoor and underground parking lot), time weather and illumination at the speed of fewer than 20 kilometres per hour. Fig. 4 shows a bird's-eye view of the outdoor scene, which is empty and lacks objects and texture information. This will bring challenges to the motion estimation approach. Fig. 1-(c) shows an example of images from multi-camera system in the outdoor scene. Accuracy, initializing robustness, loop-closure efficiency and computational cost experiments are carried out to validate the performance of the proposed approach.

TABLE I
Motion Estimation Accuracy

Sequence	Length(m)	Scene	Proposed(Full System)	Proposed(Mono-Loop)
1	208.0	outdoor	0.6417	/
2	142.5	outdoor	0.3042	/
3	64.8	outdoor	0.3693	/
4	141.5	outdoor	0.3503	/
5	323.7	underground	0.2487	/
6	329.5	underground	0.3310	/
7	737.1	outdoor	0.2114	/
8	911.5	outdoor	0.4758	0.7298
9	143.8	outdoor	0.2349	/
10	116.8	outdoor	0.3584	/
11	1671.2	outdoor	0.6553	1.0742
12	395.0	underground	0.6326	0.6260
13	982.1	underground	0.9897	1.0561
14	600.4	underground	0.4878	0.7742
15	713.7	underground	0.8870	0.9398

A. Accuracy Experiment

In this experiment, we focus on the motion estimation accuracy of algorithms. The offline experiment is carried out on the desktop computer. Our results are compared to VINS-Mono[22]. It should be noted that although this is not a fair

TABLE II
INITIALIZATION FRAMES REQUIRED

Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Proposed	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
VINS-Mono	53	/	22	/	89	/	84	43	36	26	61	20	28	28	32

comparison due to the difference in sensors configuration, it makes sense to validate the performance of the proposed system. The part of trajectories of the proposed system and ground truth is shown in Fig. 5. In Table. I, the length and collected scene of each sequence are shown. In the column Full System, the metric, ATE, is used to evaluate the accuracy of the proposed algorithm. The proposed algorithm attains excellent performance. VINS-Mono[22] can only work on sequences 1,9,10,11,12,17 (fail after running for a while on sequence 11), where the scale shows a significant drift. VINS-Mono[22] often fails when the vehicle is turning fast in our dataset. The proposed approach can perform accurate motion estimation and estimate true scale compared to VINS.

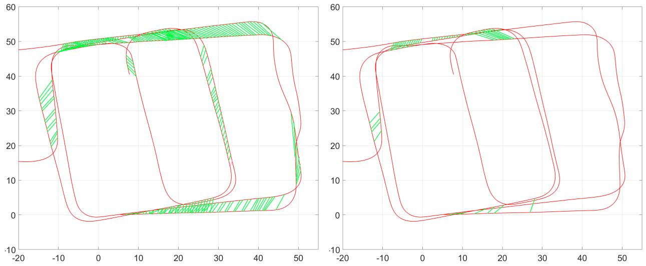
The online experiment is carried out on NVIDIA DRIVE PX2 on vehicle. As the sensors driver takes up much computing resource, this experiment can be regarded as the experiment on low computing capacity platform. The trajectories of the proposed system and ground truth is shown in Fig. 5. In this case, the proposed system also has excellent performance. The vehicle passes through the bottom part of trajectory more than one time. The total length of the trajectory is 936.7m. The ATE of the trajectory is 0.55m. The proposed approach also has a excellent performance on NVIDIA DRIVE PX2 platform.

B. *Initialization Experiment*

The initialization method in Sec. III-C has the advantage of robustness and a high rate of success. In this experiment, only the performance of initialization is evaluated. To this end, the size of the initializing window is set to 10 for both the proposed system and VINS-Mono[22]. The number of frames required for initialization is counted. Table. II shows the results of initialization. Each sequence is run 10 times, and the result of each run is very close. In the frames count column, the mode in 10 runs is shown. No data means the failure of initialization. From the table, it can be seen that the proposed method has a more robust and efficient initialization.

C. *Loop-Closure Expriment*

There is the case where the vehicle passes through a road from different directions in sequence 8,11,12,13,14,15. The loop-closure experiment is carried out on these sequences. The approach containing feature tracker, pose estimator and monocular loop-closure is compared to the proposed full system. The result of the approach with monocular loop-closure are shown in Fig. 5 and Table. I. The comparison results illustrate the improvement in motion estimation accuracy of multi-camera loop-closure. In particular, the loop-closures detected by two methods in a partial trajectory of sequence 13 is shown in Fig. 6. Some extra rotational error is added to the trajectory in Fig. 6 to make the detected loop-closure more clear in the figure. Much more loop-closure is detected when using multi-camera loop-closure module, which can bring better localization performance. In practice, it is quite common for vehicles to pass through a road from different directions. The proposed approach can work well in this case, which improves the robustness of self-driving vehicles.



(a) Multi-camera loop-closure (a) Mono-camera loop-closure

Fig. 6. Trajectory (red) and detected loop-closures (green). (a) 591 loop-closure edges. (b) 87 loop-closure edge.

D. *Computational Cost*

Lastly, the efficiency of the proposed approach is shown by counting the CPU runtime spent on the main procedures of the proposed approach on desktop computer. Sequence 13 that is the second-longest and has the most loop-closures is selected as the test sequence. The result is shown in Table. III. In this table, corner detection, KLT tracking and keyframe process are the time spent on one image in a frame. Optimization consists of the time in building and solving the optimization. Since loop detection and pose graph optimization are sensitive to the length of sequence, the maximum time in parentheses besides the average value is given. According to the result in Table. III, the proposed can run in real time. The efficiency of the proposed approach is validated.

TABLE III
TIME CONSUMING

Module	Procession	Average Time (ms)
Feature Tracker	Corner Detectction	5.6
	KLT Tracking	1.2
Pose Estimator	Optimization	15.3
Pose Graph	Keyframe Process	4.7
	Loop Detection	26.6 (max:86.6)
	Pose Graph Optimization	74.3 (max:985)

V. Conclusions

To improve the robustness and efficiency of motion estimation of self-driving vehicle, a complete system that combines the measurements of multi-camera and odometer-gyroscope is presented. In the future, the proposed system will be extended to localize vehicles globally and build a more detailed map by the connection between multi-cameras.

References

[1] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Mühlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li, *et al.*, "Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 809–816.

[2] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Towards robust visual odometry with a multi-camera system," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1154–1161.

[3] G. Hee Lee, F. Faundorfer, and 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.

[4] L. Liu, H. Li, Y. Dai, and Q. Pan, "Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2432–2444, 2018.

[5] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5155–5162.

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5695–5701.

[7] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3945–3950.

[8] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1301–1307.

[9] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1793–1800.

[10] G. Carrera, A. Angeli, and A. J. Davison, "Slam-based automatic extrinsic calibration of a multi-camera rig," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2652–2659.

[11] S. Urban and S. Hinz, "Multicol-slam-a modular real-time multi-camera slam system," *arXiv preprint arXiv:1610.07336*, 2016.

[12] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[14] 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.

[15] H. Li, R. Hartley, and J.-h. Kim, "A linear approach to motion estimation using generalized camera models," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[16] M. Henrik Stewénius, K. Aström, and D. Nistér, "Solutions to minimal generalized relative pose problems," 2005.

[17] G. H. Lee, B. Li, M. Pollefeys, and F. Fraundorfer, "Minimal solutions for the multi-camera pose estimation problem," *The international journal of robotics research*, vol. 34, no. 7, pp. 837–848, 2015.

[18] G. Hee Lee, M. Pollefeys, and F. Fraundorfer, "Relative pose estimation for a multi-camera system with known vertical direction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 540–547.

[19] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[20] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[21] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[22] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[23] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[24] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[25] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[26] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[27] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[28] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.