2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

Vehicle Model Based Visual-Tag Monocular ORB-SLAM

Wenhao Zong*, Longquan Chen, Changzhu Zhang, Zhuping Wang and Qijun Chen†, *Senior Member, IEEE*
School of Electronics and Information Engineering
Tongji University, Shanghai, China
* Email: 1310480@tongji.edu.cn
† Email: qjchen@tongji.edu.cn

***Abstract*—Monocular ORB-SLAM has been proved to be one of the best open-source SLAM method. However, it is still unsatisfying especially in low illumination indoor environment, which is caused by scale recovery and wrong feature matching. In this paper, we proposed a vehicle model based monocular ORB-SLAM method supplemented by April-Tag to improve the performance of original algorithm. This approach is practical when autonomous driving in low-light and less-feature environment like garages and tunnels. We achieve this by proposing a vehicle model based initialization method fusing April-Tag measurement to recover scale. During tracking procedure, the outliers ORB feature points will be removed by checking reprojection error calculated from April-Tag. In addition, considering vehicle model can only obtain 2D motion, the vertical transition is estimated from camera model. Afterwards, a local Bundle Adjustment(BA) is applied to optimize camera pose both from frame to frame and frame to keyframe which will reduce accumulative error of the vehicle model. Finally, a convincing result is obtained from the testing drive in a garage.**

I. Introduction

In the recent years, many methods have been developed for vehicle navigation. One of the most common method is to use Global Positioning System (GPS), which can provide meter-level accuracy. With Real-Time Kinematic (RTK) [1], GPS localization can provide centimeter-level accuracy. However, the signal from satellites only can be used in outdoor environment and the devices are usually expensive. Simultaneous Localization and Mapping (SLAM) [2][3] can build a map of an unknown environment, no matter indoor or outdoor, and localize in the map with a strong focus.

In contrast to expensive and most studied LiDAR SLAM [4][5][6], visual SLAM research is acknowledged as one of the hottest research topic these years. On the one hand, visual sensor are more intuitive than LiDAR, and has rich information. On the other hand, with the development of CPU and GPU, many complex algorithms can achieve real-time effects. Being different from LiDAR SLAM, visual SLAM uses Bundle Adjustment (BA) to build the map and optimize the camera pose. BA is known to provide accurate estimation of camera localization as well as a sparse geometrical reconstruction [7][8], under the estimation that a strong network of matches and good initial guesses are provided. Visual SLAM can be performed by using just a monocular camera, and the first real-time application of monocular SLAM in the applied BA was the work of Klein and Murray [9], known as parallel tracking and mapping (PTAM) which provided effective methods for multi-thread processing, feature point matching, keyframe choosing and frame tracking. Later, these methods became the standard procedure for visual SLAM. Unfortunately, as lack of loop closing [10][11] and low invariance to viewpoint of the relocation, PTAM is limited to small-scale operation. Also, There are some methods based on direct methods, such as dense tracking and mapping in real-time (DTAM) [12], large-scale direct monocular SLAM (LSD-SLAM) [13], etc. They are able to perform dense or semi-dense reconstructions of the environment. The advantage of these methods is that algorithms use the information of every pixel with no wasting of time extracting feature. However, direct methods have their own limitations. Assuming uniform illumination environment limits its application and the photometric consistency limits its baseline of the matches. ORB-SLAM [14] is recognized as the best feature based SLAM method, but it is still unsatisfying in less-feature environment, which is caused by wrong feature

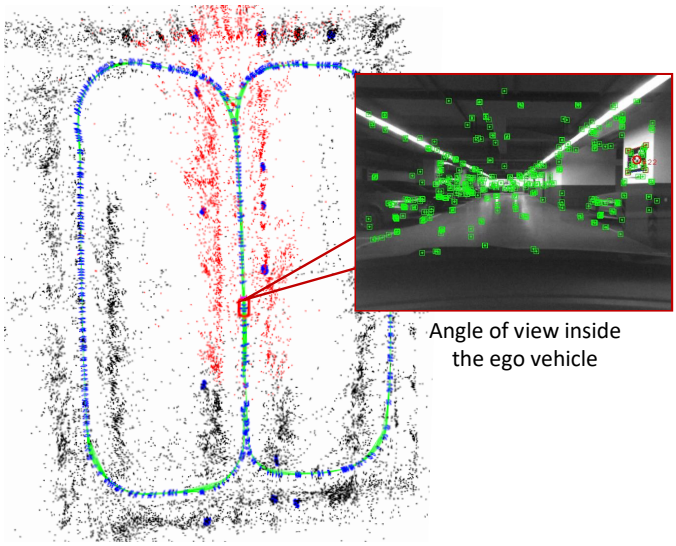

Fig. 1. Estimated map and keyframes by our system in a low-light and less-feature garage. The green points in Angle of view inside the ego vehicle are ORB feature points, The red points and number are April-Tag points and its ID. The keyframes(blue) and current vehicle(red) that share more than 100 points(black,red) observations, the red points means current local map points, black points means global map points, are connected by green lines.

978-1-5386-1645-1/17/$31.00 ©2017 IEEE

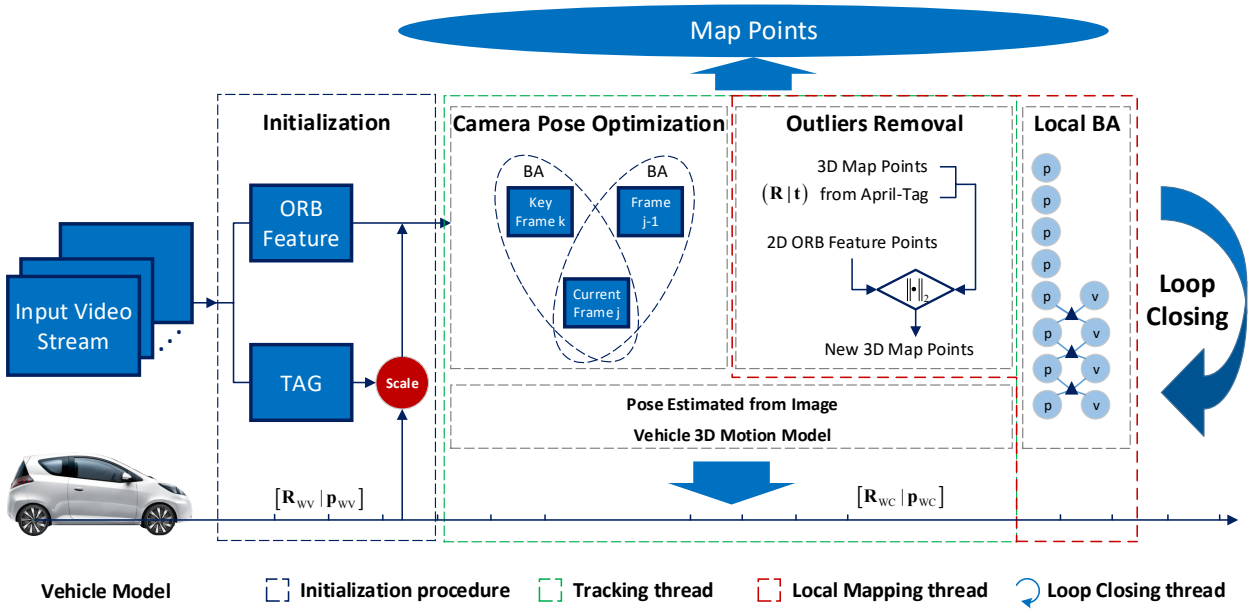

Fig. 2. System architecture of vehicle model based visual-tag monocular SLAM. This diagram is under the basic framework of ORB-SLAM which contains initialization, tracking thread, local mapping thread and loop close thread.

matching. Although using monocular camera is the cheapest and smallest sensor setup compared with stereo camera and RGB-D camera, all of these methods mentioned above can not solve a fatal problem of monocular camera SLAM which is no absolute depth and the unobservation of the real scale of the map and estimated trajectory.

My-Ha Le, Van-Dung Hoang [15] proposes a method for localization of vehicle using one point plus an edge matching region of monocular vision based in vehicle model and GPS, but it can not work well in garage where the GPS signal is weak. D ScaramuzzaF Fraundorfer [16] estimate the absolute scale scale and remove outliers with a restrictive motion model, but it limits to 2D environment. The above two have a problem that lack the capability to close loops, unlike them, R MurartalJD Tardos [17] present a novel tightly-coupled Visual-Inertial Simultaneous Localization and Mapping system that is able to close loops and reuse its map to achieve zero-drift localization in already mapped areas, but it need a high requirement for IMU performance and is hard to initialize successfully. In this paper, to solve the scale ambiguity and scale drift, we introduce a vehicle model and artificial visual landmarks, also known as ”fiducials”, which is the part of a map. Vehicle model can be used to solve scale ambiguity when the system is trying to initialize, artificial landmarks can provide much more accurate information than ORB feature. We choose April-Tag [18] as our fiducial, which is an open source robust and lightweight visual fiducial system. It is designed for recognizing artificial landmarks, and each landmark has a full 6-DOF pose. The visual fiducial can be detected and localized even if the original image has a very low resolution, the environment is at non-uniform illumination or the tag is oddly rotated and large area occluded and tucked away in a corner with a small payload. In brief, it has a strong robustness to false positives arising from natural imagery and significantly higher localization accuracy. In short, our system get a better performance in low illumination indoor environment, such as garages and tunnels, by tightly fusing vehicle measurement and loosely fusing observations of April-Tag with ORB-SLAM. The contributions are given as follows.

- The system architecture of vehicle model based visual-tag monocular ORB-SLAM is introduced in Section II;
- A vehicle model and April-Tag based initialization method is produced to recover scale in the procedure of initialization in Section III;
- An algorithm that utilizing April-Tag with convincing depth to eliminate outliers of ORB feature through re-projection is proposed in Section IV(A);
- In this paper, a method to estimate vertical motion of the vehicle is proposed to remove the assumption that vehicle is moving on a 2D plain. In addition, during tracking and mapping procedure, a vehicle model based BA optimization method is introduced in Section IV(B).

In the following sections, vectors are indicated with small, bold letters (e.g. $\mathbf{v}$)).Scalars are indicated with small letters (e.g. f_u).Matrices are indicated by non-bold captital letters (e.g. R). In this paper, we consider a conventional pinhole-camera model with a projection function π, which depends on the camera intrinsics. The focal lengths, f_u and f_v, and the cameras principal point c_u and c_v. Function π transforms 3D points in camera coordinate, into 2D points on the image coordinates. We define vehicle coordinate system V, the world coordinate system W and the camera coordinate system C.

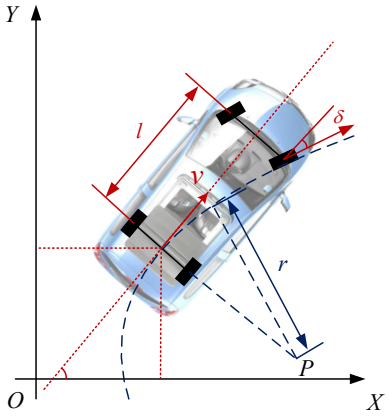

Fig. 3. Ackermann vehicle model. l is the vehicle wheel base. v is velocity of rear axle center. δ is the tire angel. P is the instantaneous center of rotation.

II. System Overview

In this section, the main architecture of vehicle model based visual-tag SLAM will be given and demonstrated in Fig. 2. Since our system is based on ORB-SLAM, three main threads including Tracking, Local Mapping and Loop Closing are preserved with some modification. In the procedure of system initialization, ORB feature, vehicle model and April-Tag will be taken into account. Both vehicle model and April-Tag can obtain absolute scale, so their weighted mean is taken as the final scale. In the Tracking thread, camera pose as well as map points will be optimized in terms of vehicle model by Bundle Adjustment both from frame to last frame and frame to keyframe. In addition, the corner points of the April-Tag is treated as the feature points similar to ORB feature points. The descriptor is set up to ensure the uniqueness of every tag corner point. Meanwhile, vertical camera pose is used to estimate vertical motion of the vehicle in order to convert 2D vehicle model to 3D. In our system outliers are removed not only by $(\mathbf{R}|\mathbf{t})$ calculated by BA, but also the one obtained from April-Tag. With the optimization work done for April-Tag pose estimation, it is considered as a strong prior within certain distance. In the Local Mapping thread, after a new keyframe is inserted, a local BA is executed to optimize the last N keyframes constrained by the vehicle model. In the loop closing thread, April-Tag is used to determine the loop closing and $(\mathbf{R}|\mathbf{t})$ between current frame and previous frame described by Bag of words serves as the input of overall BA of the whole map. The detailed algorithms and mathematical expression will be given in the following sections.

III. System Initialization

A. Vehicle Model

In this subsection, we introduce a vehicle model, which measures the vehicle speed v and steering wheel angel δ with fixed intervals Δt, typically at 100 hertz. It assumes Gaussian noise q for vehicle speed and steering wheel angle measurement. This can be formulated by Equation 1:

$$v = \tilde{v} + q_v, \delta = \tilde{\delta} + q_\delta \tag{1}$$

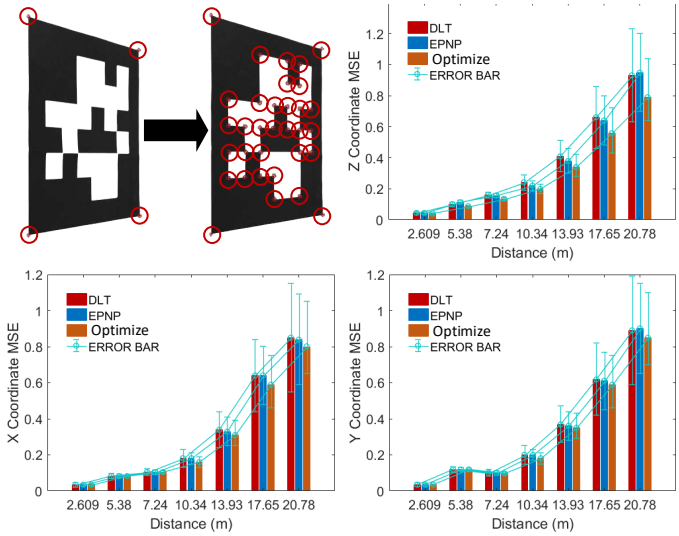

Fig. 4. April-Tag pose calculation with different PnP methods. Top left: April-Tag corner points extraction before and after optimization. Top right: mean square error and error bar of direction Z with three different PnP methods. Bottom left: mean square error and error bar of direction X with three different PnP methods. Bottom right: mean square error and error bar of direction Y with three different PnP methods.

Since vehicle always move a short distance when system is trying to initialize, we assume the motion of vehicle in initialization procedure is planar. Based on Ackermann steering geometry, the center P of circular is known as Instantaneous Center of Rotation. This can be formulated in Equation 2 and shown in Fig. 3

$$\begin{cases} \delta = \theta_{\text{steer}}/\tau \\ r = \frac{l}{\tan\delta} \\ \omega_{\text{v}} = v/r \\ v_X = r\left(1-\cos\left(\omega_{\text{v}}\right)\right) \\ v_Y = r\sin\left(\omega_{\text{v}}\right) \end{cases}, \tag{2}$$

where θ_{steer} is the steering angle measurement. τ is transmission ratio from steering wheel to tire angle whose value is usually around 16. r is instantaneous radius of rotation. v is the velocity of rear axle center. ω_{v} is instantaneous vehicle angle speed. v_X is vehicle lateral speed. v_Y is vehicle longitudinal speed.

B. Optimization for April-Tag Pose Estimation

April-Tag estimates each pose of tag by using Direct Linear Transform (DLT) algorithm [8] to compute homography, which can be written as follow.

$$\begin{aligned} &\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = sPE \\ &= s\begin{pmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} R_{11} & R_{12} & p_x \\ R_{21} & R_{22} & p_y \\ R_{31} & R_{32} & p_z \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{3}$$

where the rotation components R_{ij}, translation components p_k and scale s can be easily solved.

However, this method is only based on 4 corners of a tag, which is sensitive to tiny deformation and image quality. In order to enhance the constraint, we choose to extract more corners from a tag orderly based on Features From Accelerated Segment Test (FAST) [19]. For the reason that each tag's real size is a prior, the center of the tag is chosen as the origin of the 3D tag coordinate system. As each corner's 3D coordinate and 2D image plane coordinate are obtained, tags pose can be estimated by solving a Perspective-n-Point (PnP) problem. Some typical algorithms to solve the PnP problem are reviewed in the following part and one of those will be selected to optimize the pose estimation through an experiment.

With Efficient PnP (EPnP) [20], the i^{th} reference point in the world frame p_i^w, and its corresponding image point p_i^c, is weighted sum of four control points denoted by $c_j^w$and c_j^c respectively, and the weights α_{ij} are normalized per reference point.

$$\begin{cases} p_i^w = \sum_{j=1}^{4} \alpha_{ij} c_j^w \\ p_i^c = \sum_{j=1}^{4} \alpha_{ij} c_j^c \\ \sum_{j=1}^{4} \alpha_{ij} = 1 \end{cases} \tag{4}$$

From Equation 4, the derivation of the image reference points becomes

$$s_i p_i^c = K \sum_{j=1}^{4} \alpha_{ij} c_j^c \tag{5}$$

where K is camera projection matrix. Once the four control points c_j^c have been solved, the rotation and translation matrices that minimize the reprojection error of the world reference points are calculated.

With iterative method, the sum of squared distances between the observed projections 2D imagePoints and the projected 3D objectPoints is minimized based on Levenberg-Marquardt optimization, which is defined in Equation 7.

$$arg\min_{\mathbf{R},\mathbf{p}} \sum_{i=1}^{n} \left\| \mathbf{u}_i - \pi(\mathbf{R}p_i + \mathbf{p}) \right\|^2 \tag{6}$$

where $\mathbf{R}$ and $\mathbf{p}$ are orientation and position of April-Tag In camera reference C, $\mathbf{u}_i$ is 2D imagePoints coordinate. The camera initial pose is computed by DLT algorithm.

In this paper, the accuracy of pose estimated based on three methods are evaluated. As is shown in Fig. 4, when the camera is close to April-tag, the mean square errors of three methods are all small in three directions. However, in a long distance, the performance of iterative method is better than DLT and EPnP. In conclusion, iterative method is selected to be the tag pose optimization approach in this paper. The method and the experiment results are shown in Fig. 4.

C. Scale Recovery from Both April-Tag and Vehicle Model

Just like ORB-SLAM, our Map Initialization procedure performs an initial feature matching with the previous frame and optimizes the pose at first. However, the scale of the camera trajectory is arbitrary. Therefore, a scale factor s needs to be obtained from vehicle model and April-Tag.

1) Compute Scale by Vehicle model: The vehicle model orientation $\mathbf{R}_{\mathrm{WV}} \in \mathrm{SO}(3)$, position $\mathbf{p}_{\mathrm{WV}}$ and velocity $\mathbf{v}_{\mathrm{WV}}$, in the world reference W is denoted. We assume that the motion of vehicle in a short distance is always planar, so the vehicle motion between two consecutive keyframes can be described in Equation 7.

$$\begin{aligned} \mathbf{R}_{\mathrm{WV}}^{k+1} &= \mathbf{R}_{\mathrm{WV}}^{k} Exp \begin{pmatrix} 0 \\ \omega_{\mathrm{v}} \Delta t \\ 0 \end{pmatrix} \\ \mathbf{v}_{\mathrm{WV}}^{k} &= \mathbf{R}_{\mathrm{WV}}^{k} \begin{pmatrix} v_X \\ 0 \\ v_Y \end{pmatrix} \\ \mathbf{p}_{\mathrm{WV}}^{k+1} &= \mathbf{p}_{\mathrm{WV}}^{k} + \mathbf{v}_{\mathrm{WV}}^{k} \Delta t \end{aligned} \tag{7}$$

Factor s_1 is considered when transforming between camera C and vehicle model V coordinate systems in Equation 8.

$$\mathbf{p}_{\mathrm{WV}} s_1 = \mathbf{R}_{\mathrm{WC}} \mathbf{p}_{\mathrm{CV}} - \mathbf{p}_{\mathrm{WV}} \tag{8}$$

A set of N consecutive keyframes are stacked into a system $\mathbf{A}_{3\times N} s_1 = \mathbf{B}_{3\times N}$ which can be solved via Singular Value Decomposition (SVD) to get the scale s_1.

2) Compute Scale by April-Tag: Each April-Tag's reference is denoted by A_i, with each tag's orientation $\mathbf{R}_{\mathrm{WA_i}} \in \mathrm{SO}(3)$ and position $\mathbf{p}_{\mathrm{WA_i}}$. Two connected keyframes observing the same tag also can be used to solve another scale factor s_2 with Equation 9

$$(\mathbf{R}_{\mathrm{WC}}^{k+1} \mathbf{R}_{\mathrm{CA}_i}^{k+1} \mathbf{p}_{\mathrm{A}_i\mathrm{C}}^{k+1} - \mathbf{R}_{\mathrm{WC}}^{k} \mathbf{R}_{\mathrm{CA}_i}^{k} \mathbf{p}_{\mathrm{A}_i\mathrm{C}}^{k}) s_2 = \mathbf{p}_{\mathrm{WC}}^{k+1} - \mathbf{p}_{\mathrm{WC}}^{k} \tag{9}$$

Being similar to what mentioned above, a set of N consecutive keyframes are stacked into a system $\mathbf{A}_{3\times N} s_2 = \mathbf{B}_{3\times N}$ which can be solved via Singular Value Decomposition (SVD) to obtain the scale s_2.

Final scale factor are denoted as the weighted average value of s_1 and s_2, $s = \kappa s_1 + (1-\kappa) s_2$, where $\kappa \in [0,1]$is the weight parameter representing the confidence of s_1. In addition, the accuracy of s is affected by different keyframe threhold N.

IV. Pose Estimation

A. Outliers Feature Removal with April-Tag

With the optimization of pose extraction of April-Tag mentioned in the previous section, orientation $\mathbf{R}_{\mathrm{WA_i}}$ and position $\mathbf{p}_{\mathrm{WA_i}}$ are obtained. For each keyframe seeing April-Tag, the k^{th} corresponding map points $\mathbf{X}_{\mathrm{C}}^{k}$ are reprojected to the 2D image coordinate by $\pi\left(\mathbf{X}_{\mathrm{C}}^{k}\right)$ where $\pi(\cdot)$ is the mapping relationship from 3D word points to 2D image points. Thus, $\left\| \mathbf{x}_{\mathrm{C}}^{k} - \pi\left(\mathbf{X}_{\mathrm{C}}^{k}\right) \right\|_2$ is used to remove outliers at the beginning.

B. Pose Optimization

As the motion of vehicle is not always planar caused by ramp and road hump, vehicle motion is expanded to 3D by using ORB inlier feature. The first pose estimate is just like ORB-SLAM, and then the position in Y direction of result

is set into the vehicle position $\mathbf{p}_{\mathrm{WV}}$, with the orientation roll angle and pitch angle into the vehicle orientation $\mathbf{R}_{\mathrm{WV}}$ simultaneously.

Once the camera pose is predicted, the mappoints are matched with keypoints to optimize current frame j by minimizing the vehicle error term and the feature reprojection error in which there is a little difference between Map changed and Map Unchanged.

1) Map changed: When map is updated, the vehicle error term links current frame *j* with last keyframe *i* by

$$\begin{aligned} \theta &= \left\{ \mathbf{R}_{\mathrm{WV}}^{j}, \mathbf{p}_{\mathrm{WB}}^{j} \right\} \\ \theta^{*} &= \arg\min \left(\sum_{k} \mathbf{E}_{proj}\left(k,j\right) + \mathbf{E}_{Vehicle}\left(i,j\right) \right) \end{aligned} \tag{10}$$

where the reprojection error $\mathbf{E}_{proj}$ of current frame *j* for given match *k* is defined as follows:

$$\begin{aligned} &\mathbf{E}_{proj}\left(k,j\right) = \rho\left(e_{proj}^{\mathrm{T}} \sum\nolimits_{\mathrm{k}} e_{proj}\right) \\ &e_{proj} = \mathbf{x}^{k} - \pi\left(\mathbf{X}_{\mathrm{C}}^{k}\right) \\ &\mathbf{X}_{\mathrm{C}}^{k} = \mathbf{R}_{\mathrm{CV}}\mathbf{R}_{\mathrm{VW}}\left(\mathbf{X}_{\mathrm{W}}^{k} - \mathbf{p}_{\mathrm{WV}}^{j}\right) + \mathbf{p}_{\mathrm{CV}}, \end{aligned} \tag{11}$$

where $\mathbf{x}^{k}$ is keypoint k location in the image plane. $\mathbf{X}_{\mathrm{W}}^{k}$ is the mappoint matched with keypoint k in the world coordinates. $\sum_{\mathrm{k}}$ is information matrix, and ρ is the Huber robust cost function. The vehicle error term $\mathbf{E}_{Vehicle}$ is denoted by

$$\begin{aligned} &\mathbf{E}_{Vehicle}\left(i,j\right) = \rho\left(\left[e_{\mathrm{R}}^{T} e_{\mathrm{p}}^{T}\right] \Sigma_{I} \left[e_{\mathrm{R}}^{T} e_{\mathrm{p}}^{T}\right]^{T}\right) \\ &e_{\mathrm{R}} = Log\left(\left(Exp\left(\omega_{v}\Delta t_{ij}\right)\right)^{T} \mathbf{R}_{\mathrm{VW}}^{i} \mathbf{R}_{\mathrm{WV}}^{j}\right) \\ &e_{\mathrm{p}} = \mathbf{R}_{\mathrm{VW}}^{i}\left(\mathbf{p}_{\mathrm{WV}}^{j} - \mathbf{p}_{\mathrm{WV}}^{i}\right) - \mathbf{v}_{\mathrm{WV}}^{i}\Delta t \end{aligned} \tag{12}$$

where Σ_{I} is the information matrix of vehicle model. This optimization problem is solved by Levenberg-Marquardt algorithm implemented in g2o [21].

2) Map unchanged: After this optimization, there is no map update, and the next frame $j+1$ will be optimized with a link to frame j using previous optimization as a prior, which is denoted by Equation 13

$$\begin{aligned} \theta &= \left\{ \mathbf{R}_{\mathrm{WV}}^{j}, \mathbf{p}_{\mathrm{WV}}^{j}, \mathbf{R}_{\mathrm{WV}}^{j+1}, \mathbf{p}_{\mathrm{WV}}^{j+1} \right\} \\ \theta^{*} &= \arg\min \left(\sum_{k} \mathbf{E}_{proj}\left(k,j+1\right) + \mathbf{E}_{Vehicle}\left(j,j+1\right) \right. \\ &\qquad \left. + \mathbf{E}_{prior}\left(j\right) \right) \end{aligned} \tag{13}$$

where $\mathbf{E}_{prior}$ is a prior term:

$$\begin{aligned} &\mathrm{E}_{prior}\left(j\right) = \rho\left(\left[e_{\mathrm{R}}^{T} e_{\mathrm{p}}^{T}\right] \Sigma_{\mathrm{p}} \left[e_{\mathrm{R}}^{T} e_{\mathrm{p}}^{T}\right]^{T}\right) \\ &e_{\mathrm{R}} = Log\left(\overline{\mathbf{R}}_{\mathrm{VW}}^{j} \mathbf{R}_{\mathrm{WV}}^{j}\right) \quad e_{p} = \overline{\mathbf{p}}_{\mathrm{WV}}^{j} - \mathbf{p}_{\mathrm{WV}}^{j} \end{aligned} \tag{14}$$

where $\left(\bar{\cdot}\right)$ and Σ_{p} are the estimated states and Hessian matrix resulting from previous optimization. After this optimization, frame j is marginalized and frame $j+1$ become a new prior, then repeat this optimization again until a map change.

C. Loop Closing

In order to utilize April-Tag for loop closing, we introduce a classified method. An April-Tag has a maximum of 53 points, and each corner point of April-Tag is coding into a unique 16-bit Class-ID by

$$p_{ij} = i * 53 + j, \tag{15}$$

where p_{ij} is corner point Class-ID. i is April-Tag ID. j is the index of each corner point in April-Tag. The Class-ID of ORB point is default as -1 and stored in the offline video vocabulary, together with ORB feature descriptor which is 256-bit binary number. 0 bit to 63 bit of the number represents the value of p_{ij}; 64 bit to 127 bit of the number represents the value of p_{ij} and j; 192 bit to 255 bit of the number represents the value of p_{ij}, i and j. The matching method of April-Tag point feature descriptor is to compute the Hamming distance between two feature descriptors. When its value equals to 0, these two descriptors are considered the same.

If the bag of word vector indicates that the current keyframe is similar to the loop keyframe, rotation and translation matrix is calculated by April-Tag between these two frames. Afterwards, an overall BA is applied to all the keyframes on the trajectory similar to the original ORB-SLAM.

V. Experiment and Conclusion

In this section, experiments are presented and evaluated effectiveness of the proposed method in low-light and less-feature garages, which is shown in Fig.5. We evaluate the real-time large scale operation at different vehicle speed, verified with groundtruth data obtained from a Fiber-optic Strapdown Inertial Navigation System. Our system runs in real time, equipped with an IDS 5240CP camera at 20 fps. All experiments are carried out with an Intel(R) Core(TM) i5-6300HQ CPU(four cores @ 2.30GHz) and 8 GB RAM.

We evaluate the accuracy of Scale Recovery at four different vehicle speeds, 5 km/h, 10 km/h and 15 km/h, 20 km/h, with different initialization keyframe threshold N from 1 to 14, which is processed by the scale Recovery in section II. The goal is to test the robustness of Scale Recovery and obtain suitable threshold of keyframes. As shown in Fig.6, when the initialization keyframe threshold N is less than 4, the estimated scale factor is quite different with the optimal scale factor that computed aligning the estimated trajectory with the groundtruth at all vehicle speeds. In contrast, when the threshold N is greater than 6, the estimated scale factor tends to be stable and close to optimal scale factor at 5 km/h. At 10 km/h and 15 km/h, the threshold N needs to be 11 and 13. At 20 km/h, the threshold N needs to be 15. Therefore, the estimated scale factor will converge to the optimal scale factor more quickly at a lower speed by comparing these four pairs. Although the convergence rate of optimal scale factor is slow at a high speed, it still tends to be stable. This indicates that the system is preferred to work at a low speed to make enough variables observable. It also has a good robustness even at a high vehicle speed. The overall result is shown in Fig. 5.

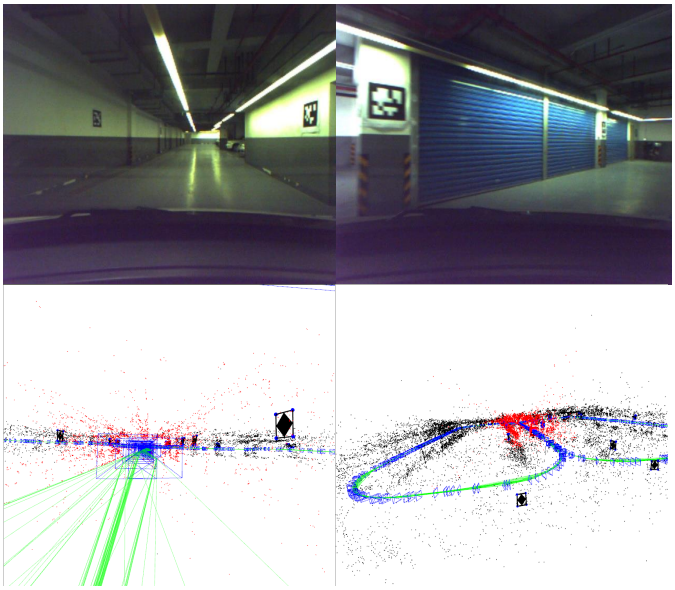

Fig. 5. Top:garage environment. Bottom: estimated map points(red and black), April-Tag(black quadrilateral) and keyframes(blue).

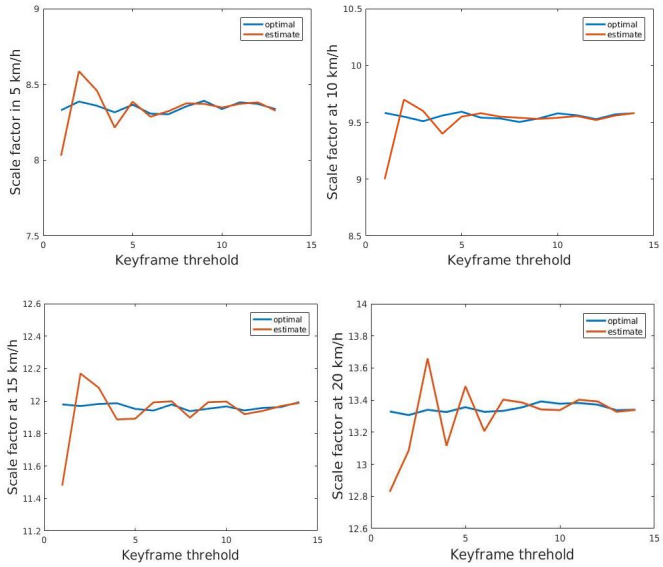

Fig. 6. scale factor at different vehicle speed. Keyframe threhold:the number of keyframes required. Top left: optimal and estimated scale factor at 5 km/h. Top right: optimal and estimated scale factor at 10 km/h. Bottom left: optimal and estimated scale factor at 15 km/h. Bottom right: optimal and estimated scale factor at 20 km/h.

VI. Future Work

In this paper, we use April-Tag as our fiducials. In the future, in order to improve the versatility of the system, we hope to find some common fiducials in the garage, such as parking space, etc. In addition, the accuracy of our system still can be improved. We hope to combine the advantages of IMU and vehicle model, fuse inertial and vehicle model measurement tightly, and optimize bias of inertial based on vehicle model.

Acknowledgment

This work is supported by the following projects:

National Natural Science Foundation of China under Grant 61573260, 61673300 and 91420103;

Science and Technology Commission of Shanghai Municipality under Grant 16JC1401200.

References

[1] A. Kleusberg, "Kinematic relative positioning using gps code and carrier beat phase observations," *Marine Geodesy*, vol. 10, no. 3-4, pp. 257–274, 2009.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: real-time single camera slam." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 6, p. 1052, 2007.

[3] M. W. M. G. Dissanayake, P. Newman, S. Clark, and H. F. Durrant-Whyte, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics & Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[4] D. M. Cole and P. M. Newman, "Using laser range data for 3d slam in outdoor environments," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 1556–1563.

[5] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to slam," in *Ieee/rsj International Conference on Intelligent Robots and Systems*, 2005, pp. 3317–3322.

[6] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 1180–1187.

[7] B. Triggs, P. F. Mclauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *International Workshop on Vision Algorithms: Theory and Practice*, 1999, pp. 298–372.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 1–10.

[10] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Ieee/rsj International Conference on Intelligent Robots and Systems*, 2006, pp. 2276–2282.

[11] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *Robotics IEEE Transactions on*, vol. 24, no. 5, pp. 1027–1037, 2008.

[12] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision*, 2011, pp. 2320–2327.

[13] J. Engel, T. Schps, and D. Cremers, *LSD-SLAM: Large-Scale Direct Monocular SLAM*. Springer International Publishing, 2014.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[15] M. H. Le, V. D. Hoang, A. Vavilin, and K. H. Jo, "One-point-plus for 5-dof localization of vehicle-mounted omnidirectional camera in long-range motion," *International Journal of Control, Automation and Systems*, vol. 11, no. 5, pp. 1018–1027, 2013.

[16] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," pp. 4293–4299, 2009.

[17] R. Murartal and J. D. Tardos, "Visual-inertial monocular slam with map reuse," 2016.

[18] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[19] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," vol. 3951, pp. 430–443, 2006.

[20] 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, pp. 155–166, 2009.

[21] H. Strasdat, "g2o: A general framework for graph optimization," *Plos One*, vol. 7, no. 8, p. e43478, 2011.