2020 16th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
December 13-15, 2020. Shenzhen, China

Visual-IMU State Estimation with GPS and OpenStreetMap for Vehicles on a Smartphone

Guohan He, Qixin Cao, Xiaoxiao Zhu, Haoyuan Miao

Abstract— **In this paper, we propose an approach for ego-motion estimation and vehicle localization using low-cost portable sensors. It combines Visual Inertial Odometry with GPS and street map information obtained from OpenStreetMap. For conventional vehicle driving scenarios, we present a lightweight and targeted multi-sensor fusion method based on graph optimization, and implement the entire system on a smartphone. Extensive experiments on the benchmark dataset and real-world data show that the system could achieve robust, accurate and real-time tracking and localization in complex and dynamic scenarios.**

I. INTRODUCTION

Ego-motion estimation and localization is a basic and important problem in autonomously driving vehicles. A class of popular and effective methods is to use the image information obtained from the camera system as Visual Odometry (VO) [1]. Some excellent systems where visual features [2] [3] or all image pixels [4] [5] are utilized have achieved accurate and stable pose tracking in an ideal environment. However, due to limitations such as lack of metric scale of monocular camera, susceptibility to dynamic environmental interference, and significant cumulative error during long-term operation, VO that only relies on a single sensor fails in a complex environment, such as long-distance driving on real roads.

Inertial measurement unit (IMU) measurements, which can estimate the metric scale and distinguish between dynamic environment and self-motion, are widely used to overcome the limitation of VO, so Visual Inertial Odometry (VIO) is proposed. Unfortunately, long-term drifting in global localization still exists. To solve such issue, loop detection and related global pose optimization are widely utilized in long-term operations. But this method relies on various loops, and is not suitable for conventional outdoor scenes, especially driving vehicles. In this scenario, it is more appropriate to integrate global localization information without accumulated errors, such as GPS information. Considering that the movement of vehicles must be strictly restricted by roads, the global localization information of which can be obtained with a certain accuracy in advance, street maps can also be used as an essential data source to eliminate accumulated errors.

From the perspective of convenience and low cost, a smartphone, which possessed a consumer-level camera, a low-cost inertial measurement unit, a consumer-level GPS chip, and adequate computing power, is a suitable platform that can easily obtain the above data sources and implement fusion tracking algorithms. The challenge mainly comes from two aspects. On the one hand, the accuracy of consumer-level sensors, especially IMU, has a clear gap with industrial products, which means fusion algorithms that fully consider the characteristic of different sensors need to be specially designed. On the other hand, the computing power of a smartphone is inferior to that of a conventional CPU, requiring a lightweight algorithm for real-time operation.

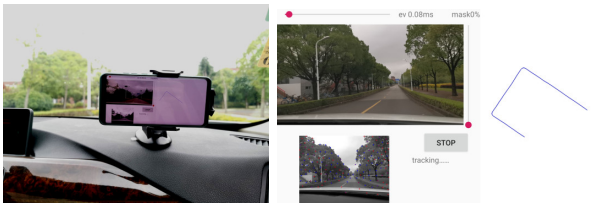

Figure 1. Illustration of our application. The phone is fixed in a position with a good view, shown at the left side. Our user interface is shown at the right side, including control buttons, feature and status feedback and trajectory visualization.

In this paper, we propose a multi-sensor fusion localization algorithm for vehicles. We further describe the applicable scenarios: data of monocular camera and IMU are continuously available, GPS data are partially available, and the vehicle's road information is available in advance through a map. The basic assumption is that VIO has high accuracy over a short distance. Based on this, we make some scene-specific improvements on the existing VIO algorithm and implement the fusion with other sensors to optimize the pose graph. The system is finally implemented stably on a smartphone, and users can easily start tracking and localization by clicking the button on the smartphone, which can be fixed at the co-pilot position of the vehicle, as shown in Fig.1. We summarize our contributions as follows:

- Method for fast localization using accessible sensor system that fuses monocular camera, IMU, GPS information and street maps.
- Real-time implementation of the system on a smartphone.

The rest of this paper is organized as follows. Section II discusses related work. Sections III, IV and V give a detailed analysis of the main modules of the system, of which Section III is for VIO estimation, Section IV is for preprocessing of street maps and GPS information and Section V is for optimization of pose graph for multi-sensor fusion. Extensive experiments and results analysis are displayed in Section VII.

All authors are with School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. Email: {freewind, qxcao, ttl, mhy2015}@sjtu.edu.cn

978-1-7281-7709-0/20/$31.00 ©2020 IEEE

Finally, we conclude this paper and illustrate several future works in Section VIII.

II. Related Work

Because of the excellent complementarity between the monocular camera and IMU, scholarly works on monocular visual-inertial state estimation are extensive. The most direct method to handle visual and inertial measurements is loosely coupled sensor fusion [6] [7], which combines the results of independent estimation of IMU and camera. A more popular method is tightly coupled visual-inertial fusion, where camera and IMU measurements are jointly optimized in original pose estimation. Mainstream algorithms are divided into two categories. One uses extended Kalman filter (EKF) which performs state propagation under the assumption that the state is only related to the previous moment, such as MSCKF [8] and ROVIO [9]. The other uses nonlinear optimization, usually maintaining and optimizing all measurements over a sliding window to optimize the state estimation with bounded computation, such as OKVIS [10], VI-ORB [11] and VINS-Mono [12]. The two categories of algorithms have respective advantages and there is no obvious distinction between them at present.

To overcome drift assumption, algorithms about loop closure in vision-only systems were introduced into some well-known systems such as [11] and [12]. They usually use bag-of-words (BOW) for loop detection and use methods such as pose graph optimization to perform global pose adjustment after loop detection. GPS information is another source used to eliminate accumulated errors. e.g., GOMSF [13] treats fusion with GPS information as the alignment of the local coordinates of the VIO and the global coordinates and updates the alignment transformation by optimizing the latest pose image in a sliding window. Rehder et al. [14] present a graph-based approach, using sparse global measurements to maintain a good average accuracy in long-distance use. There has also been research towards map information utilization for global localization. e.g., Floros et al. [15] add map data from OpenStreetMaps into the observation model and use particle filtering to achieve data fusion, to compensate for the drift that visual odometry accumulates over time.

Since our system is eventually deployed on smartphones, research on state estimation and localization on mobile devices should also be paid attention to. Ventura et al. [16] propose the combination of a keyframe-based monocular SLAM system and a global localization method used in pre-modeled scenarios. Besides tracking process on the mobile client, a server process assists localizing the keyframes with the pre-made map and corrects the tracking results on the client. Li et al. [17] propose a monocular visual-inertial SLAM system and tested it with an iPhone. Limited by sensor accuracy, computing power, etc., SLAM systems on mobile devices are generally used in small scenarios such as indoors, and augmented reality is an important application.

Unlike the general methods, our system focuses on long-term driving on roads. Algorithms about loop closure are not used because in this scenario, loop closure rarely occurs and requires significant computing power and memory, which are both limited for smartphones. Instead, a sliding window graph-based optimization approach combining VIO, GPS and map data is chosen.

III. Visual Inertial Odometry

In the front-end of the VIO module, Harris corners [18] are detected in each image and tracked by the KLT sparse optical flow algorithm [19], which have high response stability and repeatability and can be implemented very efficiently. RANSAC framework is used for preliminary outlier rejection [20]. For IMU measurements, we preintegrate them between continuous keyframes determined by parallax like [12].

In the back-end, we proceed with a tightly coupled monocular VIO based on a sliding window and visual-inertial bundle adjustment. In a sliding window, there are *n* keyframes, the IMU preintegration results between them, and *m* features that are continuously observed. Optimization variables include the state of *k*th keyframe x_k consisting of the pose $p_{b_k}^w$, orientation $q_{b_k}^w$ and velocity $v_{b_k}^w$, the IMU linear acceleration and rotational velocity biases b_a, b_g and the inverse depth of *l*th feature λ_l. The full state vector χ is defined as:

$$\begin{aligned} \chi &= [x_1, x_2, \ldots x_n, \lambda_1, \lambda_2, \ldots \lambda_n] \\ x_k &= [p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a, b_g] \end{aligned} \tag{1}$$

We get a maximum posteriori estimation by minimizing the sum of the Mahalanobis norm of all measurement residuals as:

$$\min_{\chi} \{ \left\| r_p - H_p \chi \right\|^2 + \sum \left\| r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) \right\|_{P_{b_{k+1}}^{b_k}}^2 + \sum \left\| r_C(\hat{z}_l^{c_j}, \chi) \right\|_{P_l^{cj}}^2 \} \tag{2}$$

Where $\{r_p, H_p\}$ is the prior information from marginalization, $r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi)$ and $r_C(\hat{z}_l^{c_j}, \chi)$ are residuals for IMU and visual measurements. The method is similar to [12] and we will not repeat it. The difference is that when trying to add the residuals for new features, the reprojection errors will be calculated using IMU preintegration estimation in advance. Only features with the reprojection error less than the experience threshold associated with the current velocity estimation will be added. Others are marked as outliers and removed later to simplify calculations, which has a certain improvement in dynamic world cases with motion conflicts because RANSAC does not perform well when the motion of environment is consistent (e.g., traffic jams) [21]. The Ceres solver [22] is used for solving this nonlinear problem.

IV. Preprocessing of Map and GPS Data

Map data is obtained from OpenStreetMap (OSM), a well-known open-source map collaboration project. Users can download the needed map of most urban areas from the corresponding website in the XML format, which is structured using three basic entities: *nodes*, *ways* and *relations* [23]. The

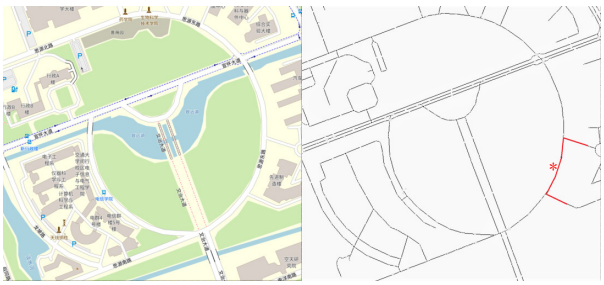

Figure 2. Illustration of OpenStreetMap. Original OSM map is shown at the left side. The extracted street graph is shown at the right side. The set of possible road points corresponding to the coarse GPS positioning indicated by * is highlighted in red.

latitude and longitude of points on the road of interest can be easily resolved from it.

We use GPS measurements to initially obtain the current approximate location, and then determine the set of related points by the relationship between *nodes* and *ways*. Noting that the points in the original set are possibly sparse and uneven, new points are added by interpolation to ensure that the point interval is stable and limited so that the relationship between the estimated localization and the street can be quantified as the shortest distance to these points. This process repeats continuously as the vehicle moves to dynamically maintain a limited set of possible localization points on the road, as shown in Fig. 2.

Since GPS measurements are only used to correct drift assumption, the GPS measurements used can be very sparse but preferably with good accuracy. Therefore, screening is necessary. Compared with the image, the frequency of GPS measurement is much lower, so only those whose timestamps just match the frames will be retained. Measurements that cannot be matched with the street, in other words, those whose minimum distance to the aforementioned point set is too large will also be eliminated.

V. Graph Optimization for Multi-sensor Fusion

Due to the characteristics of the sensors and the sliding window algorithm, the VIO we implemented in Section III still suffers from accumulated drifts, especially in long-term use. Pose graph optimization is developed to ensure the accuracy of global localization, using the map and GPS data obtained in Section IV.

A. State Vector

For global localization, the adjustment of the pose p_i^w and orientation q_i^w of each keyframe (numbered from 1 to n) is the most essential requirement, which is defined in the global coordinate W in VIO. Since the IMU can accurately and reliably determine the direction of gravity, the pitch and roll angles are considered to have no cumulative errors, which means pitch and roll are fixed and only yaw needs to be corrected for orientation. The metric scale s is also added for the possibility of scale drift, especially when using low-cost IMUs. Besides, the world coordinate of VIO is generally different from the reference coordinate of GPS and map, and the transformation between them can be regarded as a special

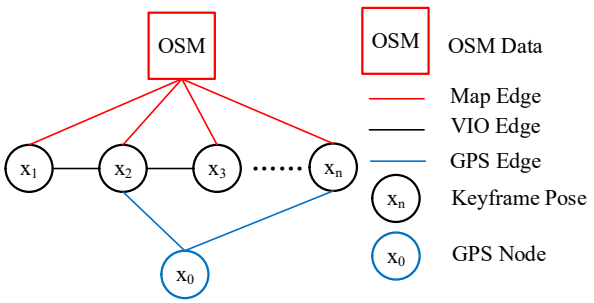

Figure 3. Illustration of the pose graph. The keyframe serves as a vertex and connects other vertexes by VIO edges, GPS edges and Map edges. The transformation between GPS reference system and VIO reference system serves as a special keyframe.

keyframe state, including the pose p_0^w and orientation q_0^w. The full state vector χ in our five degrees-of-freedom (DOF) optimization algorithm is defined as:

$$\chi_p = [p_0^w, p_1^w, p_2^w, \dots p_n^w, q_0^w, q_1^w, q_2^w, \dots q_n^w, s] \tag{3}$$

B. Vertexes and Edges in the Pose Graph

Every keyframe, keyframe *0* included, serves as a vertex in the pose graph and connected with each other by different types of edges from different data sources, as shown in Fig. 3.

1) VIO Edge: A keyframe, keyframe *0* not included, establishes several VIO edges to its previous keyframes, representing the relative transformation between two adjacent keyframes directly acquired by VIO. The transformation between keyframe i and keyframe j includes the relative position $\hat{p}_{ij}^i$ and yaw angle $\hat{\psi}_{ij}^i$ is obtained from the original estimation of VIO (represented as $(\hat{\bullet})$), shown in (4). $\hat{R}_i^w$ represents the rotation matrix from q_i^w.

$$\begin{aligned} \hat{p}_{ij}^i &= \hat{R}_i^{w-1}(\hat{p}_j^w - \hat{p}_i^w) \\ \hat{\psi}_{ij} &= \hat{\psi}_i - \hat{\psi}_j \end{aligned} \tag{4}$$

Combined with the scale factor s, the residual of the edge between adjacent keyframes i and j is defined as:

$$r_{ij}^o(p_i^w, \psi_i, p_j^w, \psi_j) = \begin{bmatrix} R(\hat{\phi}_i, \hat{\theta}_i, \psi_i)^{-1}(p_j^w - p_i^w) - s\hat{p}_{ij}^i \\ \psi_j - \psi_i - \psi_{ij} \end{bmatrix} \tag{5}$$

where $\hat{\phi}_i$ and $\hat{\theta}_i$ are the fixed pitch and roll angles from the VIO, respectively.

2) GPS Edge: A keyframe with GPS information connects the keyframe *0* by a GPS edge in the pose graph, which is similar to the VIO edge. The difference lies in the scale factor and yaw angle because GPS data generally have no orientation information and scale drift. The residual of the GPS edge of keyframes j is defined as:

$$r_j^G(p_0^w, \psi_0, p_j^w, \psi_j) = c_j^G(R(\hat{\phi}_0, \hat{\theta}_0, \psi_0)^{-1}(p_j^w - p_0^w) - \hat{p}_{0j}^0) \tag{6}$$

where c_j^G is the reliability of the GPS data of the keyframe j, obtained from the accuracy related to the number of satellites, distance to road, etc.

3) Map Edge: For normal vehicles, all pose estimations are constrained by the range of the road. The pose p_i^w of the keyframe i needs to be close enough to the pose of a point in the point set S of the street maintained in Section IV. The residual of the Map edge of keyframes i is defined as:

$$r_i^M(p_0^w, \psi_0, p_i^w, \psi_i) = \begin{cases} \sigma(p^*)^{-1}(d_i - \sigma(p^*)) & d_i > \sigma(p^*) \\ 0 & d_i < \sigma(p^*) \end{cases} \quad (7)$$

$$d_i = \min_{p \in S} \left\| R(\hat{\phi}_0, \hat{\theta}_0, \psi_0)^{-1}(p_i^w - p_0^w) - p \right\|$$

where p is the pose of a point in set S, p^* is the pose p corresponding to d_i, and $\sigma(p^*)$ is the allowable deviation on p^*, related to the road width and map accuracy there.

C. 5-DOF Optimization

The whole graph of three types of edges are optimized by minimizing the following cost function:

$$\min_{\chi} \{ \sum \left\| r_{ij}^o \right\|^2 + \sum \left\| r_j^G \right\|^2 + \sum \left\| r_i^M \right\|^2 \} \quad (8)$$

In order to solve the transformation between GPS and VIO coordinates more accurately, the pose graph optimization starts after the vehicle has moved enough distance and has certain GPS data. The pose graph optimization and VIO run in two separate threads.

VI. Experimental Results

A. Dataset Experiments

Since there is no dataset that fully meets our scenario, the outdoor dataset KITTI [24] is used as an alternative to evaluate our method with targeted changes. A stereo visual odometry similar to [25] replaces our VIO to evaluate our graph optimization algorithm for multi-sensor fusion because there is no IMU data that meets the requirements in the dataset. We have also augmented the dataset by downloading the corresponding OSM maps and taken the ground truth of one frame out of every five hundred frames as sparse GPS data. As shown in Fig. 4, compared with loop optimization, our approach using GPS and OSM map data achieved accuracy improvements from 1.16m to 0.89m in absolute trajectory error (ATE) for sequence 00 and from 4.45m to 0.91m for sequence 02. The improvement is more pronounced when the odometry is not accurate enough, like in sequence 02.

B. Real-world experiments

We conducted the real-world experiment on a HUAWEI P30 Pro, with 30-Hz image, 500-Hz IMU, 1-Hz GPS and OSM data of target area in the XML format. The phone is fixed to the car through the bracket and the application tracks in real-time. Ground truth is obtained from a differential-GPS system (DGPS) with centimeter-level positioning accuracy at 20-Hz. We conducted experiments in both ideal and challenging scenarios.

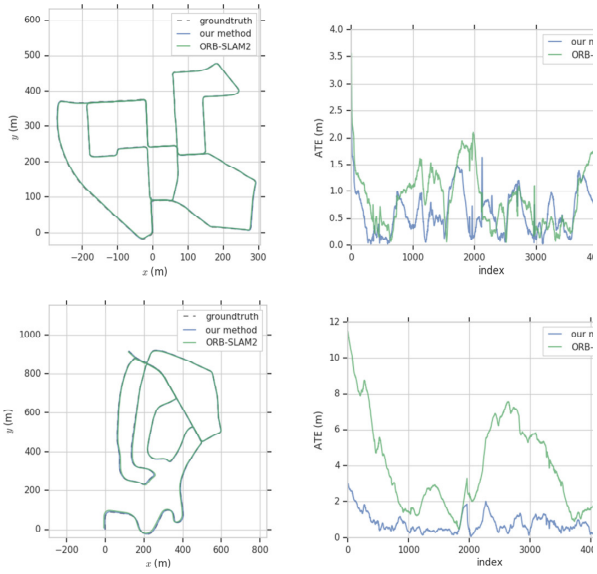

Figure 4. Results for Sequence 00 and 02 of KITTI. The first row shows the results for Sequence 00. The estimated path of our algorithm and ORB-SLAM2 with loop optimization is on the left, and the absolute trajectory error comparison between two methods is on the right. In the

1) Ideal case: The ideal dataset is collected on the SJTU campus, the total length of which is 1.4km. The vehicle is driving at a steady low speed and there are few obstacles. As can be seen in Fig. 5, the proposed fusion algorithm outperforms the traditional visual-inertia method, with a root-mean-square error (RMSE) of 1.87m in comparison to 42.92m of VINS-Mono with loop optimization, which is limited by the accuracy of the low-cost IMU. It is worth noting that the maximum error in 1496 poses is only 4.37m, which can fully indicate that the algorithm has good localization accuracy in an ideal environment.

2) Challenging cases: We conducted experiments on two challenging data sets with different characteristics and the results are shown in Fig. 6. The first dataset goes around the whole SJTU campus, some scenes of which have less or repeated textures. The dataset lasts for 16 min with only one loop and the total path length is 7.8 km. Compared with optimization based on limited closed loops, our method overcomes accumulated drift better, with a RMSE of 2.82 m and a maximum error of 9.26 m. The other dataset is about real streets in Shanghai, China, including dynamic traffic flow, frequent starts and stops, and changing speeds, and lasts for 24 min. Due to the influence of overpasses, etc., the GPS accuracy is very poor in specific areas, and some OSM data also have obvious deviations. The total length is 9.1 km and the whole process restores the true driving status of the vehicle as much as possible. In this complex scenario, our method achieves a RMSE of 6.51 m and a maximum error of 16.22 m, while the odometry has been difficult to maintain accuracy. The results show that our method still achieves relatively stable and accurate localization in complex environments.

3) Run-time Efficiency: For mobile applications, time efficiency is also an important indicator. Timing statistics are

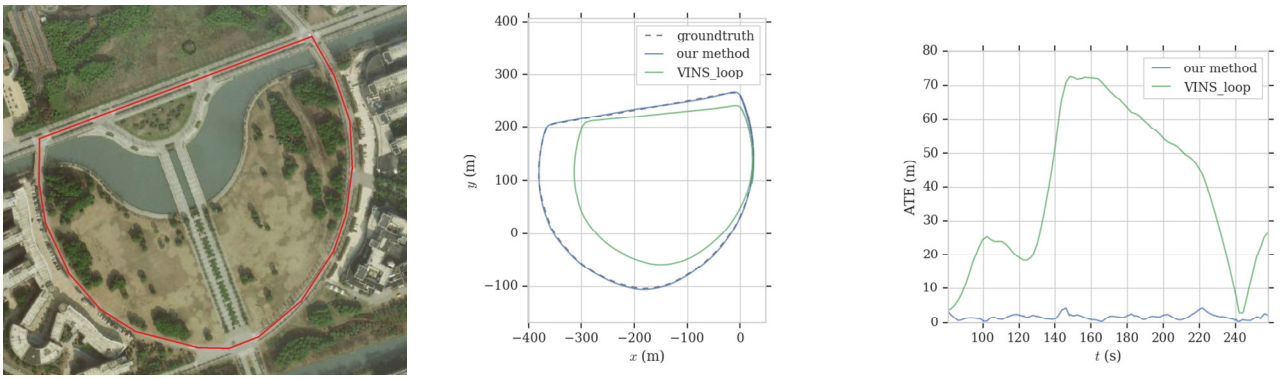

Figure 5. Results for ideal case. From left to right is the trajectory on the satellite map, the estimated path of our algorithm and VINS-Mono with loop optimization, and the absolute trajectory error comparison between two methods.

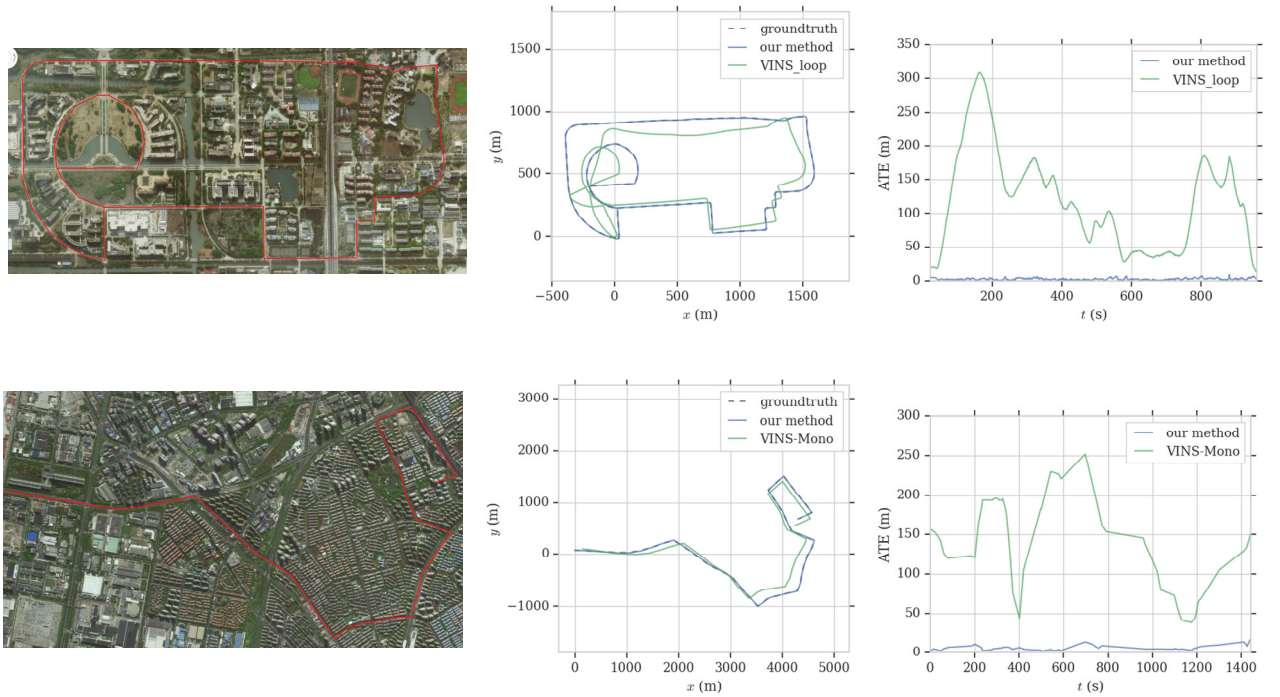

Figure 6. Results for challenging cases. The first row shows the results for dataset on SJTU campus. From left to right is the trajectory on the satellite map, the estimated path of our algorithm and VINS-Mono, and the absolute trajectory error comparison between two methods. In the second row the same plots for dataset of streets in Shanghai, China are illustrated.

shown in Table I. The average run-time and execution frequency of key steps indicate that the program can meet the requirements of real-time operation on mobile phones.

TABLE I. TIMING STATISTICS

Tread	Modules	Time (ms)	Rate (Hz)
1	Feature tracking	25	30
2	Window optimization	50	10
3	Pose graph optimization	200	0.1

VII. CONCLUSION AND FUTURE RESEARCH

We have presented a lightweight approach for a vehicle's global localization and motion estimation. Our approach is based on VIO and fuses GPS and OSM data to overcome the accumulated drifts. We have implemented the details of the algorithm on the smartphone and evaluated it on a series of challenging test cases. The results show that our algorithm is capable of running on a mobile platform stably in real-time and high accuracy can be obtained. Future work will explore the precise localization in some specific scenarios where some sensors have poor accuracy.

ACKNOWLEDGMENT

This research has been supported by National Natural

Science Foundation of China (Grant No. 61673261, No. 61703273) and Shanghai Municipal Science and Technology Innovation Action Plan 2019 Science and Technology Support Project in the Biomedical Field (Grant No. 19441908300).

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry part I: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1051–1067, 2007.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147 – 1163, 2015.

[4] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for RGBD cameras," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, Tokyo, Japan, Sep 2013, pp. 2100–2106.

[5] 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, March 2018.

[6] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in Proc. IEEE Int. Conf. Robot. Autom., 2012, pp. 957–964.

[7] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.

[8] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, Apr. 2007, pp. 3565–3572.

[9] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 298–304.

[10] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2014.

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

[12] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.

[13] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart and M. Chli, "GOMSF: Graph-Optimization Based Multi-Sensor Fusion for robust UAV Pose estimation," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 1421-1428.

[14] J. Rehder, K. Gupta, S. Nuske and S. Singh, "Global pose estimation with limited GPS and long range visual odometry," *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012, pp. 627-633.

[15] G. Floros, B. van der Zander and B. Leibe, "OpenStreetSLAM: Global vehicle localization using OpenStreetMaps," *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013, pp. 1054-1059.

[16] J. Ventura, C. Arth, G. Reitmayr and D. Schmalstieg, "Global Localization from Monocular SLAM on a Mobile Phone," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 531-539, April 2014, doi: 10.1109/TVCG.2014.27.

[17] P. Li, T. Qin, B. Hu, F. Zhu and S. Shen, "Monocular visual-inertial state estimation for mobile augmented reality", *Proc. IEEE Int. Symp. Mixed Augmented Reality*, pp. 11-21, 2017.

[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference (AVC'88)*, 1988.

[19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, Canada, Aug. 1981, pp. 24–28.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[21] B. P. W. Babu, D. Cyganski, J. Duckworth and S. Kim, "Detection and Resolution of Motion Conflict in Visual Inertial Odometry," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 996-1002.

[22] S. Agarwal *et al*., "Ceres solver." [Online]. Available: http://ceressolver.org

[23] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[24] A. Geiger, P. Lenz, and U. R., "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[25] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017