

GNSS Reconstrained Visual–Inertial Odometry System Using Factor Graphs

Yu Chen¹, Bo Xu¹, Bin Wang¹, Jiaming Na², *Member, IEEE*, and Pei Yang

Abstract—Monocular vision sensors are often affected by the rapid direction change in load platform and violent illumination change when the mobile device moves autonomously with high maneuverability. The images collected by the visual sensor will also have a lot of dynamic blur, which together with the weak texture environment reduces the continuity and accuracy of the visual autonomous navigation system. To enhance the stability of the system, in the letter, we propose a vision-led multisource data fusion navigation algorithm. The system combines the visual information for trajectory estimation, adds the inertial measurement unit (IMU) measurement information to the sliding window for optimization, and finally uses the global navigation satellite system (GNSS) data as a reconstraint condition through factor graph optimization to further optimize the trajectory accuracy. Experiments on the public datasets containing a variety of different scene categories show that the trajectory tracking results generated by our algorithm are more complete and stable and can better meet the system’s autonomous navigation requirements.

Index Terms—Factor graph, global navigation satellite system (GNSS), multisource data fusion, visual autonomous navigation, visual–inertial odometry (VIO).

I. INTRODUCTION

IN RECENT years, the technology of mobile intelligent robot has attracted more and more attention both in the society and research area. The simultaneous localization and mapping (SLAM) technology provides key support in the autonomous navigation of intelligent robots with real-time trajectory and environmental information and further plays a decisive role for the robot’s subsequent path planning and navigation. In view of the uncertainty in outdoor environments, obtaining accurate pose and environmental information through visual sensors and inertial navigation devices therefore faces many challenges. At present, classic solutions such as visual inertial system (VINS)-mono [1] and oriented FAST and rotated BRIEF (ORB)-SLAM3 [2] system have higher accuracy and robustness in ideal environment, but due to the error from feature tracking, especially the ORB feature matching [3], [4], with the error accumulation of inertial

navigation device, the system would produce a huge trajectory drift when the loop environment is not detected for a long time.

The current differential global navigation satellite system (GNSS) technology [real-time kinematic (RTK)] based on carrier phase has a positioning accuracy of centimeter level [5]; however, in practical applications of GNSS major-based systems, the occlusion of tall objects such as trees and buildings will affect the strength of GNSS signals, resulting in positioning failure. For this reason, Miller et al. [6] proposed a map assistance method that combines the GNSS data for visual SLAM and particle filtering. Schleicher et al. [7] proposed a method that combines the monocular vision ranging method and GNSS measurement to restore position and attitude. Berrabah et al. [8] proposed real-time extended Kalman filter (EKF) hierarchical SLAM combined with the GNSS data, but did not use the height information provided by GNSS. Chu et al. [9] developed an integrated vision/inertial measurement unit (IMU)/GNSS system based on the EKF design, which can provide 15-state high-bandwidth navigation solutions. However, the EKF only uses the latest state estimation due to the marginalization of the past state. As a result, the nonlinear factors representing these measured values cannot be relinearized well during the estimation process.

Different from the above solutions, in the letter, we use factor graph optimization to fuse the real-time vision, inertial navigation, and GNSS data collected by the mobile platform. Factor graph [10] is a probability graph model of state variables and factor nodes. Given all the available sensor measurements as factors, it will encode the posterior probability of the state over time. The inertial/visual information fusion method based on factor graph is a major direction of the factor graph algorithm [11]. This kind of algorithms uses a factor graph model to express the fusion of inertial information and visual information, which is closely integrated with the SLAM technology. Factor graph encodes the relationship between unknown variable nodes and known measured values, and fusing the observed values from different and asynchronous sensors becomes a new problem of connecting the factor defined by the measured value with the corresponding node in the factor graph [12], [13], [14]. With the help of the factor graph algorithm, the GNSS information is added as a constraint in the visual–inertial odometry (VIO) system, which can further improve the navigation quality of the system and the accuracy of the final generated trajectory.

II. METHODOLOGY

As shown in Fig. 1, based on the VIO system with visual constraints and IMU constraints, we use the node coordinate

Manuscript received 31 March 2022; revised 4 October 2022; accepted 10 January 2023. Date of publication 13 January 2023; date of current version 3 February 2023. This work was supported by the National Natural Science Foundation of China (NSFC) under Project 42004002. (Corresponding author: Bin Wang.)

Yu Chen, Bin Wang, and Pei Yang are with the School of Geomatics and Technology, Nanjing Tech University, Nanjing 211816, China (e-mail: ychen7121@njtech.edu.cn; binwangsgg@njtech.edu.cn; peiyang@njtech.edu.cn).

Bo Xu is with the School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China (e-mail: boxu1995@whu.edu.cn).

Jiaming Na is with the College of Civil Engineering, Nanjing Forestry University, Nanjing 210037, China (e-mail: jiaming.na@njfu.edu.cn).

Digital Object Identifier 10.1109/LGRS.2023.3236803

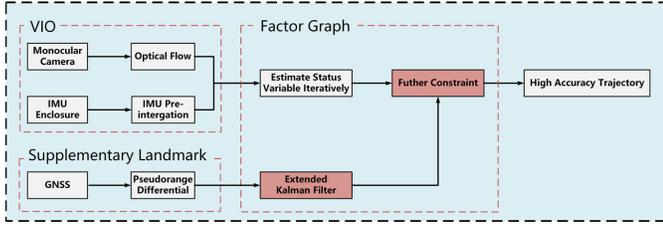


Fig. 1. Algorithm overall flow.

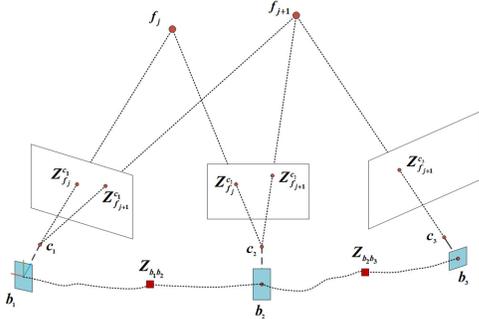


Fig. 2. Illustration of VIO.

information provided by GNSS as a further constraint through the factor graph and reconstrain the trajectory synchronized with GNSS time nodes.

A. Basic Principles of the VIO System

Fig. 2 shows the basic principle of visual inertial navigation odometer and stipulates the following notation [15]. c_i and b_i , respectively, represent the camera observation and IMU observation at time $t = i$. (\cdot) represents the observation value in frame c . q_{bc} and p_{bc} are the internal parameters between the camera system and the IMU system. f_j indicates the landmark point of the point feature. z represents the observation on the image frame.

1) *Point Feature Measurement Model*: According to the point features, assuming that the normalized plane coordinate of point k on the c_j th frame is $z_{f_k}^{c_j} = [u_{f_k}^{c_j}, v_{f_k}^{c_j}, 1]^T$, the reprojection error can be defined as

$$r_f(z_{f_k}^{c_j}, X) = \begin{bmatrix} \frac{x^j}{z^j} - u_{f_k}^{c_j} \\ \frac{y^j}{z^j} - v_{f_k}^{c_j} \end{bmatrix} \quad (1)$$

where $z_{b_i b_j}$ is the feature point on the normalized plane first observed by c_i frame, and λ_k is the inverse depth of the feature point in this frame.

2) *IMU Measurement Model*: The IMU measurement residual model is constructed by preintegrating the original IMU observations of two consecutive frames b_i and b_j . The model can be defined as follows:

$$r_b(z_{b_i b_j}, X) = \begin{bmatrix} r_p \\ r_\theta \\ r_v \\ r_{ba} \\ r_{bg} \end{bmatrix} = \begin{bmatrix} R_{b_{i w}}(P_{w b_j} - P_{w b_i} - V_i^w \Delta t + \frac{1}{2} g^w \Delta t^2) - \hat{a}_{b_i b_j} \\ 2[\hat{q}_{b_i b_j} \otimes (q_{b_{i w}} \otimes q_{w b_j})]_{xyz} \\ R_{b_{i w}}(v_j^w - v_i^w + g^w \Delta t) - \hat{\beta}_{b_i b_j} \\ b_a^{b_j} - b_a^{b_i} \\ b_g^{b_j} - b_g^{b_i} \end{bmatrix} \quad (2)$$

where $z_{b_i b_j} = [\hat{a}_{b_i b_j}, \hat{\beta}_{b_i b_j}, \hat{q}_{b_i b_j}]$ is the preintegration measurement of IMU. $[\cdot]_{xyz}$ is the real part of quaternion, which is used to estimate the 3-D rotation error.

3) Optimization of the VIO System:

In the letter, we use the sliding window algorithm to optimize the point feature visual measurement error and IMU measurement error, so as to optimize the tight coupling of visual information and inertial navigation information. The variable optimized in the sliding window at time t can be defined as follows:

$$X = [x_n, x_{n+1}, \dots, x_{n+N}, \lambda_m, \lambda_{m+1}, \dots, \lambda_{m+M}]$$

$$x_i = [p_{w b_i}, q_{w b_i}, v_i^w, b_a^{b_i}, b_g^{b_i}]^T, \quad i \in [n, n+N] \quad (3)$$

where x_i represents the state variable of the IMU body at the i th moment, $p_{w b_i}$ is the position of the IMU body in the world coordinate system, while $q_{w b_i}$ represents its direction and v_i^w represents its speed in the same coordinate system. $b_a^{b_i}$ and $b_g^{b_i}$ represent the acceleration and angular velocity bias at the same moment, respectively. Subscripts n indicate the IMU body state variables, and m represents the number of the beginning of the feature point in the sliding window. N is the number of keyframes in the sliding window, and M is the number of feature points observed by all the keyframes in the sliding window. k is the inverse depth of the k th landmark in the camera coordinate system where the landmark is first time observed.

By constructing the residual equation to optimize all the state variables in the sliding window, the least-square solution is obtained as

$$\min \rho \left(\|r_p - J_p X\|_{\sum p}^2 \right) + \sum_{i \in B} \rho \left(\|r_b(z_{b_i b_{i+1}}, X)\|_{\sum b_i b_{i+1}}^2 \right) + \sum_{(i,j) \in F} \rho \left(\|r_f(z_{f_i c_i}, X)\|_{\sum f_i c_i}^2 \right). \quad (4)$$

In the formula, $\{r_p, J_p\}$ is the prior information calculated after one frame of edge in the sliding window. J_p is the prior Jacobian matrix obtained by calculating the Hessian matrix according to the last sliding window optimization. $r_b(z_{b_i b_{i+1}}, X)$ represents the IMU measurement residual between states x_i and x_{i+1} . $r_f(z_{f_i c_i}, X)$ is the reprojection error of point feature. B is the set of all the preproduct components in the sliding window. F represents the set of point features observed by the camera. ρ is the Cauchy robust kernel function.

B. GNSS Double-Difference Positioning

The observations between the rover and the reference value are taken as the single difference between stations, and the previous observations continue to participate with the observations from different satellites p and q to make the single difference between the stars. The station-satellite double-difference observation equation can be taken as

$$\lambda \nabla \Delta \Phi_{ij}^{pq} = \rho_j^q - \rho_i^q - \rho_j^p + \rho_i^p - \lambda \nabla \Delta N_{ij}^{pq} + \nabla \Delta e_{\Phi, ij}^{pq} \quad (5)$$

where λ is the wavelength of the carrier phase, Φ is the observed value of the carrier phase, ρ is the geometric distance

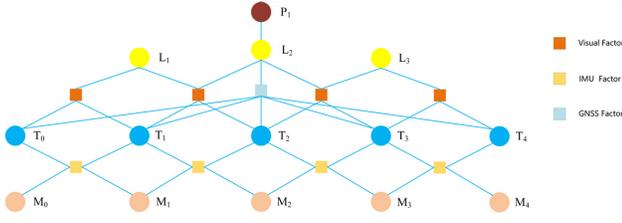


Fig. 3. Multiple constraint factor graph.

between the stars and the ground, N is the ambiguity of the whole week, ∇ is the difference operator between stars, and $\nabla\Delta$ is the station star double-difference operator. At the same time, there is $\lambda \nabla \Delta \Phi_{ij}^{pq} = \lambda \Delta \Phi_{ij}^q - \lambda \Delta \Phi_{ij}^p$, that is, $\lambda(\Phi_j^q - \Phi_i^q - \Phi_j^p + \Phi_i^p)$, in the same way $\nabla \Delta N_{ij}^{pq}$.

Linearize ρ_j^q and ρ_i^p in the observation equation, then there is

$$\rho_j^q = \rho_{j0}^q + [-l_j^q - m_j^q - n_j^q]dX \quad (6)$$

$$\rho_i^p = \rho_{i0}^p + [-l_i^p - m_i^p - n_i^p]dX. \quad (7)$$

In the equations, ρ_{j0}^q and ρ_{i0}^p are the approximate coordinates. If $\nabla \Delta e_{\Phi,ij}^{pq}$ is abbreviated, the linearized double-difference carrier observation equation can be made as

$$\lambda \nabla \Delta \Phi_{ij}^{pq} = [-(l_j^q - l_i^p) - (m_j^q - m_i^p) - (n_j^q - n_i^p)]dX - \lambda \nabla \Delta N_{ij}^{pq} + \nabla \Delta \rho_{ij0}^{pq} + \nabla \Delta e_{\Phi}. \quad (8)$$

Among $\nabla \Delta \rho_{ij0}^{pq} = \rho_{j0}^q - \rho_{i0}^p - \rho_{j0}^p + \rho_{i0}^q$, after simplification, there is

$$\lambda \nabla \Delta \Phi_{ij}^{pq} - \nabla \Delta \rho_{ij0}^{pq} = -(l_j^q - l_i^p) \cdot dX - \lambda \nabla \Delta N_{ij}^{pq} + \nabla \Delta e_{\Phi}. \quad (9)$$

The above formula is the double-difference RTK positioning error equation. Using the distance and spatial correlation characteristics between the station and the satellite, the common systematic errors of different observation values are eliminated and weakened by double difference, and the fixed fuzzy real-time high-precision mobile station position solution can be obtained using the least square.

C. GNSS-VIO Fusion Factor Graph

As a kind of undirected graph, factor graph is usually formed by combining two kinds of nodes, one is called the variable node, which is used to represent optimized variables, and the other is called the factor node, which is used to represent factors. As shown in Fig. 3, in the multiconstraint factor graph of GNSS-assisted VIO fusion in this letter, there are three kinds of “variable nodes,” one is the visual input variable node T_i , one is the IMU input variable node M_i , and the other is a fixed node P_i generated from GNSS data. There are also three types of “factor nodes,” which are visual factor nodes in red, inertial navigation factor nodes in yellow, and GNSS input variables as reconstraint factor nodes in blue of landmark nodes L_i . The edges represent different constraints, through which the trajectory is optimized.

To facilitate optimization, the GNSS processing strategy adopted in this letter is to first convert the GNSS date into

TABLE I
SPECIFIC DATA INFORMATION OF KITTI IMAGE SEQUENCE

Serial number	Sequence name	Start image	End point image	IMU data frequency	GNSS data frequency
02	2011-10-03-drive-0034	000000	004660	100Hz	100Hz
07	2011-09-30-drive-0027	000000	001100	100Hz	100Hz
08	2011-09-30-drive-0028	001100	005170	100Hz	100Hz

general horizontal axis Mercator coordinates, and then associate each GNSS data with the trajectory node [16]. Then the position of GNSS can be used as a prior position information, which becomes a node and a reconstraint in the factor graph. The error between the translation vector of the trajectory node and the GNSS position is defined as follows:

$$e_i^g = T_t - G_i \quad (10)$$

where T_t is the trajectory of the mobile platform at time t , and G_i is the fixed node generated by GNSS data. The integrated navigation algorithm based on factor graphs needs to solve the maximum posterior probability of the parameter to be estimated, that is,

$$X^{\text{MAP}} = \arg \min_X \sum_i \|T_t - G_i\|_{\Sigma_i}^2. \quad (11)$$

III. EXPERIMENTS AND ANALYSIS

A. Monocular/IMU/GNSS Fusion Navigation Experiment

At present, the industry still lacks a complete open solution for monocular/IMU/GNSS integrated navigation. In the section, we compared our own algorithm with the monocular/IMU integrated navigation systems such as VINS-mono, OKVIS, and ORB-SLAM3 to verify the constraint effect from GNSS to the tight coupling system of monocular and IMU.

1) *Description of Experimental Datasets:* To evaluate the performance of this algorithm, the observational datasets KITTI 02, KITTI 07, and KITTI 08 in Table I are used including both the categories: the extract and the sync data. The sync data remove the distortion of the camera and synchronize the IMU data with the GNSS data. This type of data releases only 10-Hz frequency of IMU and GNSS data, which cannot meet the requirements of multisensor data fusion. And the extract data (raw data) are the original observation data; although the images are not removed distortion, the data release 100-Hz frequency of IMU data and GNSS data. Thus, the 10-Hz IMU data and global positioning system (GPS) data in the corresponding sync dataset are replaced with the 100-Hz data in the extract dataset. The specific information of the three groups of KITTI data are as follows.

In the above datasets, the measured IMU data in the extract dataset have intermittent and reverse time scales, as shown in Fig. 4. We only preprocess the reverse order problem.

2) *Analysis of Experimental Results:* In the letter, we evaluate the test results of the above three sets from our algorithm, VINS-mono algorithm, ORB-SLAM3 algorithm, and OKVIS algorithm using absolute trajectory error (ATE).

As shown in Fig. 5, in terms of trajectory tracking results, our algorithm has achieved best trajectory generation results among the three sets of trajectories. The VINS-mono algorithm also achieved better results, followed by the ORB-SLAM3 algorithm, but the OKVIS was weaker. Part of the reason

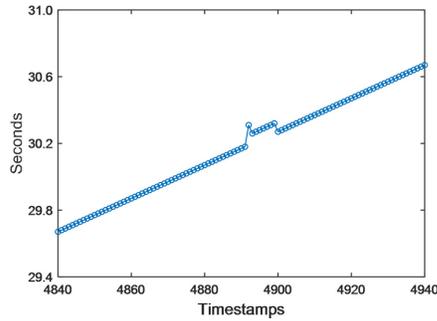


Fig. 4. IMU time scale jump.

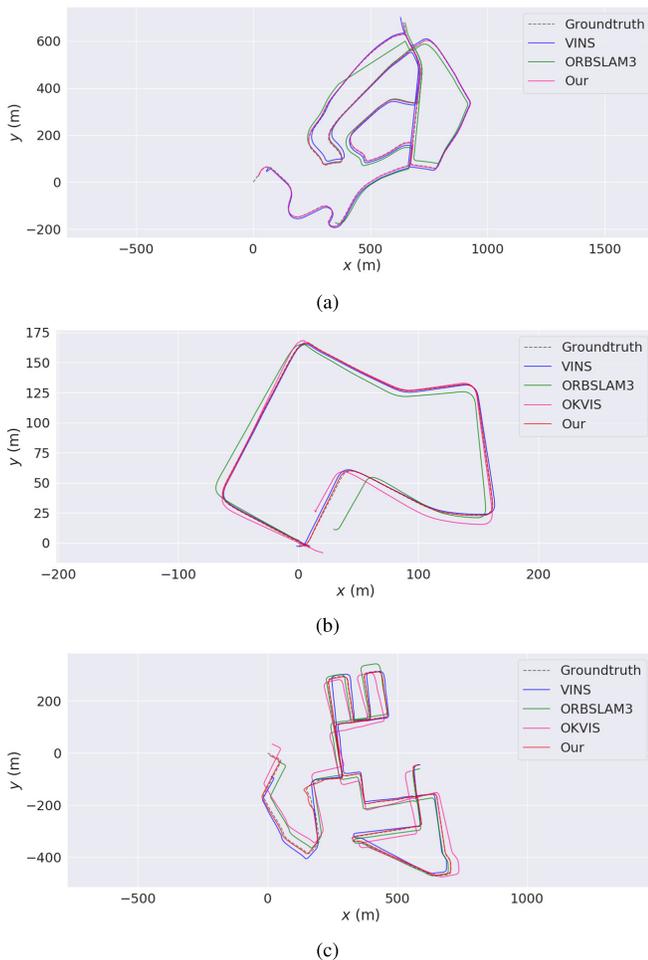


Fig. 5. Comparison of algorithm trajectories. (a) Algorithm trajectories of KITTI 02. (b) Algorithm trajectories of KITTI 07. (c) Algorithm trajectories of KITTI 08.

for these results is the issue of timescale jumps in the IMU data. The intermittent timescale problem of KITTI data mainly occurs in KITTI 08 data. Our algorithm preprocesses the intermittent of IMU timescale in the program, so it can quickly and accurately initialize the pose and track the trajectory. The other algorithms, except VINS-mono, do not adopt the above mechanism. When facing the problem of intermittent time scale in IMU data, they may lose tracking and reinitialize map points, resulting in large errors to the overall trajectory in the continuous initialization steps.

TABLE II
TRAJECTORY ACCURACY RESULTS OF EACH METHOD ON THE KITTI DATASET

sequence	ATE(in m) Transl. MAX			ATE(in m) Transl. RMSE			ATE(in m) Transl. MIN		
	KITTI02	KITTI07	KITTI08	KITTI02	KITTI07	KITTI08	KITTI02	KITTI07	KITTI08
VINS-Mono	45.635	3.745	82.561	12.765	1.750	17.821	1.931	0.760	2.525
ORB-SLAM3	30.306	31.163	68.007	16.198	11.419	18.619	1.589	1.527	2.391
OKVIS	fail	30.293	75.842	fail	9.175	32.696	fail	0.339	1.120
Ours	9.611	2.172	30.382	2.876	0.886	4.776	0.780	0.169	1.092

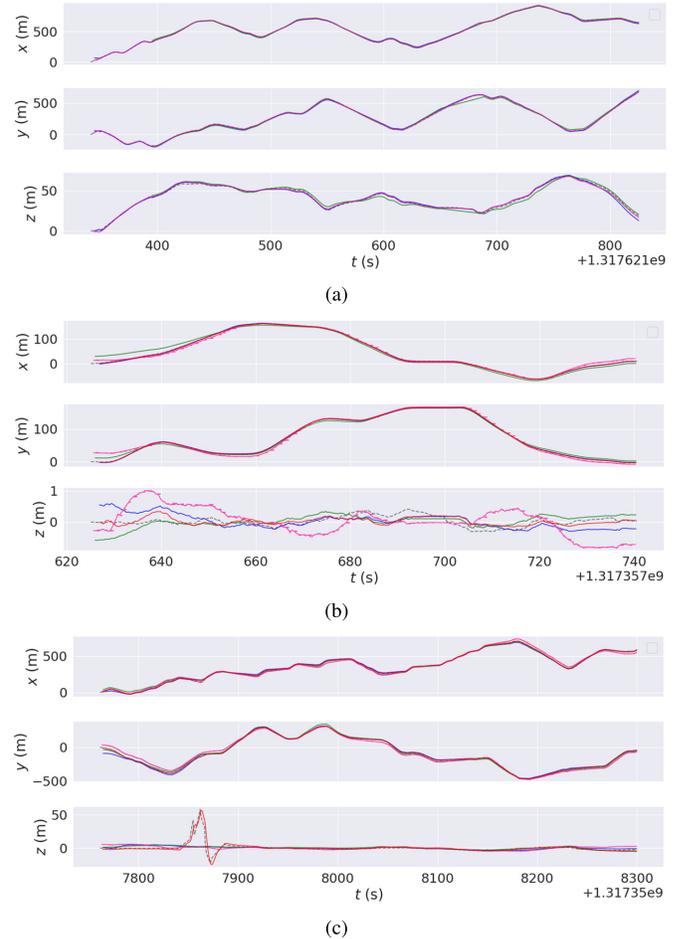


Fig. 6. Comparison of trajectory accuracy. (a) Trajectory accuracy of KITTI 02. (b) Trajectory accuracy of KITTI 07. (c) Trajectory accuracy of KITTI 08.

In the letter, we listed the trajectory estimation accuracy of all the algorithms in Table II. By the reconstrained of GNSS data, our algorithm is further refined from the original VIO trajectory results. In the experiment, we use a low-weight GNSS data for constraints, mainly considering that the actual accuracy of GNSS data may be lower than the GNSS data in the KITTI public dataset due to the accumulation of errors such as occlusion and jitter during the actual operation. A high-weight GNSS data constraint may affect the final trajectory accuracy of the system.

As can be seen from Fig. 6, the accuracy of each algorithm in the X direction and the Y direction is not much different from the true trajectory. However, there seems to be large differences in the Z direction (elevation direction). Since the range of the Z-axis is different from X and Y (0–50 m or even 0–1 m vs. 0–100 m or even 0–500 m), the actual differences are not so large. In Fig. 6(c), the trajectory trend of our method

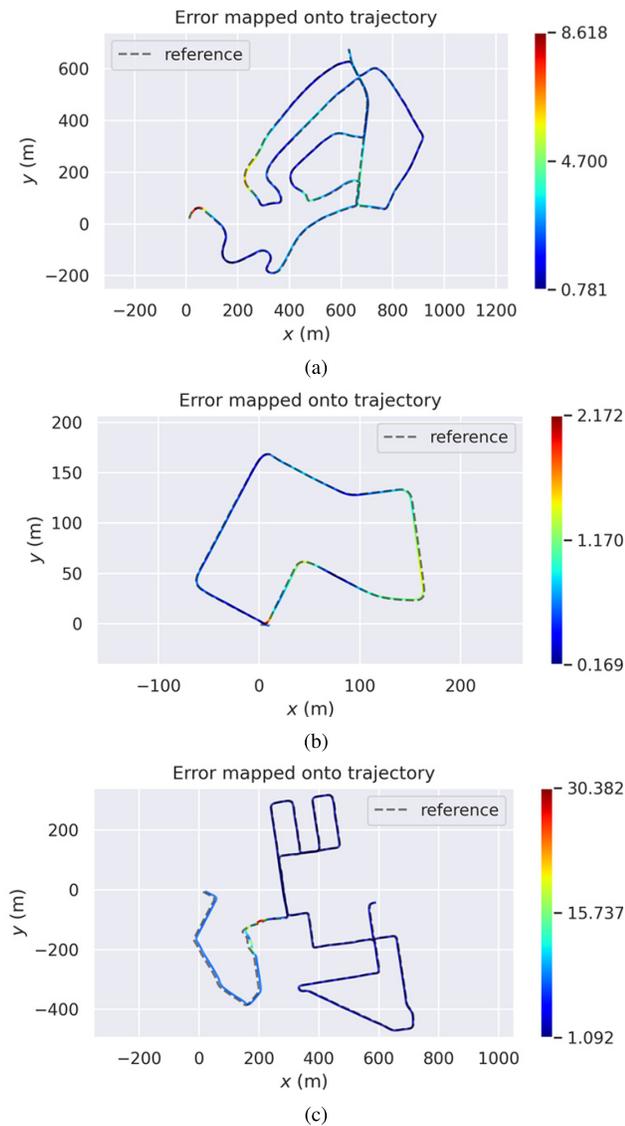


Fig. 7. Algorithm trajectory versus ground truth. (a) Error map of Algorithm in KITTI 02. (b) Error map of Algorithm in KITTI 07. (c) Error map of Algorithm in KITTI 08.

has a small difference from the ground truth, while the other methods have a large one in the specific area, this is because our method adopts a higher weight of the GNSS constraint in the KITTI 08 dataset, which is close to the GNSS ground truth and also lost in the same area due to occlusion and other reasons, resulting in the jump of trajectory results. The result also shows that a GNSS constraint with high weight may affect the final trajectory accuracy of the system.

In Fig. 7, the coincidence degree between our method and the ground truth is reflected by the color of the track. The more the color is toward dark blue, the higher the track coincidence, and the higher the track accuracy. As can be seen, our method has error in the initial stage of the trajectory; with the operation of the system and the gradual tracking of the trajectory, the system accuracy is getting higher. This indicates that under the continuous constraint of GNSS data, the trajectory estimation accuracy of our method can be continuously improved with the operation of the system.

IV. CONCLUSION

In the letter, a visual odometer optimization algorithm integrating monocular vision, IMU inertial navigation data, and GNSS data is proposed, in which the GNSS data can be used for reconstraint with the monocular and IMU inertial navigation data are tightly coupled. In the experiment, three sets of KITTI original observation data are used to evaluate the trajectory accuracy of our algorithm and other excellent visual inertial navigation odometer systems as VINS-mono, ORB-SLAM3, and OKVIS. The results show that in the actual trajectory tracking process, the proposed algorithm can further restrict the overall trajectory accuracy with the help of GNSS data on the trajectory estimation of mobile devices independently through monocular vision and inertial navigation data. The experiment verifies the effectiveness of our algorithm through the actual trajectory tracking results by three sets of KITTI data: KITTI 02, KITTI 07, and KITTI 08.

REFERENCES

- [1] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [3] Y. Chen, L. Yan, B. Xu, and Y. Liu, "Multi-stage matching approach for mobile platform visual imagery," *IEEE Access*, vol. 7, pp. 160523–160535, 2019.
- [4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
- [5] M. Zhang, X. Xu, Y. Chen, and M. Li, "A lightweight and accurate localization algorithm using multiple inertial measurement units," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1508–1515, Apr. 2020.
- [6] I. Miller, M. Campbell, and D. Huttenlocher, "Map-aided localization in sparse global positioning system environments using vision and particle filtering," *J. Field Robot.*, vol. 28, no. 5, pp. 619–643, 2011.
- [7] D. Schleicher, L. M. Bergasa, M. Ocana, R. Barea, and M. E. Lopez, "Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 440–452, Sep. 2009.
- [8] S. A. Berrabah, H. Sahli, and Y. Baudoin, "Visual-based simultaneous localization and mapping and global positioning system correction for geo-localization of a mobile robot," *Meas. Sci. Technol.*, vol. 22, no. 12, Dec. 2011, Art. no. 124003.
- [9] T. Chu, N. Guo, S. Backén, and D. Akos, "Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments," *Sensors*, vol. 12, no. 3, pp. 3162–3185, 2012.
- [10] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [11] Y. Ben Elisha and V. Indelman, "Active online visual-inertial navigation and sensor calibration via belief space planning and factor graph based incremental smoothing," in *Proc. IEEE/RSS Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2616–2622.
- [12] Q. Zeng, W. Chen, J. Liu, and H. Wang, "An improved multi-sensor fusion navigation algorithm based on the factor graph," *Sensors*, vol. 17, no. 3, p. 641, Mar. 2017.
- [13] C. Tang, L. Zhang, Y. Zhang, and Y. Wang, "Factor graph aided distributed multi-navigation cooperative positioning algorithm," in *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS+)*, Oct. 2018, pp. 2421–2428.
- [14] G. Junqiang, T. Xiaqing, Z. Huan, and W. Meng, "Land vehicle INS/GNSS/OD integrated navigation algorithm based on factor graph," *Syst. Eng. Electron.*, vol. 40, no. 11, pp. 2547–2553, 2018.
- [15] B. Xu, Y. Chen, S. Zhang, and J. Wang, "Improved point-line visual-inertial odometry system using Helmert variance component estimation," *Remote Sens.*, vol. 12, no. 18, p. 2901, 2020.
- [16] Z. Yu and L. Jiang, "The application of improved 3D SLAM algorithm in mobile robot," (in Chinese), *Mech. Des. Manuf.*, vol. 1, no. 1, pp. 29–32, 2020.