Sensor Synchronization for Android Phone Tightly-Coupled Visual-Inertial SLAM

Zheyu Feng, Jianwen Li and Taogao Dai

Abstract At present, the majority of Android phones only support satellite positioning and cellular localization. Both of them are of poor indoor performance, which limits the development of the relevant indoor location based services. In this paper, we attempt to achieve positioning with raw image and Inertial Measurement Unit (IMU) data from Android phone. We first introduce a state-of-the-art framework for tightly-coupled monocular visual-inertial Simultaneous Localization and Mapping (SLAM) using image and IMU data. Then we focus on the unsynchronization problem between camera and IMU of Android phone, and propose a grid search algorithm based on spherical quaternion interpolation for delay estimation. The results of indoor and outdoor experiments show that the algorithm can estimate the delay of image timestamp effectively, and the percentage of positioning plane error is 0.79% indoors and 8.09% outdoors respectively.

Keywords Android phone · Tightly-coupled · Synchronization
SLAM · VINS

1 Introduction

With the popularization of smartphones and the increasing demand for location information, location based services, especially on smart phone, have become an indispensable part of people's daily life. The foundation of location based service is positioning. Although Google has launched Tango project in June 2014 to achieve

Z. Feng · J. Li (✉)
Information Engineering University, Zhengzhou 450001, China
e-mail: zzljw@126.com

Z. Feng
e-mail: von9604@gmail.com

T. Dai
63883 Troops, Luoyang 471000, China
e-mail: 18538881619@163.com

J. Sun et al. (eds.), *China Satellite Navigation Conference (CSNC) 2018 Proceedings*, Lecture Notes in Electrical Engineering 499,
https://doi.org/10.1007/978-981-13-0029-5_52

visual navigation using infrared sensor and image sensor, the main positioning methods for Android phone are still satellite positioning and cellular localization. Satellite positioning is of high accuracy with poor anti-interference capability, and cannot be used indoors. Meanwhile, although cellular localization is available indoors, the positioning accuracy can only reach several hundred meters [1], which cannot meet the requirement of indoor navigation. It's obvious that the technology bottleneck of positioning is one of the main factors that limit the development of indoor location based services.

SLAM, which aims to achieve localization in unknown environment, is one of the research hotspots of autonomous navigation for mobile robot at present. Visual SLAM is attracting more and more attention. Because image sensor is usually of wide detection range, and visual image has large amount of information and rich features easy to extract [2]. Whereas, all feature-based visual SLAM solutions do not have robust dynamic performance. They are easy to lose tracking with severe motion.

IMU has a high sampling frequency and is highly sensitive to the motion of the carrier. Additionally, accurate motion measured from IMU can be used for monocular SLAM to recover scale. Therefore, the Integration of a monocular camera and an IMU becomes a popular low-cost SLAM solution. A robust and versatile monocular visual-inertial state estimator based on sliding window graph-based optimization framework called VINS is proposed in [3], which has been well applied in Apple's smart devices [4]. Thus we attempt to achieve positioning with this framework with data from Android phone.

The visual-inertial System sensors usually synchronize directly on hardware level, so sensors are assumed synchronized before data processing [3–6]. However, at present, the image and the IMU are not synchronized in Android system, which will result in failure for the visual-inertial SLAM. To our best knowledge, the main research about sensor synchronization focuses on correcting rolling shutter effect [7] of an image with gyroscope data to achieve image stabilization [8, 9] or more accurate feature extraction [10]. In [11–13], synchronization between camera and IMU is considered in the visual-inertial SLAM and online calibration of temporal parameter is proposed. However, the algorithm is hard to implement for the VINS graph-based optimization framework. Therefore, in this paper, aiming at solving the unsynchronization problem, a simple algorithm for image delay estimation is proposed. Experimental results show that, the framework proposed in [3] can be improved for Android phone with this algorithm.

2 VINS Framework

In this section, we will briefly introduce the VINS framework applied [3]. The framework can be divided into four parts: sensor data pre-processing, system initialization, nonlinear optimization and loop closure. The overall framework is shown in Fig. 1.

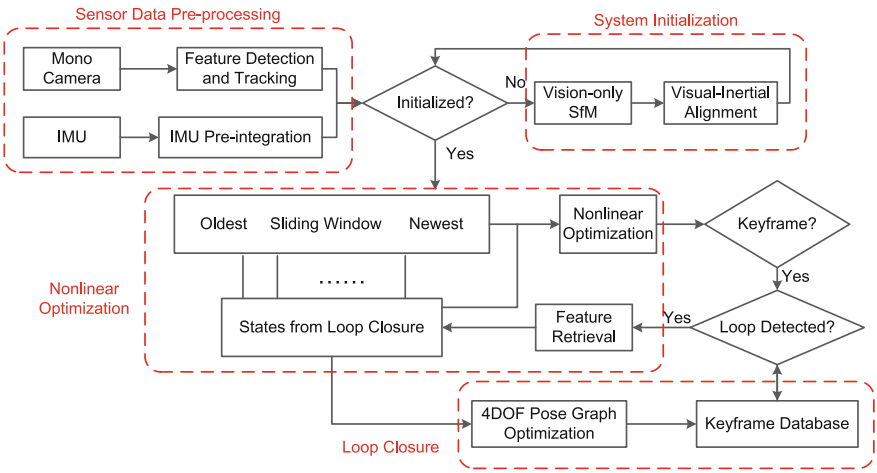

Fig. 1 VINS overall framework

The specific process of the framework is as follows:

(1) **Sensor Data Pre-processing** For the camera output images, features are tracked by KLT sparse optical flow algorithm. Simultaneously, new feature points are detected to ensure that the number of features of each image is fixed. After that, RANSAC fundamental matrix test is applied to remove outliers. In this process, the keyframes are also selected. If the average parallax of a frame is higher than the set threshold or the number of feature points tracked is below the set threshold, the frame is regarded as a keyframe. On the other hand, the gyroscope and accelerometer measurements are pre-integrated. The image feature points and IMU pre-integrals obtained in this step are saved in a sliding window for subsequent system initialization and non-linear optimization.

(2) **System Initialization** First of all, by setting IMU average acceleration threshold, the system ensures the observability of the scale. Then, the structure of the visual frames is restored by triangulating the image feature points in the sliding window. The relative rotations and the up-to-scale translations between adjacent image frames are obtained. Thus initial values for system scale, gravity, position, velocity, attitude, and bias can be yielded from these rotations and translations with the pre-integrals of IMU measurements. After that, the initial gravity value is optimized to the standard value.

(3) **Nonlinear Optimization** After initialization, velocity, position, attitude, IMU-camera extrinsic parameters, bias, and feature point depths are nonlinearly optimized in the tightly-coupled state estimator. After optimization, the states in the sliding window are required to be added and deleted. If the second frame in the sliding window is a keyframe, it will be retained and the last frame will be deleted; if the second frame in the sliding window is not a keyframe, its visual

observations will be removed while its IMU measurements will be retained. The states are marginalized out to preserve a priori information when they are deleted. If the number of feature points tracked from the latest frame is less than 5 or the two consecutive state values differ greatly, the system considers fault happened and reinitializes, with previously recorded keyframes kept.

(4) **Loop Closure** FAST feature points are extracted and identified with BRIEF descriptor. Then these points are classified with bag-of-words technique. When a loop closure is recognized for a new image frame, the states in the sliding window are optimized together with the feature points of the looped keyframe. And the relative pose between the new image frame and the looped frame. Since the yaws and coordinates (x, y, z) of the entire system contain accumulated drifts, it is also necessary to finally optimize all poses in 4 degrees of freedom to obtain accurate states in the sliding window.

In summary, the VINS Framework algorithm incorporates the latest research achievements in the field of monocular visual-inertial SLAM. It has excellent implementation on measurement data pre-processing, system initialization, nonlinear optimization and loop-closure detection. However, because the monocular camera and IMU in Android phone are not synchronized, the visual-inertial SLAM cannot be applied directly for Android phone.

3 Delay Estimation

In the Android system, sensor data is obtained through the system-provided callback function. Camera images are acquired using the function onPreviewFrame (byte [] data, Camera camera) call, while IMU data can be obtained with the function onSensorChanged (SensorEvent event). SensorEvent contains sensor data and accurate measurement time [14, 15]. However, the callback function for camera image does not provide a timestamp. As a result, the time when the function is called is taken as image taken time for timestamping. Thus, there is a delay t_d between the timestamp t' of the image and the image taken time t, namely $t' = t + t_d$. The delay is the time interval between the moment of taking an image and the moment when the image is passed to the callback function. In addition, the delay is related to the specific system operation situation (Figs. 2 and 3).

3.1 System Configuration

In order to make this delay as constant as possible, the system is required to do the following settings [16].

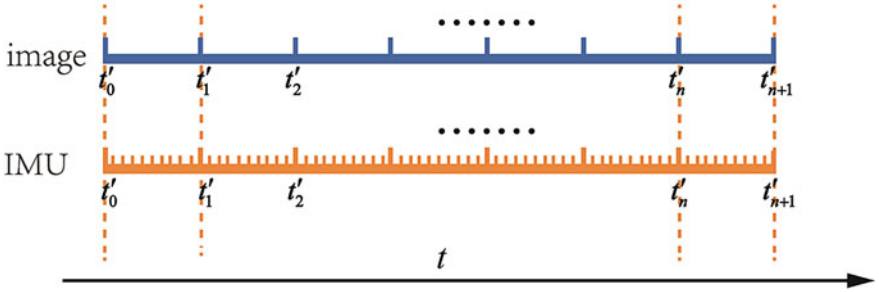

Fig. 2 Timestamps when camera and IMU are synchronized $t_d = 0$

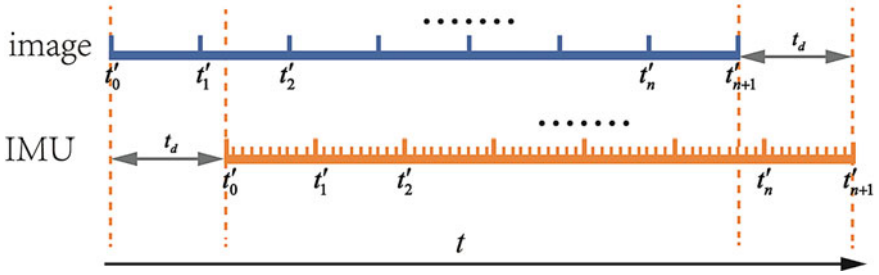

Fig. 3 Timestamps when camera image timestamps contain a delay $t_d > 0$

- Disable the camera auto exposure, lock the exposure time;
- Put the IMU data and camera image callback function into a separate thread respectively, to avoid the user interface (UI) thread blocking leading to delay instability;
- Avoid too many operations in the camera image callback function, to ensure that the callback function has been completed before called next time.

3.2 *Algorithm for Delay Estimation*

Assuming the IMU attitude change between t_i and t_{i+1} is $\gamma_{b_{i+1}}^{b_i} = \gamma_{b_{i+1}}^{b_i}(t_i, t_{i+1})$, and the camera attitude change is $\gamma_{c_{i+1}}^{c_i} = \gamma_{c_{i+1}}^{c_i}(t_i, t_{i+1})$, assuming the extrinsic rotation between the IMU and the camera q_c^b is known, we can get the following relationship.

$$\begin{aligned}\gamma_{b_{i+1}}^{b_i} &= q_{b_i}^{w^{-1}} \otimes q_{b_{i+1}}^{w} \\ &= (q_{c_i}^{w} \otimes q_b^c)^{-1} \otimes q_{c_{i+1}}^{w} \otimes q_b^c \\ &= q_c^b \otimes \gamma_{c_{i+1}}^{c_i} \otimes q_b^c\end{aligned} \tag{1}$$

Then we have

$$\gamma_{b_{i+1}}^{b_i} \otimes (q_c^b \otimes \gamma_{c_{i+1}}^{c_i} \otimes q_b^c)^{-1} = \mathbf{0} \tag{2}$$

where $\otimes$ is quaternion multiplication, $(\cdot)^{-1}$ is quaternion inversion. Select two frames with adjacent timestamps t_i' and t_{i+1}' respectively. And a fundamental matrix is calculated with the feature pairs are matched for calculation. Then fundamental matrix is decomposed to obtain the camera's attitude change.

Due to the delay of image timestamp, the camera attitude change $\hat{\gamma}_{c_{i+1}}^{c_i}$ form t_i to t_{i+1} is actually obtained. And the IMU angular velocity measurements from t_i' to t_{i+1}' are pre-integrated [17] to get the IMU attitude change $\hat{\gamma}_{b_{i+1}'}^{b_i'}$.

In order to obtain the delay, we first use a sliding window of size n to store the IMU attitude changes and the camera attitude changes. Then the IMU attitude changes are taken as reference. And the camera attitude changes at each aligned time are obtained by spherical interpolation [18] when the delay parameter changes. Thus residuals can be calculated, and we can find the delay that results in the minimum residual.

$$e = \sum_{i=1}^{k} \frac{\hat{\gamma}_{b_{i+1}'}^{b_i'} \otimes (q_c^b \otimes \hat{\gamma}_{c_{i+1}'}^{c_i'} \otimes q_b^c)^{-1}}{k} \tag{3}$$

In fact, in the window, the number of the camera attitude changes could vary with the different delay parameter. In order to ensure that the number of the camera attitude changes in the window is constant, we need to remove the non-public parts of the referenced IMU attitude changes according to the search range. The specific algorithm is shown in Fig. 4.

We determined the search range as [−0.5s, 0.5s], and found that the result was stable when the window size $n = 50$. The variation of residual with delay is shown in Fig. 5.

According to the analysis above, the image timestamps in Android phone must be greater than the image taken time. Thus the delay must be above 0, and only the delay greater than 0 need to be considered. It can be seen that the residual has a minimum value within the search range. We add this algorithm into the VINS system initialization part to ensure sensor synchronization for subsequent procedures.

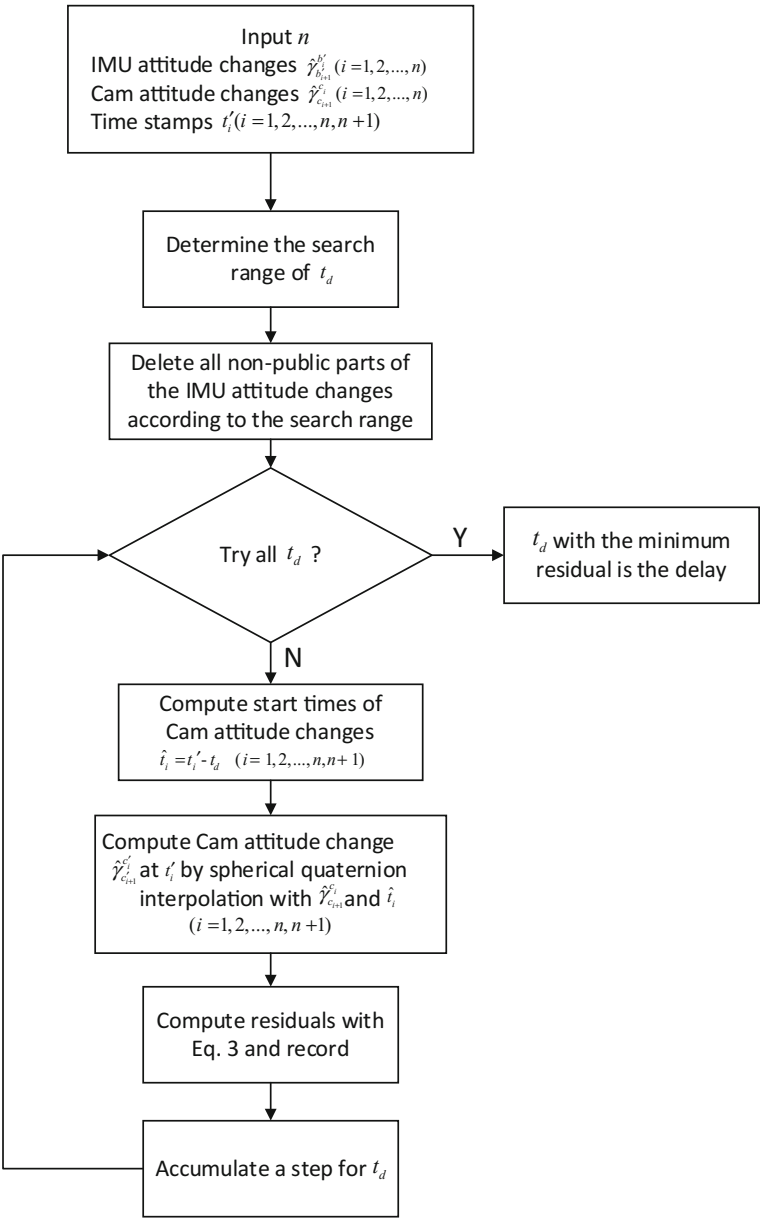

Fig. 4 The algorithm searches in the determined range of t_d, and obtains the camera attitude changes at each aligned time by spherical interpolation. It calculates the residual between the attitude changes of the camera and the IMU in the window. The delay with the minimum residual is the delay between image timestamp and image taken time

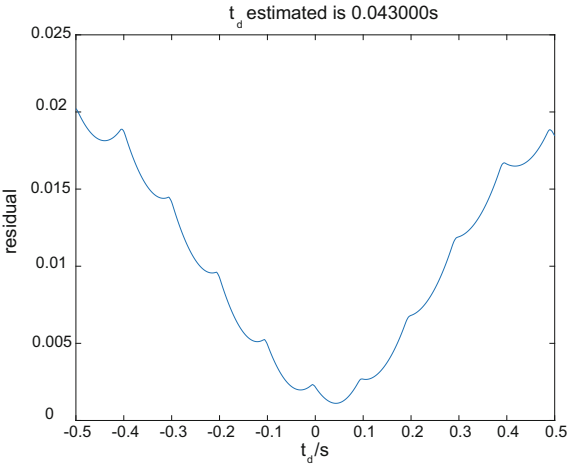

Fig. 5 Variation of residual with delay. It's apparent residuals are approximately symmetrical about the estimated delay

4 Experiments and Analysis

In order to verify the validity of the algorithm, we use a Huawei P9 mobile phone to obtain images and IMU data indoors and outdoors respectively. The improved VINS algorithm is used to locate the phone with loop closure detection disabled. During the experiments, mobile phone image sampling rate is set to 20 Hz, IMU sampling rate is 100 Hz. Experimenter holds Android mobile phone on hand and walk along a closed experimental line with the speed of about 1.5 m/s. The start and end point of the closed experimental lines are coincident. The accuracy is verified by calculating the coordinate difference between the start and end point.

4.1 Indoor Experiment

Indoor experiment is performed in an indoor library, the experimental route is about a square. In the process of system initialization, the variation of the residual with different delay is shown in Fig. 6, so we can get the delay of image timestamp $t_d = 0.0240$ s.

Positioning results after synchronization is shown in Fig. 7. The blue points in Fig. 7a are the map points constructed by the visual-inertial SLAM, and the black line is the trajectory of the Android phone. The orange points are the map point constructed with current visual frame. The final positioning plane accuracy is shown in Table 1.

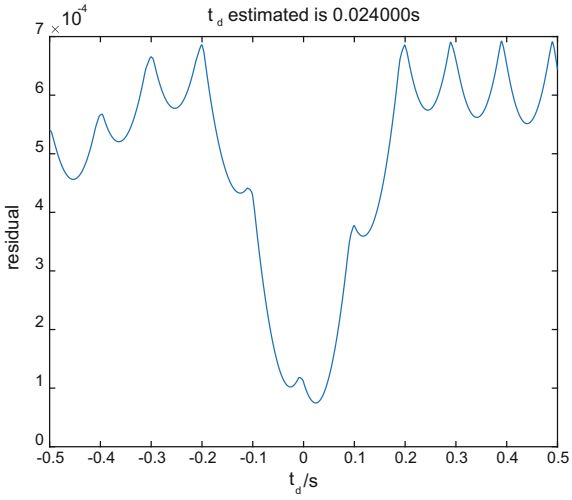

Fig. 6 Variation of residual with different delay in the indoor experiment

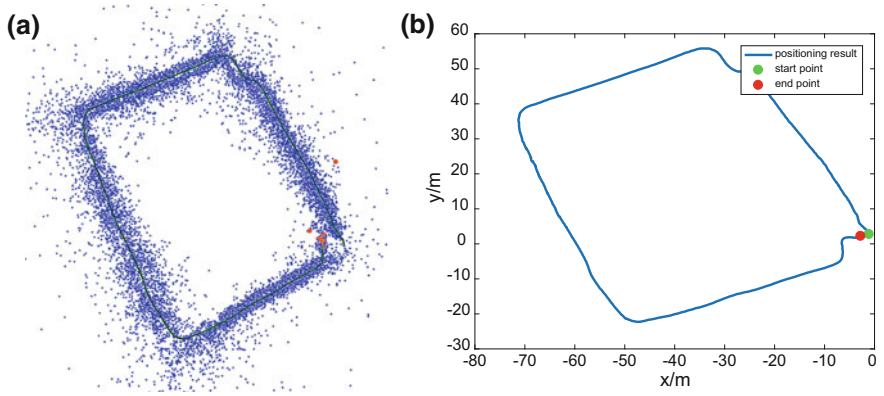

Fig. 7 Positioning results of the indoor experiment. **a** Carrier trajectory and map points during data processing. **b** Final positioning result

Table 1 Plane accuracy of indoor and outdoor experiments

	Duration (s)	Distance (m)	Plane accuracy (m)	Error percentage (%)
Indoor	164.216	230.704	1.829	0.79
Outdoor	271.594	400.000	32.366	8.09

4.2 Outdoor Experiment

Outdoor experiment is performed in an athletic field, the experimental route is the runway there. In the process of system initialization, the variation of residual with different delay is shown in Fig. 8, we can get the delay of image timestamp $t_d = 0.0162$ s. Positioning results after synchronization is shown in Fig. 9. The final positioning plane accuracy is shown in Table 1.

4.3 Result Analysis

It can be seen from above results that the delay search algorithm can effectively estimate the delay of image timestamp so that the VINS algorithm can be carried out smoothly. The indoor positioning accuracy is much better than the outdoor positioning accuracy, because in the large-scale outdoor environment, most of map points are far away from the camera. In this case, the same pixel error of the image can result in significantly larger triangulation error of map points, so the accuracy of poses is much worse.

5 Summary

In this paper, aiming to solve the unsynchronization problem between Android camera and IMU, an algorithm for delay estimation is proposed. With this algorithm, the VINS algorithm could be applied to the Android raw data successfully.

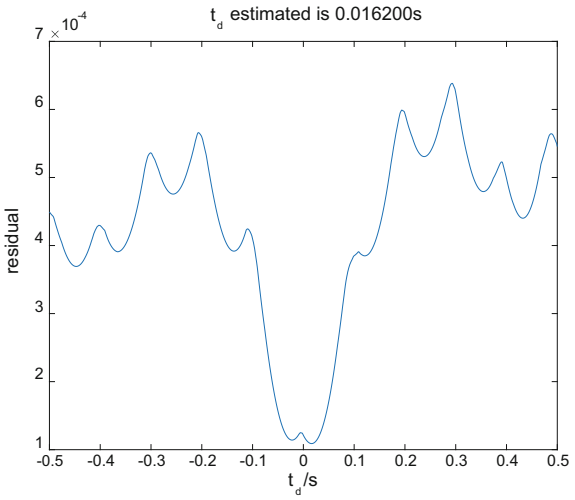

Fig. 8 Variation of residual with different delay in outdoor experiment

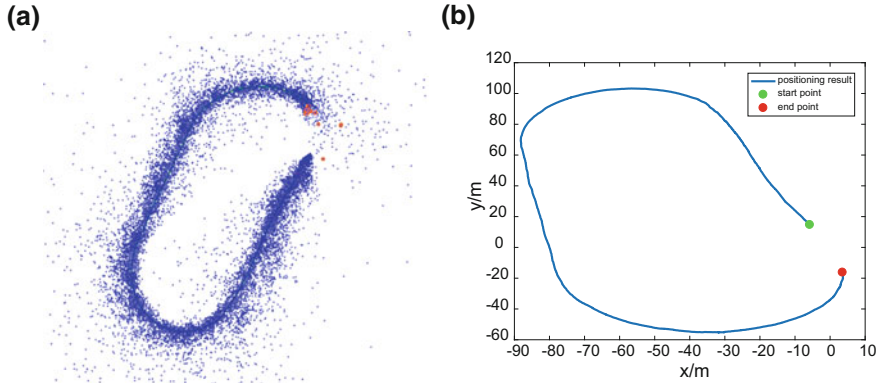

Fig. 9 Positioning results of outdoor experiment. **a** Carrier trajectory and map points during data processing. **b** Final positioning result

The experimental results show that the positioning accuracy indoor and outdoor after the synchronization is 0.79 and 8.09%, respectively. The validity of the synchronization algorithm is verified, and the good performance of the VINS algorithm for indoor localization is also proved. It is foreseeable that in the near future, visual-inertial SLAM will be an effective means for Android users to achieve indoor positioning and navigation.

References

1. Xu Y (2016) The research and implementation of positioning technology based on base station of mobile terminal. Dalian Maritime University, China
2. Quan M, Piao S, Li G (2016) An overview of visual SLAM. CAAI Trans Intell Syst 11 (06):768–776
3. Qin T, Li P, Shen S (2017) VINS-mono: a robust and versatile monocular visual-inertial state estimator. arXiv preprint arXiv:1708,03852
4. Li P, Qin T, Botao H et al (2017) Monocular visual-inertial state estimation for mobile augmented reality. In: Proceedings of the IEEE international symposium on mixed and augmented reality, Nantes, France
5. Corke P, Lobo J, Dias J (2007) An introduction to inertial and visual sensin. Int J Rob Res 26 (6):519–535
6. Kelly J, Sukhatme G (2011) Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration. Int J Rob Res 30(1):56–79
7. Geyer C, Meingast M, Sastry S (2005) Geometric models of rolling shutter cameras. In: Proceedings workshop omnidirectional vision
8. Karpenko A, Jacobs D, Baek J, Levoy M (2011) Digital video stabilization and rolling shutter correction using gyroscopes. Technical Report, Stanford University
9. Hanning G, Forslöw N, Forssén PE, Ringaby E, Törnqvist D, Callmer J (2011) Stabilizing cell phone video using inertial measurement sensors. In: Proceedings of IEEE international workshop on mobile vision

10. Hwangbo M, Kim J-S, Kanade T (2011) Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation. Int J Rob Res 30(14):1755–1774
11. Jia C, Evans BL (2012) Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements. In: Proceedings of IEEE international workshop multimedia signal processing
12. Li M, Kim B, Mourikis A (2013) Real-time motion tracking on a cellphone using inertial sensing and a rolling shutter camera. In: Proceedings of IEEE international robotics and automation
13. Li M, Mourikis A (2013) 3-D motion estimation and online temporal calibration for camera-IMU systems. In: Proceedings of IEEE international robotics and automation
14. Google. Camera API. https://developer.android.com/
15. Google. Motion sensors. https://developer.android.com/
16. Latimer R, Holloway J, Veeraraghavan A et al (2014) SocialSync: sub-frame synchronization in a smartphone camera network. Computer vision—ECCV 2014 workshops. Springer International Publishing, Basel, pp 561–575
17. Forster C, Carlone L, Dellaert F et al (2015) IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Robotics: science and systems
18. Shoemake K (1985) Animating rotation with quaternion curves. In: International conference on CGIT, pp 245–254