SLAM System Based on Tightly Coupled Visual-inertial

Zhang Chen, Wang Mei*, Yu Yunlei and Gao Yongyun

College of Mechatronics Engineering and Automation, Shanghai University, Shanghai, 200444, China

*Corresponding author’s e-mail: mwang@shu.edu.cn

Abstract. Aiming at the problem of mobile robot positioning, IMU information is used to improve the performance of visual SLAM. Firstly, instead of the calculation of feature descriptor, LK optical flow is used in the processing of visual information, to increase the speed of feature extraction. And detectors are used to remove outliers. Secondly, the visual information and IMU information are initialized separately to obtain the camera pose and IMU pose. And then Visual-inertia joint initialization is conducted by tightly coupling of the IMU information and the visual information, to improve the calculation. Finally, the pose state estimation is achieved through back-end optimization. The validity of the proposed algorithm is verified by the experiments.

1. Introduction
In recent years, with the development of mobile robots, more and more researchers have explored the problem of simultaneous localization and mapping (SLAM). Although pure vision can achieve positioning and mapping in 3D [1]-[3] the reliability is still poor according to the problems, such as motion blur, occlusion, fast motion, pure rotation, and scale uncertainty. While IMU (Inertial Measurement Unit) can provide the angular velocity and acceleration of the moving target, it is used with the vision usually to realize the positioning. And the technique of filtering or optimization is used to improve the positioning accuracy and robustness. Some techniques, can be used to overcome the shortcomings mentioned above, such as visual pose estimation, pre-integration [4]-[5] of IMU state quantities, tight coupling technology [6] that combined IMU with vision, and new least squares solution strategy. They can effectively implement pose estimation in SLAM problems.

2. The SLAM Method Based on Tightly coupled visual inertia
In the SLAM system that combines IMU with vision sensors, the visual and IMU information are processed separately to obtain their own pose, including their position, speed, and rotation angle. Then the information of camera and IMU are tightly coupled and initialized to obtain the joint state variables. The backend optimization in the sliding window is used to achieve pose estimation finally. The framework of Visual-IMU SLAM (VI-SLAM) is shown in Figure 1.

In the VI-SLAM system, three coordinate systems of the world, as shown in Figure 2, the IMU and the camera are involved, and can be represented as \((\cdot)^w\), \((\cdot)^b\), and \((\cdot)^c\) separately.
3. Visual tracking and visual initialization
To perform visual pose estimation, key frames should be extracted from the image sequence by visual tracking, and the rotation angle and translation are calculated by visual initialization.

3.1. Visual tracking
Feature extraction and feature descriptor matching are used usually in the vision-based SLAM method, and then the motion of the camera is estimated by the principles of projective geometry. However, it will be difficult in scenes of lacking obvious texture information, in which features are missing seriously. Without the calculation of descriptors, Lucas-Kanade (LK) optical flow method combined with FAST features can deal with the problem well.

After the extraction of FAST features, the LK optical flow algorithm is used for tracking, and the points that fail to track are eliminated at the same time.

FAST feature at time $t$ is located at $(x, y)$, and will move to $(x + dx, y + dy)$ at time $t + dt$, according to the assumption of invariance of gray (as shown in equation (1)).

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

(1)

Based on the assumption that the FAST features in a certain window on the image have the same motion, the Taylor's formula is used to expand the formula (1) to track the features.

In order to control the number of features at about 200, new features are extracted in some specific area. And then the features are sifted out by a detector, based on the principle of uniform distribution. The detector, based on graph-cut optimized RANSAC (GC-RANSAC) [7] will eliminate outer points.

GC-RANSAC algorithm combines the RANSAC image registration algorithm with the robust energy optimization algorithm Graph Cuts. A Gaussian kernel function is introduced to describe carefully the matching degree of points and the model. And according to the principle of spatial consistency, the strength of the fitted model is measured by the penalty function. The detector can improve the accuracy of feature extraction and calculation speed.

In order to ensure the existence of the targets tracked in the key frames, new key frames will be inserted directly when the number of the feature points falls below the threshold. And when the mean parallax, between the current frame and the latest key frame, exceeds a certain threshold, the new frame will be kept as a new key frame.

3.2. Visual initialization
In visual Initialization, the pose of the camera is obtained by the projective geometry method,[1]-[2]. The relative rotation and scale translation between the adjacent key frames are calculated by the five-point method. Then, after the features of adjacent key frames are triangulated, the camera pose is estimated by PnP. Finally, the result is optimized furthermore by the Bundle Adjustment to minimize the reprojection error.

4. IMU pre-integration
Since the large difference of the sampling frequency between the IMU and the camera, IMU pre-integration is used to get IMU information between two adjacent key frames.

Suppose that, the actual values of acceleration and rotation angle at time $t$ are $\alpha_t$ and $\omega_t$. The value measured by accelerometer and gyroscope are $\hat{\alpha}_t$ and $\hat{\omega}_t$. The relationship between them shown as
\[ \begin{align*}
\dot{a}_t &= a_t + b_{at} + R_w^b g^w + n_a \\
\dot{\omega}_t &= \omega_t + b_{\omega t} + n_\omega
\end{align*} \] (2)

Where, \( R_w^b g^w \) is the value of gravity acceleration in the IMU coordinate system. \( b_{at} \) and \( b_{\omega t} \) are the offsets of acceleration and the gyroscope at time \( t \), following the random walk model. \( n_a \) and \( n_\omega \) are Gaussian Noise, \( n_a \sim N(0, \sigma_a^2) \), \( n_\omega \sim N(0, \sigma_\omega^2) \).

The position \( p_{k+1} \), speed \( v_{k+1} \), and rotation \( q_{k+1} \) of the IMU at \( t_{k+1} \) can be calculated iteratively by the values at \( t_k \) and the increments calculated by IMU sampling data between adjacent key frames.

\[
\begin{align*}
p_{k+1} &= p_k + v_k \Delta t - \frac{1}{2} g \Delta t^2 + \Delta a \\
v_{k+1} &= v_k - g \Delta t + \Delta \beta \\
q_{k+1} &= \Delta \gamma
\end{align*} \] (3)

where \( \Delta t = t_{k+1} - t_k \). \( \Delta a, \Delta \beta, \Delta \gamma \) are the increments of the position, speed and rotation angle accordingly. Considering the formula (2) and the transformation from the world coordinate system to the IMU one \( R_w^b \) and \( q_w^b \), \( \Delta a, \Delta \beta, \Delta \gamma \) can be calculated as below.

\[
\begin{align*}
\Delta a &= \int R_w^b (\dot{a}_t - b_{at} - n_a) dt^2 \\
\Delta \beta &= \int R_w^b (\dot{\omega}_t - b_{\omega t} - n_\omega) dt \\
\Delta \gamma &= \int \frac{1}{2} \Omega (\dot{\omega}_t - b_{\omega t} - n_\omega) q_w^b dt
\end{align*} \] (4)

where \( \Omega (\omega) \) represents the angular velocity matrix. And after first-order linear expansion, formula (4) is transformed into the discrete type.

\[
\begin{align*}
\Delta a &\approx \Delta \dot{a} + J_\omega^b \delta b_{ak} + J_\omega^b \delta b_{\omega k} \\
\Delta \beta &\approx \Delta \dot{\beta} + J_\omega^b \delta b_{ak} + J_\omega^b \delta b_{\omega k} \\
\Delta \gamma &\approx \Delta \dot{\gamma} \otimes \left[ \begin{array}{c} 1 \\ \frac{1}{2} J_\omega^b \delta b_{ak} \end{array} \right]
\end{align*} \] (5)

where \( \Delta \dot{a}, \Delta \dot{\beta}, \Delta \dot{\gamma} \) are the measurement information of the increments of position, velocity and rotation angle of the IMU, which can be calculated directly by the measured values. Considering its small value, the offset can be ignored temporally [8], and will be calibrated later by the offset errors \( \delta b_{ak} \) and \( \delta b_{\omega k} \) during joint initialization. The noise is zero. \( J_\omega^a \) and \( J_\omega^\beta \) represent the residual Jacobis.

The measurement information \( \Delta \dot{a}_{k+1}, \Delta \dot{\beta}_{k+1}, \) and \( \Delta \dot{\gamma}_{k+1} \) at time \( t_{k+1} \) can be iteratively calculated from the measurement information at \( t_k \). In order to increase the accuracy, Runge-Kutta method is applied and 4 time points are extracted between two adjacent keyframes for integration (such as (6)).

\[
\begin{align*}
\Delta \dot{a}_{k+1} &= \Delta \dot{a}_k + \Delta \dot{\beta}_k \Delta t + \frac{\Delta t^2}{12} (k_1 + 2k_2 + 2k_3 + k_4) \\
\Delta \dot{\beta}_{k+1} &= \Delta \dot{\beta}_k + \frac{\Delta t}{12} (k_1 + 2k_2 + 2k_3 + k_4) \\
\Delta \dot{\gamma}_{k+1} &= \Delta \dot{\gamma}_k \otimes \left[ \begin{array}{c} 1 \\ \frac{\Delta t}{12} (l_1 + l_2 + l_3 + l_4) \end{array} \right]
\end{align*} \] (6)

As in formula (2), \( a_t = \dot{a}_t - b_{at}, \omega_t = \dot{\omega}_t - b_{\omega t} \), while the noise is zero. So \( k_1 = \Delta \dot{\beta}_k a_{tk}, k_2 = \Delta \dot{\gamma}_k \frac{\Delta t}{12} a_{tk}, k_3 = \Delta \dot{\gamma}_k \frac{\Delta t}{12} a_{tk} + k_1 \frac{\Delta t}{2}, k_4 = \Delta \dot{\gamma}_k a_{tk+1} + k_2 \frac{\Delta t}{2} \) and \( l_1 = \omega_{tk}, l_2 = \omega_{tk} \frac{\Delta t}{2} + l_1 \frac{\Delta t}{2}, l_3 = \omega_{tk} \frac{\Delta t}{2} + l_2 \frac{\Delta t}{2}, l_4 = \omega_{tk+1} \) + l_3 \Delta t.

5. Visual-inertia joint initialization

The joint initialization of Vision and IMU is to calibrate the offsets of accelerometer and gyroscope, and to optimize the direction of gravity. In the process of the joint initialization and the backend optimization below, the residual error will be solved by applying the least square method. Instead of Gauss Newton, Dog-Leg is used to solve the nonlinear least square problem, which means higher accuracy.
Assume that the world coordinate system coincides with the initial camera coordinate system. At the k-th key frame, the pose of the camera obtained by visual initialization is \((p_{c_k}^w, q_{c_k}^w)\) in the world coordinate system and \((p_{c_k}^{bk}, q_{c_k}^{bk})\) in the IMU coordinates.

The calibration of the IMU bias of the gyroscope, between two successive key frames in the sliding window, can be transformed into a least-squares problem (7)

\[
\min_{\delta b_{wk}} \sum_{k\in[1,N]} \left\| q_{c_k}^{bk} - \otimes q_{c_k}^{bk} \otimes \Delta y_{IMU} \right\|^2
\]

(7)

Where, \(p\) and \(q\) are the visual pose. And the IMU state \(\Delta y_{IMU} = \Delta \dot{y}_{k+1} \otimes \left[ \frac{1}{2} J_{b_w} \delta \omega \right] \) is the residual Jacobian of the gyroscope, and \(N\) is the number of key frames.

The accelerometer needs to be initialized and calibrated. The vector to be calibrated shows below:

\[
\chi_{IMU} = \left[ v_{b_k}^w, v_{b_k}^b, \ldots, v_{b_k}^n, g^c \right]
\]

where, \(v_{b_k}^i\) is the speed of the i-th keyframe image in the IMU coordinate system at time \(t_k\) and \(g^c\) is the gravity vector in the camera coordinate system.

In formula (3), the pose of IMU that obtained by IMU pre-integration, is substituted by the camera pose \(p_{c_k}^w, p_{c_k+1}^w\) that obtained by visual initialization. The increments of position and velocity are shown in (8).

\[
\Delta \alpha' = R_w^b \left( p_{c_k+1}^w - p_{c_k}^w + \frac{1}{2} g^w \Delta t^2 - v_{c_k}^w \Delta t \right)
\]

\[
\Delta \beta' = R_w^b \left( v_{c_k+1}^w - v_{c_k}^w + g^w \Delta t \right)
\]

(8)

Then the linear measurement model of \(\chi_{IMU}\) is expressed as (9).

\[
\tilde{x}_{k+1}^{bk} \approx \begin{bmatrix} \Delta \alpha' + p_{c_k}^w + p_{c_k}^{bk} \\ \Delta \beta' \end{bmatrix} = H_{k+1} \chi_{IMU}
\]

(9)

where \(\tilde{x}_{k+1}^{bk}\) is the observed values of the position and velocity after joint initialization, and \(H_{k+1}\) is the Jacobian matrix after IMU and visual fusion. Finally, the initial calibration of the accelerometer is transformed into a least squares problem.

\[
\min_{\chi_{IMU}} \sum_{k\in[1,N]} \left\| \tilde{x}_{k+1}^{bk} - H_{k+1} \chi_{IMU} \right\|^2
\]

(10)

In order to increase the robustness of the system, gravity must be optimized. The gravity is generally set to 9.8 \(\text{m/s}^2\), and the direction of the acceleration of gravity can be optimized only. The pair of orthogonal bases \(b_1\) and \(b_2\) of the tangent space of the gravity vector are used to fine-tune the direction of gravity.

\[
\hat{g} = \| g \| \tilde{g} + w_1 b_1 + w_2 b_2
\]

(11)

where, \(w_1\) and \(w_2\) are control parameters [4], and \(\tilde{g}\) is the direction vector of the gravity in the camera coordinate system.

Finally, the calibrated offsets \(\delta b_{ak}, \delta \omega_{ak}\) the calibrated gravity \(g\) are used to update the accelerometer and gyroscope as above.

6. Backend optimization

Backend optimization performs non-linear optimization on the information, such as the rotation angle and translation of the camera, the position, speed, rotation angle of the IMU, and the offsets of the IMU, and their corresponding errors, to get the final estimated pose.

6.1. State vector and objective function

Backend optimization is performed in a sliding window, and the state vector is defined as:

\[
\chi = [x_0, x_1, \ldots, x_n, x_0^b, \lambda_0, \lambda_1, \ldots, \lambda_n]
\]

in which , \(x_k = [p_k^w, v_k^w, q_k^w, b_{ak}, b_{ak}, \omega_{ak}]\), \(k \in [0, N]\) represents the position, speed, rotation angle, and the offset information. \(x^b_k = [p_k^b, q_k^b]\) denotes the transformative parameter between the camera detector and the IMU detector. \(\lambda_i\) is the inverse depth information of the feature.
In backend optimization, the information of the marginalized prior information, IMU measurement residuals and visual measurement residuals is taking into account, and the optimization goal of the least squares problem is defined as

$$\min_{\chi} \left\{ \left\| r_p - H_p \right\|^2 + \sum_{k \in \mathcal{B}} \left\| r_p(\chi_{k+1}^{bk}) - r_p(\chi_{k+1}) \right\|_{\Lambda_{k+1}^{k}}^2 + \sum_{(i,j) \in \mathcal{E}} \left\| r_c(\chi_{i}^{c}), \chi \right\|^2 \right\} \tag{12}$$

Where $\Lambda_{k+1}^{k}$ is the covariance matrix of the IMU pre-integrated noise term, $\Lambda_{i}^{j}$ is the covariance matrix of visual observation noise, and $r_{p}(\chi_{k+1}^{bk}, \chi) = [\delta p_{k}^{w} \delta v_{k}^{w} \delta q_{k}^{w} \delta b_{a_k} \delta b_{w_k}]^T$ indicates the position, velocity and rotation angle of the IMU, with the measurement residuals of offsets for accelerometer and gyroscope. $r_{c}(\chi_{i}^{c}, \chi) = [\delta p_{c_k}^{c} \delta q_{c_k}^{c}]^T$ represents the measurement residual of the camera position and rotation angle, which relevant to the inverse depth information. $\{r_{p}, H_{p}\}$ is marginalized prior information [9], for preserving the visual and IMU pose information of the frames that slide out of the sliding window.

In backend optimization, the least squares method is constructed about the marginalized prior information, the measurement residuals of vision and IMU, to obtain the optimized pose information.

6.2. Marginalization

Marginalization is an information update strategy, when the SLAM system continues to explore new environments and the sliding window moves on. It is necessary to limit the amount of information in the optimization process and remain the useful information as well.

During the Marginalization process, the parameters of frames that slide out of the window are no longer optimized, but they are still the constraints that influence the data inside the sliding window. Deleting directly the information of the frames that move out, will cause the lost of the constraints. Here, the constraints information is transformed into the prior distribution of variables to be optimized and added to nonlinear optimization as part of the error.

Furthermore, it needs to be determined, whether the next new frame of the sliding window is a key frame. If next new frame is a key frame, the oldest frame is marginalized, otherwise the new one is directly marginalized. The marginalization process is shown in Figure 3.

![Figure 3](Image)

**Figure 3.** In the figure, at time $t = k$, the state quantity in the system is characterized by $L_t$, i.e. $[1, 4]$, camera pose $\xi_j$, $j \in [1, 3]$, and at $t = k + 1$, the new camera pose $\xi_k$ is added, the earliest camera pose $\xi_1$ is removed, and the information of feature $L_k$ is retained to form a priori information to be added to the back-end optimization.

7. Experiment

7.1. Experiment of Public dataset

The EUROC vision-inertial data set was used to evaluate the VI-SLAM algorithm developed above and verify its reliability. The EUROC data set, including stereo images and synchronized IMU measurements, was collected by a micro-aircraft. The results will be evaluated and compared with the actual ground state information (https://github.com/MichaelGrupp/evo).

The four sequences in EURCO (MH_01_easy, MH_03_medium, V1_02_medium, V2_02_medium) are selected. The flight trajectory obtained by VI-SLAM is shown in Figure 4, where A is the result of MH_01_easy experiment, B is MH_03_medium, C is V1_02_medium, and D is V2_02_medium.
The calculated trace is illustrated as the solid line, and the real trace is dashed line. As shown in the figure, the calculated trajectories are very close to the real ones. The trajectory error along the time axis is shown in figure 5. And the statistic values, such as standard deviation (std), root error (rmse), maximum error (max) and minimum error (min) are listed in Table 1.

| Image sequence      | std  | rmse  | max   | min   |
|---------------------|------|-------|-------|-------|
| MH_01_easy          | 0.041| 0.094 | 0.232 | 0.009 |
| MH_03_medium        | 0.031| 0.070 | 0.161 | 0.006 |
| V1_02_medium        | 0.039| 0.070 | 0.267 | 0.014 |
| V2_02_medium        | 0.088| 0.168 | 0.385 | 0.012 |

Because the drone moves violently, the error in the test V2_02_medium is large and distributed uniformly. The maximum error is 0.385m and the root mean square of error is 0.168m. Because the aircraft moved relatively violently in the starting stage of test V1_02_medium, the error fluctuations occurred at the beginning of the sequence, with a maximum error of 0.267m. Due to the stable movement in the later period, the error was relatively small. For MH_01_easy and MH_03_medium sequences, the maximum error is between 0.16m and 0.23m, and the accuracy of the results is relatively high. In summary, the accuracy and robustness of SLAM are guaranteed by integrating the IMU information.

7.2. Experiment in real environment

In order to further verify the actual effect of the algorithm, the RealSense D435i camera introduced by Intel was used to carry out mapping and positioning experiments. The sensor is a depth camera equipped with an IMU unit (see Figure 6). The indoor environment of laboratory and corridor were selected for testing. The results of feature extraction(left) and pose estimation (right) are shown in Figure 7.

Because the drone moves violently, the error in the test V2_02_medium is large and distributed uniformly. The maximum error is 0.385m and the root mean square of error is 0.168m. Because the aircraft moved relatively violently in the starting stage of test V1_02_medium, the error fluctuations occurred at the beginning of the sequence, with a maximum error of 0.267m. Due to the stable movement in the later period, the error was relatively small. For MH_01_easy and MH_03_medium sequences, the maximum error is between 0.16m and 0.23m, and the accuracy of the results is relatively high. In summary, the accuracy and robustness of SLAM are guaranteed by integrating the IMU information.

7.2. Experiment in real environment

In order to further verify the actual effect of the algorithm, the RealSense D435i camera introduced by Intel was used to carry out mapping and positioning experiments. The sensor is a depth camera equipped with an IMU unit (see Figure 6). The indoor environment of laboratory and corridor were selected for testing. The results of feature extraction(left) and pose estimation (right) are shown in Figure 7.

The trajectory is smooth enough, except for the position in which the camera rotates violently. In the corridor environment, the error of the calculated trajectory is between 0.3m-0.5m, relative to the reference position set before. The proposed algorithm works well in reality.
8. Conclusion
An IMU-assisted tracking method, which based on the tightly coupling of visual-inertia, is proposed to improve the robustness of positioning. It combines IMU state information with the pose information extracted from visual tracking. And the SLAM public data set and real scene are used to verify the accuracy of the IMU-assisted tracking method. Furthermore, closed-loop detection can be applied to improve positioning accuracy and robustness in our future research.

References
[1] Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós J 2015 ORB-SLAM: A Versatile and Accurate Monocular SLAM System, vol 31 IEEE Transactions on Robotics pp 1147-1163
[2] Raúl Mur-Artal and Juan D. Tardós J 2017 ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, vol 33 IEEE Transactions on Robotics pp 1255-1262
[3] J. Engel, V. Koltun, and D. Cremers J 2017 Direct sparse odometry, vol 40 IEEE Transactions on Pattern Analysis and Machine Intelligence pp 611-625
[4] Mur-Artal R and Tardos J D J 2017 Visual-Inertial Monocular SLAM With Map Reuse , vol 2 IEEE Robotics and Automation Letters pp 796-803
[5] Forster C, Carlone L, Dellaert F J 2015 IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation, Georgia Institute of Technology pp 112-134.
[6] Tong Qin, Peiliang Li and Shaojie Shen J 2018 VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator, vol 34 IEEE Transactions on Robotics pp 1004-1020
[7] Dániel Baráth and Jiri Matas C 2018 Graph-Cut RANSAC, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVRP 2018)
[8] S. Shen, N. Michael, and V. Kumar J 2015 Tightly-coupled monocular visual inertial fusion for autonomous flight of rotorcraft MAVs, IEEE Int. Conf. on Robot. and Autom (ICRA 2015)
[9] G. Sibley, L. Matthies, and G. Sukhatme J 2010 Sliding window filter with application to planetary landing, vol 27 Field Robot pp 587-608