Advances in Engineering Research, volume 166
3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

Visual Inertial SLAM System Based on Embedded Parallel Processing

Jianyue Zhang[a)], Ge Li[b)] and Xuehe Zhang[c)]

School of Harbin Institute of Technology, Harbin 150000, China.

[a)]hitzhangjianyue@163.com
[b)]lige@hit.edu.cn
[c)]zhangxuehe@hit.edu.cn

Abstract. Based on the embedded computing platform, we propose a visual inertial fusion SLAM system. The software algorithm combines the matching of monocular visual feature recognition and IMU measurement pre-integration algorithms. And the back-end uses tightly-integrated nonlinear optimization algorithms to process data. The hardware uses NVIDIA's Jetson TX1 parallel processing computing platform and low-cost sensors to retrieve data. The robot uses the open source robot operating system ROS as the operating system and the upper computer uses the UBUNUT system. Based on the embedded parallel processing, visual inertial SLAM system has the characteristics of low cost and good stability.

Keywords: Visual Inertial, Parallel Processing, Nonlinear Optimization

INTRODUCTION

With the advancement of SLAM technology and the maturity of the positioning and map reconstruction theory, vision-based SLAM technology has developed rapidly. There are many mature SLAM solutions at home and abroad. Each of these solutions has its own applicable scenarios and application conditions. Visual SLAM solutions often do not have a certain degree of versatility. Therefore, depending on the specific application scenario and the requirements of low-cost, high-precision, dense construction, selecting the appropriate visual algorithm implementation and hardware conditions is the key to solving the SLAM problem.

At the beginning, Mono-SLAM is the first real-time monocular SLAM solution using the EKF optimization framework. Afterwards, the EKF framework became a common solution to the SLAM solution. Until the advent of PTAM, the nonlinear optimization framework was used for the solution of SLAM problem. The later appeared LSD-SLAM is a typical representative of monocular visual direct application. ORB-SLAM has a wide range of application scenarios. It uses an optimized framework for graph optimization and is applicable to visual sensors such as monocular, binocular, and RGB-D. It is currently a mature visual SLAM solution. As the vision scheme continues to mature, more visual SLAM solutions will be proposed. This is due to the fact that visual SLAM technology can reduce the cost of using laser radar, and has the advantages of rich feature information and wide application scenarios. However, the visual SLAM, especially the low-cost monocular vision, has the disadvantages of easily losing the tracking target and high computational performance requirements. At the same time, the optimization framework has been developing. Currently, nonlinear or graph optimization schemes rely heavily on computational capabilities. This has become a major obstacle to the in-depth development of SLAM technology.

In general, the future development trend of visual SLAM has two major categories: First, it is developing in the direction of low cost, lightweight, and miniaturization, allowing SLAM to operate well on small devices such as embedded devices or mobile phones, and occupying resources as little as possible. For example, to implement the

functions of robots, AR/VR devices. On the other hand, high-performance computing equipment is used to achieve precise 3D reconstruction and scene perception. In these applications, the goal is to accurately reconstruct the scene without limiting the portability of computing resources and devices. Therefore, GPUs can be used, and there is also a combination of this direction and deep learning. The combination of vision-based SLAM and embedded mobile computing devices to achieve real-time online positioning and mapping and semantic SLAM annotation is the current hot research direction.

VISUAL INERTIAL SLAM ALGORITHM

Tightly-Integrated Nonlinear Optimization

The sensor module mainly includes a monocular camera and an IMU. The monocular camera relies on the OPENCV library to read each frame of the image and publish it as a topic in the ROS. The IMU data is also published as a topic. The monocular camera uses a low-cost USB camera and the camera model is a common pinhole camera model, as shown in Figure 1.

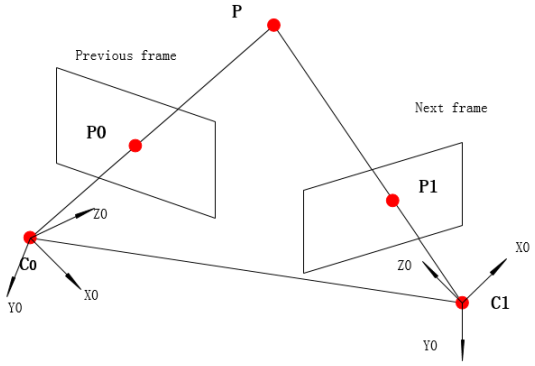

Figure 1 Pinhole camera model

This equation is called Epipolar Constraint. The geometric meaning of the polar constraint is that the mixed product of x1, t, and Rx0 is 0 indicating that the three vectors are coplanar. That is to say the three sides of the triangle in the above figure are coplanar. E is called the essential matrix, which is determined by the outer parameters R and t. The two-dimensional information from two frames can be obtained as an essential matrix and then decomposed into R and t. This is the new form of Epipolar Constraint.

$$x_1 T t \times R x_0 = 0 \tag{1}$$

The following formulas complete the conversion of the world coordinate system (in millimeters) to the pixel coordinate system (in pixels), the world coordinate system to the camera coordinate system, the camera coordinate system to the image coordinate system, and the image coordinate system to the pixel coordinate system.

$$Zc\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & uo \\ 0 & 1/dy & vo \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} Xw \\ Yw \\ Zw \\ 1 \end{bmatrix} = \begin{bmatrix} fx & 0 & cx & 0 \\ 0 & fy & cy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} Xw \\ Yw \\ Zw \\ 1 \end{bmatrix} = M1M1\bar{X}w = M\bar{X}w \tag{2}$$

The inertial attitude sensor integrates three-axis gyroscopes, three-axis accelerometers, three-axis geomagnetic sensors and data processing chips. The data processing chip embeds data fusion algorithms, including static error calibration, dynamic error estimation, and data fusion. The attitude and heading information of the measured object can be output in real time, and it is packaged in a PLCC (stamp hole) for easy testing and system integration. It is suitable for VR, handheld devices, wearable devices, motion capture, and indoor robot navigation.

The IMU output frequency is high (100-1KHz), so the optimization variable will grow rapidly, making real-time optimization difficult. Christian Forster proposed using a method of pre-integrating IMU sampling data between two frames of images into a constraint which reduces the optimization variables. For standard IMU integration between two frames of images, the initial state is given by the state estimate of the first frame. However, in each iteration of the optimization, the state estimation is changing, then the IMU pre-integration needs to be repeated. This problem can be solved by the constraint of relative motion and re-parameterization. Then the re-parameterization is called IMU pre-integration. IMUs generally have white noise and zero bias. However, the visual image is not biased when it is stationary, so the visual image is used to determine the zero bias, and the IMU is used to determine the

rotational motion and rapid movement. By re-parameterization, the IMU measurements between key frames are integrated into relative motion constraints, avoiding duplicate integration due to initial condition changes, and integrating the IMU data before the next key frame arrives.

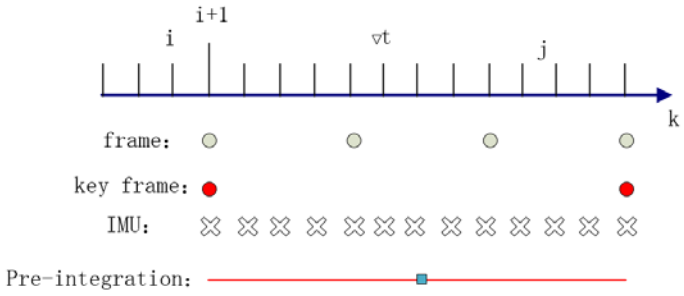

Figure 2 Camera and IMU different frame rates

The IMU's variable formula can be expressed as

$$\begin{cases} {}_{B}W'_{WB}(t) = {}_{B}W_{WB}(t) + b_g(t) + \eta_g(t) \\ {}_{B}a'(t) = R_{WB}^{T}(t)({}_{W}a(t) - {}_{W}g) + b_a(t) + \eta_a(t) \end{cases} \tag{3}$$

The measured angular velocity and acceleration are the sum of the actual angular velocity, acceleration, deviation and noise. The deviations of the IMU are random walks, and the noise is Gaussian. The IMU's kinematic formula is expressed in differential form.In the end, IMU's motion expression can be obtained, namely the difference equation:

$$\begin{cases} R(t+\Delta t) = R(t)\mathrm{Exp}\left((\tilde{\omega}(t) - b^g(t) - \eta^{gd}(t))\Delta t\right) \\ v(t+\Delta t) = v(t) + g\Delta t + R(t)(\tilde{a}(t) - b^a(t) - \eta^{ad}(t))\Delta t \\ p(t+\Delta t) = p(t) + v(t)\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R(t)(\tilde{a}(t) - b^a(t) - \eta^{ad}(t))\Delta t^2 \end{cases} \tag{4}$$

Therefore, the measurement model between two key frames is:

$$\begin{cases} \Delta R_{ij} \doteq R_i^T R_j = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\omega}_k - b_k^g - \eta_k^{gd}\right)\Delta t\right) \\ \Delta v_{ij} \doteq R_i^T(v_j - v_i - g\Delta t_{ij}) = \sum_{k=i}^{j-1} \Delta R_{ik}(\tilde{a}_k - b_k^a - \eta_k^{ad})\Delta t \\ \Delta p_{ij} \doteq R_i^T\left(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}\sum_{k=i}^{j-1} g\Delta t^2\right) \\ \quad = \sum_{k=i}^{j-1}\left[\Delta v_{ik}\Delta t + \frac{1}{2}\Delta R_{ik}(\tilde{a}_k - b_k^a - \eta_k^{ad})\Delta t^2\right] \end{cases} \tag{5}$$

Pose estimation is the key to robot navigation and positioning. Both 3D reconstruction and trajectory planning require highly accurate state metrics (position, velocity, direction, etc.). In this section, we present a visual inertial pose estimation based on monocular visual tight integration of sliding windows to provide accurate state estimation for the entire system. Since monocular visual INS fusion is a highly nonlinear process, good pose initialization is required before pose estimation. According to the principle of epipolar geometry and the IMU's pre-integration algorithm, the rotation and translation between two key frames can be accurately calculated and pose estimation can be performed. We will introduce data preprocessing, stable pose initialization and tight fusion optimization framework.

VINS can be divided into two categories according to optimization methods: one is a filtering-based optimization algorithm, and the other is a convex optimization-based optimization algorithm. Filter-based optimization algorithms are generally considered to be more efficient and faster to compute. The disadvantage of this algorithm is that it may lead to sub-optimal results in order to repair system nonlinearities. The optimization

method based on convex optimization can obtain better calculation results by reconstructing the past state of the linearization system. The cost is to spend more computing resources.

The IMU and camera data are collected at certain time intervals for pose estimation. Due to its multi-view constraints guarantee more accurate accuracy. As shown in Figure 5, the fixed window contains data for several different frames of the IMU and camera. In the slip window we define the following vector:

$$\begin{cases} \chi=[x_0, x_1, ...x_n, x_c^b, \lambda_0, \lambda_1...\lambda_m] \\ x_k=[p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a^b, b_g^b], k\in[0,n] \\ x_c^b=[p_c^b, q_c^b] \end{cases} \tag{6}$$

Where denotes the state of the k-th keyframe, including the accelerometer and gyroscope deviations in the world coordinate system's position, velocity, angle, and principal coordinate system. We use the IMU data as the main coordinate system, the world coordinate system is associated with the gravity-related real world, and the gravity vector is set after processing in the initialization process. Throughout the estimation framework, we use quaternions to indicate optional equipment. Indicates external parameters, including rotation and translation between camera and IMU positions. Is the number of key frames in the sliding window. Refers to the inverse depth of the first feature in the observation volume.

There are two types of metrics in our state estimation framework: one is an image and the other is IMU data. Both measurements were pre-processed prior to optimization. The image was tracked and characterized, and the IMU performed pre-integral processing. Deviations were taken into account in the IMU pre-integration process and later in the optimization process. For each frame of image, there is a KLT sparse optical flow algorithm (Lucas and Lucas) tracking, and corner features are detected simultaneously to maintain the minimum number of feature points (100-300) in each frame of the image. The detector enforces uniform feature distribution by setting a minimum interval of 20-30 pixels between two adjacent features. In the basic matrix verification, the RANSAC step simply performs outlier exclusion. At the same time select key frames at this step. There are two criteria for keyframe selection, one of which is the average disparity. If the average disparity of the tracked feature exceeds a certain threshold, we treat the image as a keyframe. Both translation and rotation cause parallax, but purely rotated features cannot be triangulated. To avoid this, we use IMU data results to compensate for rotation when calculating disparity. Another standard is tracking quality. If the number of tracking features is below a certain threshold, we also treat new frames as key frames. We define the IMU measurement data (angular velocity and acceleration) as and the measurement result is related to deviation and noise.

After initialization, we continue to use the slip window nonlinear estimation for high-precision fusion optimization, taking into account the IMU bias. The largest posterior estimate has been obtained by minimizing the sum of the covariance norm of all measured-quantity residuals:

$$\min\left\{\left\|\gamma_p - H_p\chi\right\|^2 + \sum_{k\in\beta}\left\|\gamma_\beta\left(\hat{z}_{b_{k+1}}^{b_k}, \chi\right)\right\|_{p_{b_{k+1}}^{b_k}}^2 + \sum_{(l,j)\in C}\left\|\gamma_c\left(\hat{z}_l^{c_j}, \chi\right)\right\|_{p_l^{c_j}}^2\right\} \tag{7}$$

$\gamma_\beta\left(\hat{z}_{b_{k+1}}^{b_k}, \chi\right)$、$\gamma_c\left(\hat{z}_l^{c_j}, \chi\right)$are the residuals measured by the IMU and camera, respectively. β is the measurement of IMU and C is the measurement of feature detection. The nonlinear system is linearized by the least deviation and solved by the Newton Gauss method. The measurement model of the IMU and the measurement model of the camera have been previously described in detail.

The IMU measurement frequency is much higher than the visual measurement. The frequency of our nonlinear optimization estimation is limited by the visual measurement. In order to facilitate the performance of real-time control, the estimated output is directly fused with the latest IMU measurements, which are used as high-frequency feedback in the control loop.

GPU Parallel Processing Map Reconstruction

The selection of key frames for dense construction is more stringent than the key frame selection in pose estimation. We use two thresholds to control the selection of key frames: The first one is used to exclude frames that do not contain enough depth information. If the calculated distance is less than the threshold, we only save it. The second threshold is the key frame selection parameterized to accommodate the most recent sample layer setting.

From experience, this threshold setting is guaranteed to be large enough to ensure that there is enough depth information for feature recovery and not too large to avoid the depth of the most recent sampling layer being invisible.

For each key frame, multiple test frames are required for deep updates. There is only one key frame at any time, and the key frame is switched by a distance metric. Only when a frame changes relative to the current key frame by more than one distance threshold, the frame is regarded as a new key frame. We sample the depth of each pixel in the key frame to obtain multiple virtual planes with different depth values. These planes are all perpendicular to the z axis in the primary coordinate system. Each pixel in the key frame is projected onto a virtual plane and back projected into the test frame. By calculating the depth difference, a similar cost is obtained for each possible depth of each pixel. Collect all the costs and focus on a 3D cost block.

When there are multiple test frame images, a cost block is obtained for each image. These cost blocks are all based on the same reference picture and can be aggregated into a single cost block to reduce their sensitivity to noise. This process can be accelerated with GPU parallel implementations. Intuitively, for each pixel, its true depth is the depth corresponding to the smallest similar cost in the centralized cost block. After that, the semi-global smoothing optimization is applied to remove the outliers and the weakly-textured region is interpolated again. The final step of the deep optimization uses a parabolic fit.

EXPERIMENTAL VERIFICATION RESULTS

The aircraft continues to move around the square. The visual interface shows that the robot's motion track is square and accurate. This is due to the effect of the loop detection in the algorithm. The number of adjacent frame feature matching feature points can be seen to be not very large. It can be understood that the number of corner detections is affected by the image quality, but it is also sufficient to use the PNP algorithm to obtain the pose after matching.

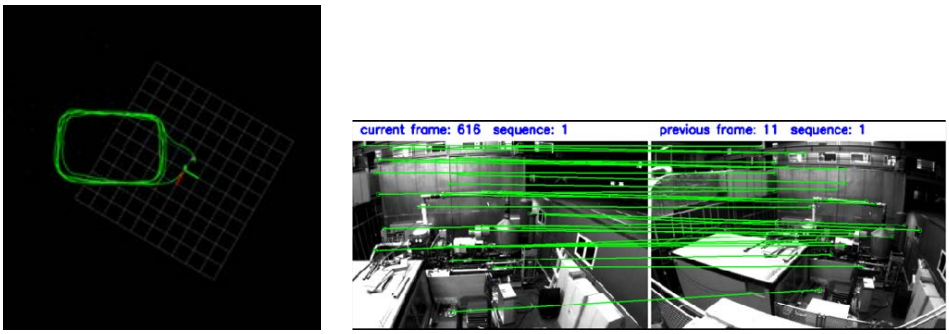

Figure 3 Value estimation and feature matching

This is a rendering of a GPU-accelerated monocular dense map reconstruction. Real scenes and color visual maps and grid maps based on the PCL point cloud library all have some correspondence. The effect is good.

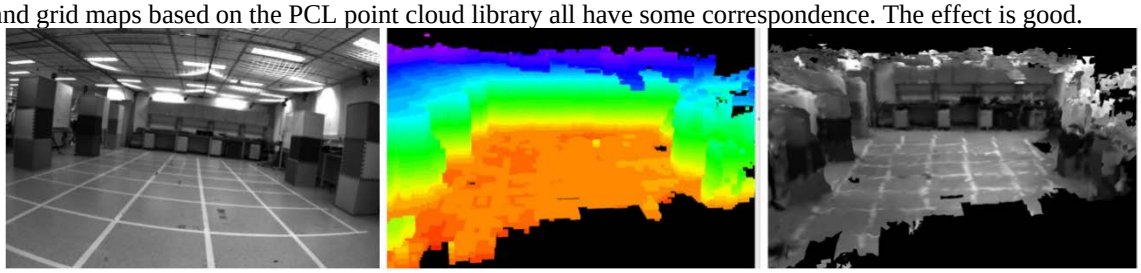

Figure 4 Dense map reconstruction

CONCLUSION

The pose and pose estimation algorithms for fusion vision and IMU data include vision, IMU data pre-processing, initialization, tight coupling optimization, loopback detection, etc. The robot's motion trajectory is displayed in real time. Under the assurance of accuracy of pose estimation. Through pixel-level triangulation, outlier removal, and deep fusion per frame, the PCL library is used to acquire dense 3D maps. Using GPU acceleration for

the reconstruction of the map, powerful computing capabilities ensure data processing accuracy and real-time performance.

ACKNOWLEDGMENTS

This work was done under the guidance of Li Ge, a teacher at the Institute of Robotics at the HIT. Thank you very much to Li Ge and Zhang Xuehe for their help. Finally, I would like to thank the National Natural Science Foundation of China (nos. 61473104) for funding.

REFERENCES

1. Patrik Schmuck, Margarita Chli. Multi-UAV Collaborative Monocular SLAM [J]. IEEE International Conference on Robotics and Automation, Singapore.2017: 1-3.
2. Ahmed Hussein,Pablo Mar´ın-Plaza,David Mart´ın, et al.Autonomous Off-Road Navigation using Stereo-Vision and Laser-Rangefinder Fusion for Outdoor Obstacles Detection [J]. Intelligent Vehicles Symposium, 2016: 2-4.
3. G Dissanayake,H Durrant-Whyte, T Bailey,et al. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem [J]. Robotics & Automation IEEE Transactions on,2001,17 (3) :229 - 241
4. DH Won,S Chun,S Sung,YJ Lee,et al. INS/vSLAM system using distributed particle filter [J]. International Journal of Control Automation,2010 , 8 (6) :1232-1240
5. G Zhou ,J Ye,W Ren,T Wang,Z Li. On-board inertial-assisted visual odometer on an embedded system[J].IEEE International Conference on Robotics & Automation .2014 :2602-2608
6. JS Hu,MY Chen.A sliding-window visual-IMU odometer based on tri-focal tensor geometry[J]. Microsystem Technologies.2014 :3963-3968
7. RO Castle, G Klein, DW Murray. Combining monoSLAM with object recognition for scene augmentation using a wearable camera[J]. Image & Vision Computing,2010,28 (11) :1548-1556
8. MAAAtashgah,SMB Malaek. An integrated virtual environment for feasibility studies and implementation of aerial MonoSLAM [J],Virtual Reality,2012 ,16 (3) :215-232
9. K Konolige,M Agrawal. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping[J]. IEEE Transactions on Robotics,2008 , 24 (5) :1066-1077
10. C Estrada,J Neira,JD Tardos. Hierarchical SLAM: real-time accurate mapping of large environments[J]. IEEE Transactions on Robotics, 2005 , 21 (4) :588-596
11. AJ Davison, ID Reid, ND Molton, O Stasse.MonoSLAM: real-time single camera SLAM[J].IEEE Transactions on Pattern Analysis & Machine Intelligence2007 , 29 (6) :1052
12. G Klein,D Murray.Parallel Tracking and Mapping for Small AR Workspaces,IEEE &Acm International Symposium on Mixed & Augmented Reality2008 :1-10
13. Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. IEEE, 2011: 2564-2571.
14. J Engel,T Schöps,LSD-SLAM: Large-Scale Direct Monocular SLAM [C]. ASpringer International Publishing , 2014 , 8690 :834-849.
15. XH Wang,PF Li. Improved Data Association Method in Binocular Vision-SLAM [J]. International Conference on Intelligent Computation Technology & Automation2010 , 2 :502-505
16. Corke P I, Good M C. Dynamic effects in visual closed-loop systems[J]. IEEETransactions on Robotics & Automation, 1996, 12(5):671-683
17. Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]// IEEE International Symposium on Mixed and AugmentedReality. IEEE, 2011:127-136.
18. H Zhang , Y Liu , J Tan. Loop Closing Detection in RGB-D SLAM Combining Appearance and Geometric Constraints [J]. Sensors, 2015 , 15 (6) :14639-60
19. R Mur-Artal, JD Tardós,et al. Visual-Inertial Monocular SLAM With Map Reuse[J]. IEEE Robotics & Automation Letters2016 , 2 (2) :796-803
20. J Kelly, GS Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration [J]. IEEE International Symposium on Computational Intelligence in Robotics & Automation，2009 , 30 (1) :360-368

21. T Bailey , J Nieto , J Guivant, M Stevens. Consistency of the EKF-SLAM Algorithm [C].IEEE/RSJ International Conference on Intelligent Robots & Systems2006 :3562-3568
22. Yi Lin ,Fei Gao ,Tong Qin,et al.Autonomous Aerial Navigation UsingMonocular Visual-Inertial Fusion [J]. Journal of Field Robotics,2017