A Robust Visual-Inertial Odometry Based on RANSAC Modeling and Motion Conflict in Dynamic Environments

Shenglin Zhao [1,2], Haoyuan Cai [1] , Yaqian Liu [1,2] , Chengnie Liao [1,2] and Chunxiu Liu [1]

[1] State Key Laboratory of Transducer Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Haidian District, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, I00086, China

Abstract

Visual-inertial navigation systems (VINS) face challenges in highly dynamic environments. Current mainstream solutions filter dynamic objects based on the semantics of the object category. Such approaches require semantic classifiers to encompass all possibly-moving object classes, which makes them hard to scale and deploy. This paper proposes a dynamic feature point recognition method without advanced training. It utilizes the conflict information between IMU pre-integration and visual measurement to determine whether the RANSAC-modeled essential matrix locates in the static world or dynamic object. And it helps to filter out dynamic feature points when there is one primary moving object. We add a moving object in the visual field as interference and make an artificial dataset based on the EuRoC dataset. Experiments show that after adding the interference, the error of VINS-Mono increases by about 12 times, while our dynamic-VIO only increases by about 1.4 times. When VINS-Mono diverged due to interference, dynamic-VIO can remain robust. This method is capable of both dynamic objects moving in the environment and partially occluding.

Keywords

visual-inertial odometry, dynamic environment, dynamic points recognition, obstructed view

1. Introduction

Most typical visual SLAMs require a strong assumption that the scene is rigid [1]. The Visual Inertial Navigation System (VINS), which is widely applied to autonomous driving and micro MAVs [2, 3, 4], is usually more robust than visual SLAMs owing to its tightly coupled optimization with IMU participation. But it is still inseparable from the rigidity assumption. The visually dynamic environment is still a significant challenge.

The mainstream method uses semantic segmentation to identify and filter dynamic feature points. However, the semantics of the model trained by this supervised learning is limited. The dynamic interference in the actual environment is complex and changeable. In addition to rigid dynamic objects, there are flexible, dynamic interference and irregular shape visual field occlusion. In this case, maintaining robustness is our first demand. This paper presents a non-training-based robust VIO, which identifies and filters dynamic feature points based on RANSAC modeling when the estimated motion from inertial and visual information conflict. This method can keep robust when there is a single main dynamic object in the field of view.

2. Related Work

IPIN 2022 WiP Proceedings, September 5 – 7, 2022, Beijing, China
EMAIL: shurlinzhao@foxmail.com (A. 1); hycai@mail.ie.ac.cn (A. 2); liuyaqian20@mails.ucas.ac.cn (A. 3); liaochengnie21@mails.ucas.ac.cn (A. 4); cxliu@mail.ie.ac.cn (A. 5);
ORCID: 0000-0001-7949-8232 (A. 1); 0000-0001-8822-2756 (A. 2); 0000-0002-3589-4353 (A. 3); 0000-0002-5960-8710 (A. 4); 0000-0003-3718-9364 (A. 5)

CEUR Workshop Proceedings (CEUR-WS.org)

Most approaches use supervised deep learning methods to recognize moving objects and filter them. [5] proposes to learn the rigidity of a scene in a supervised manner from an extensive collection of dynamic scene data and directly infer a rigidity mask from two sequential images with depths. [6] proposes a generalized Hidden Markov Model (HMM) that formulates the tracking and management of the primary motion and the secondary motion as a single estimation problem and an epipolar constrained deep neural network that generates a per-pixel motion conflict probability map. It can distinguish the motion conflict of each road marking point at the pixel level. This method can not only identify the feature points belonging to the main motion but also calculate the motion of other dynamic objects to realize the simultaneous tracking of various motions.

Among the deep learning methods, different semantic segmentation methods (e.g., Mask R-CNN, SegNet) are trendy because of their scalability, intuitiveness, and good performance. [7] constructs an SSD object detector that combines prior knowledge to detect dynamic objects in the newly detection thread at the semantic level based on the convolutional neural network. They also proposed a missed detection compensation algorithm based on the speed invariance in adjacent frames to improve the recall rate of detection. [1] presents a real-time visual dynamic SLAM, RDS-SLAM, built on ORB-SLAM3 and adds a semantic thread and a semantic-based optimization thread for robust tracking and mapping in dynamic environments in real-time. The parallel thread enables the tracking thread to eliminate the wait for semantic information. [8] proposes a dynamic RGB-D SLAM based on semantic information and optical flow (DRSO-SLAM). They use the Mask R-CNN semantic segmentation network to obtain semantic information in indoor dynamic scenes and use the epipolar geometry to filter out the actual dynamic feature points.

However, the method based on pre-training is challenging to expand and deploy, which is also a problem that many researchers are aware of. The Autonomous Systems Lab, ETHZ, presents a novel end-to-end occupancy grid-based pipeline that can automatically label a wide variety of arbitrary dynamic objects. And it can thus generalize to different environments without the need for expensive manual labeling and, at the same time, avoids assumptions about the presence of a predefined set of known objects in the scene [9]. There are also some methods that are not based on deep learning. [10] regards structural regularities in the form of planes such as walls and ground surfaces as crucially static and presents a robust plane-based monocular VIO (RP-VIO), which also improves the robustness and accuracy in challenging dynamic environments. [11] adds adaptive multi-resolution range images and uses tightly-coupled lidar inertial odometry first to remove moving objects and then match lidar scan to the submap. This article also discusses how to make better use of IMU to make VIO more robust.

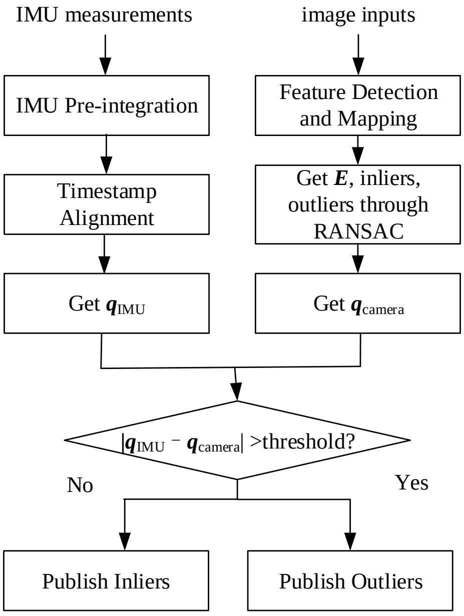

IMU measurements
image inputs
IMU Pre-integration
Feature Detection and Mapping
Timestamp Alignment
Get $\boldsymbol{E}$, inliers, outliers through RANSAC
Get $\boldsymbol{q}_{\mathrm{IMU}}$
Get $\boldsymbol{q}_{\mathrm{camera}}$
$|\boldsymbol{q}_{\mathrm{IMU}} - \boldsymbol{q}_{\mathrm{camera}}|$ >threshold?
No
Yes
Publish Inliers
Publish Outliers

Figure 1: The pipeline of the dynamic-VIO algorithm

3. Method

This work is based on VINS-Mono [12]. The main idea is when the motion perceived by IMU conflicts with the visual measurement, we choose to believe the IMU measurement. It does not mean using the measurement and calculation of IMU as the output but using this indication of IMU to reverse the selection range of feature points to provide more reliable features for the back end. Figure 1 shows the specific algorithm process. The IMU and camera obtain their measured motion information (rotation information is used here) through pre-integration and visual geometry, respectively, and then compare their differences. If the difference is slight, the feature tracker publishes feature points normally. If the difference can not be ignored, it selects the feature points inversely and publishes the outliers.

3.1. IMU pre-integration

The sampling rate of the camera is slower than that of IMU. Only when the camera has one sample coming in can the IMU integration be participated in backend optimization. So we can pre-integrate the IMU measurements over an image sampling period in the sensor coordinate system. This strategy saves a significant amount of computational resources [13]. Another significance of pre-integration is that IMU needs to calculate at the same time as the camera. Therefore, we need to save the IMU measurements in advance according to (1), (2), and (3) before that time point.

$$\boldsymbol{\alpha}_{b_i b_j} = \iint_{t\in[i,j]} (\boldsymbol{q}_{b_i b_t}\boldsymbol{a}^{b_t})\delta t^2, \tag{1}$$

$$\boldsymbol{\beta}_{b_i b_j} = \int_{t\in[i,j]} (\boldsymbol{q}_{b_i b_t}\boldsymbol{a}^{b_t})\delta t, \tag{2}$$

$$\boldsymbol{q}_{b_i b_j} = \int_{t\in[i,j]} \boldsymbol{q}_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega}^{b_t} \end{bmatrix} \delta t, \tag{3}$$

3.2. Timestamp Alignment between images and IMUs

The frequency of IMU is usually higher than that of the camera, and the sampling timestamps are not synchronized perfectly. As shown in Figure 2: Timestamps of IMU and camera sampling, there must be two IMU measurements at both ends of the sampling time tc of an image frame on the time axis. We need to interpolate the IMU measurements to align the timestamps. The acceleration $\boldsymbol{a}$ interpolates linearly using Linear intERPolation (LERP) as (4), (5), and the angular velocity $\boldsymbol{\omega}$ uses the Spherical Linear intERPolation (SLERP) as (6), (7). Considering the computational efficiency, we can use LERP to interpolate $\boldsymbol{\omega}$ in the program. That is because the sampling rate of the gyroscope is fast (200Hz), and the angle change can be minimal. As we can see later, we use IMU pre-integration only to decide the primary motion, and we do not need to calculate the accurate value of interpolation. As a result, the error of interpolation using LERP or SLERP can be ignored.

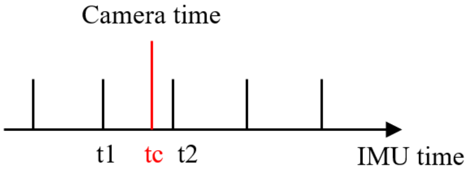


Figure 2: Timestamps of IMU and camera sampling

$$\boldsymbol{a}_{tc} = (1-\tau)\boldsymbol{a}_{t1} + \tau\boldsymbol{a}_{t2}, \tag{4}$$

$$\tau = \frac{t_2 - t}{t_2 - t_1}, \tag{5}$$

$$\boldsymbol{\omega}_{tc} = \frac{\sin[(1-\tau)\theta]}{sin\theta}\boldsymbol{\omega}_{t1} + \frac{\sin(\tau\theta)}{sin\theta}\boldsymbol{\omega}_{t2}, \tag{6}$$

$$\theta = \arccos(\boldsymbol{\omega}_{t1} \cdot \boldsymbol{\omega}_{t2}), \tag{7}$$

Then, we pre-integrate the acceleration and angular velocity at time tc and get the real pre-integration between the two frames. We use the pre-integrated quaternion $\boldsymbol{q}_{\mathrm{IMU}}$ for further determination.

3.3. Inliers and Outliers by RANSAC

We use feature point detection and matching to correlate two images. Once the two images match, we calculate the essential matrix $\boldsymbol{E}$ and decompose it by SVD. The essential matrix $\boldsymbol{E}$ is solved with Random Sample Consensus (RANSAC). RANSAC computes a model that matches the most points by repeatedly selecting a group of random subsets in the data, and the selected subsets are assumed to be inlier points. Then we get inlier points that match the final model and the outlier points that match not the model. Both inliers and outliers are essential to our algorithm, and the next section will explain why. After triangulation, we can get one unique set of $\boldsymbol{R}, \boldsymbol{T}$ from four solutions from SVD. At last, we transform the rotation matrix $\boldsymbol{R}$ into quaternion $\boldsymbol{q}_{\mathrm{camera}}$ for further determination.

3.4. Feature Points Publishing

Normally, element values of the quaternions obtained by the camera and the IMU are similar. If so, it means that the RANSAC algorithm has modeled the visually obtained essential matrix into the correct scene, which also means that there are no very prominent dynamic objects.

If this is not the case, that is, $\boldsymbol{q}_{\mathrm{IMU}}$ and $\boldsymbol{q}_{\mathrm{camera}}$ have a large difference, it means that RANSAC must have built the model on the incorrect object. Because the IMU measurements are always more reliable than the camera measurement under normal circumstances. At this point, most of the field of view is occluded by dynamic objects, and the outliers obtained by RANSAC are actually located in the static world. Under this circumstance, we should not publish the inlier points to the backend but the outlier points.

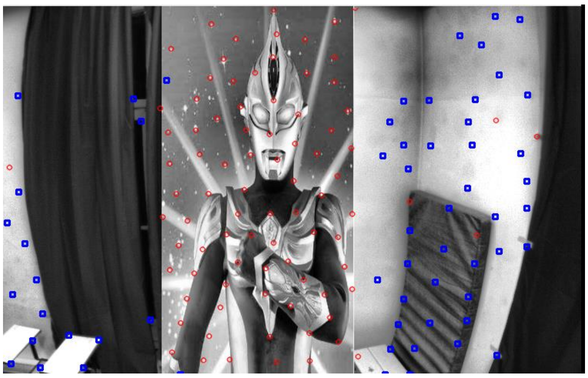


Figure 3: A dynamic points recognition example. Red points are recognized to be the dynamic points, and the blue points lie in the static world.

4. Experiment

We overlaid a leaf image over the original image based on the EuRoC dataset [14] and generated the new dataset called EuRoC_mask. The EuRoC ground truth is obtained from Leica Nova MS503 laser tracker and Vicon motion capture system. The length of the testing path is about 80m. These datasets simulated the process of the leaf gradually moving from the left end of the image to the right and then disappearing to the right end of the view, as shown in Figure 4: EuRoC dataset masked by a moving leaf. The experiment environments are Intel Core i5-9400F，6×2.90GHz, Ubuntu 16.04, and ROS Kinetic.

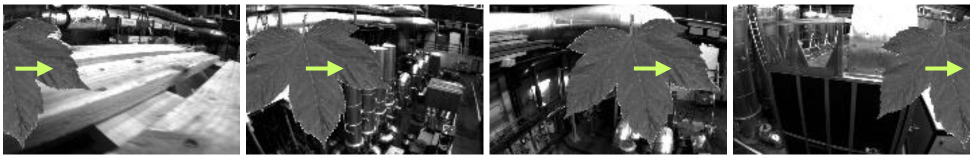


Figure 4: EuRoC dataset masked by a moving leaf

We run VINS-Mono (no-loop) and the dynamic-VIO (no-loop) proposed by this paper on these datasets. The number of feature points detected and tracked is set to 150. Table 1 shows the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE). When running on the masked datasets, the positioning error of VINS-Mono increases significantly, while dynamic-VIO can still keep the error in a small range.

Table 1
The positioning accuracy of the two algorithms on different datasets

Dataset	Absolute Trajectory Error RMSE (m)	
	VINS-Mono	Dynamic-VIO
MH01_easy_mask	1.563484	0.215349
MH03_medium_mask	3.482319	0.348051
MH05_difficult_mask	2.339641	0.335812
V1_02_medium_mask	2.03004	0.227272

In addition, a typical case is that when VINS diverges, as shown in Figure 5: VINS-Mono trajectory on V2_02_mask, dynamic-VIO can still maintain convergence, as shown in Figure 6: The positioning trajectory of VINS-Mono and dynamic-VIO..

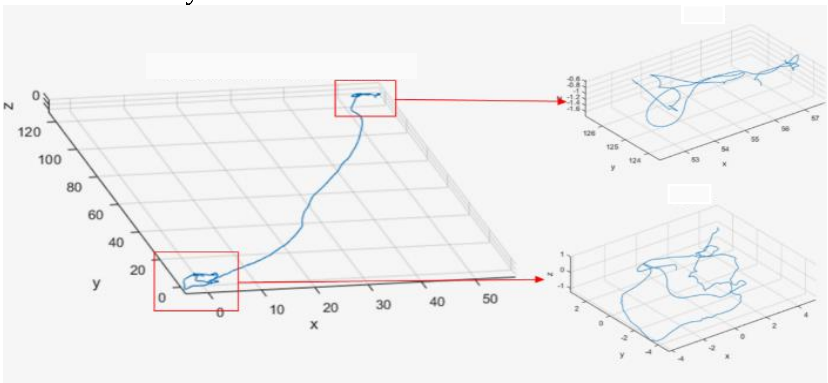
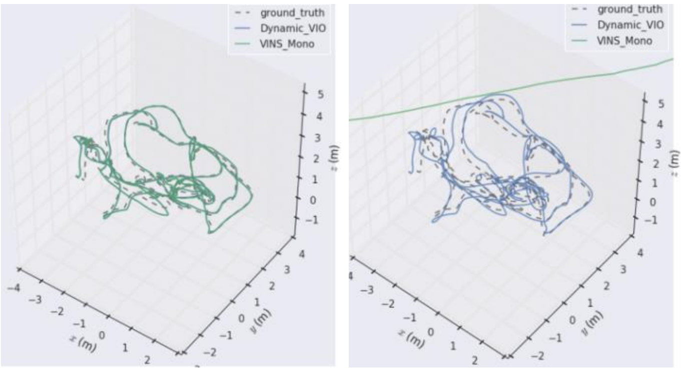


Figure 5: VINS-Mono trajectory on V2_02_mask



V2_02 **V2_02_mask**

Figure 6: The positioning trajectory of VINS-Mono and dynamic-VIO.

This research also plans to test flexible, dynamic objects and static visual field occlusion. Theoretically, this method should also be capable of flexible objects because the modeling process of RANSAC does not depend on object characteristics. As long as the motion sampled by IMU and camera is different, it is sufficient to identify the primary dynamic object in the static world. For static visual field occlusion, it depends on the degree of occlusion. For a small part of occlusion, it can be regarded as a particular case of this method, and this part of the occlusion area will be filtered out in each calculation period. If the occlusion part is too large, resulting in an insufficient number of feature points collected for a long time, this method will also fail.

However, the binary judgment between inertial information and visual information determines that this method can only recognize the largest dynamic object, which also comes from the reason that RANSAC selects the model satisfied by most feature points. If there are multiple dynamic objects in the environment and they are similar in size, this method will fail. In this case, this method will not do better than semantic segmentation-based approaches.

5. Conclusion

This paper proposed a robust VIO in dynamic environments. According to the principle of RANSAC modeling and the level of motion conflict, this algorithm flexibly selects inlier points and outlier points to publish, which can avoid the interference of the major dynamic object. Experiments on artificial datasets show that this method can keep robust and maintain the same level of trajectory error, while VINS-Mono should fail or get the positioning error increase significantly. Theoretically, this method should be applicable to rigid, flexible, dynamic objects and visual field occlusion. This method does not need data training and can be conveniently deployed when there is one major dynamic object, and high robustness is required.

6. Acknowledgments

This work is supported by the National Key Research and Development Plan (2020YFC2004501 and 2020YFC2004503), National Natural Science Foundation of China (NSFC) (Nos. 61774157 and 81771388), and Beijing Natural Science Foundation (No. 4182075).

7. References

[1] Y. Liu and J. Miura, “RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods,” IEEE Access, vol. 9, pp. 23772–23785, 2021.

[2] Z. Yu, L. Zhu, and G. Lu, “Tightly-coupled Fusion of VINS and Motion Constraint for Autonomous Vehicle,” IEEE Trans. Veh. Technol., pp. 1–1, 2022.

[3] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, “VINS on wheels,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore, May 2017, pp. 5155–5162.

[4] M. A. K. Gomaa, O. De Silva, G. K. I. Mann, and R. G. Gosine, “Observability-Constrained VINS for MAVs Using Interacting Multiple Model Algorithm,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 3, pp. 1423–1442, Jun. 2021.

[5] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz, “Learning Rigidity in Dynamic Scenes with a Moving Camera for 3D Motion Field Estimation,” in Computer Vision – ECCV 2018, Cham, 2018, vol. 11209, pp. 484–501.

[6] B. P. Wisely Babu, Z. Yan, M. Ye, and L. Ren, “On Exploiting Per-Pixel Motion Conflicts to Extract Secondary Motions,” in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, Oct. 2018, pp. 46–56.

[7] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, “Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment,” Robotics and Autonomous Systems, vol. 117, pp. 1–16, Jul. 2019.

[8] N. Yu, M. Gan, H. Yu, and K. Yang, “DRSO-SLAM: A Dynamic RGB-D SLAM Algorithm for Indoor Dynamic Scenes,” in 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, May 2021, pp. 1052–1058.

[9] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dube, R. Siegwart, and A. Cramariuc, “Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data,” p. 7.

[10] K. Ram, C. Kharyal, S. S. Harithas, and K. Madhava Krishna, “RP-VIO: Robust Plane-based Visual-Inertial Odometry for Dynamic Environments,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, Sep. 2021, pp. 9198–9205.

[11] “RF-LIO: Removal-First Tightly-coupled Lidar Inertial Odometry in High Dynamic Environments,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, Sep. 2021, pp. 4421–4428.

[12] 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,

[13] S. Shen, N. Michael, and V. Kumar, “Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs,” in 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May 2015, pp. 5303–5310.

[14] M. Burri et al., “The EuRoC micro aerial vehicle datasets,” The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.Wang, Xin, Tapani Ahonen, and Jari Nurmi. "Applying CDMA technique to network-on-chip." IEEE transactions on very large scale integration (VLSI) systems 15.10 (2007): 1091-1100.