Report of LARVIO on IROS 2020 FPV Drone Racing VIO Competition Dataset

Xiaochen Qiu[1*], Hongliang Huang, Gongyan Chen, Dapeng Wang and Hai Zhang
{qiuxiaochen, huanghl, chengongyan, wangdapeng, zhanghai}@buaa.edu.cn

***Abstract*—This report briefly introduces the monocular visual-inertial odometry LARVIO, and illustrates its performance on IROS 2020 FPV drone racing dataset.**

I. Introduction

LARVIO is short for Lightweight, Accurate and Robust monocular Visual Inertial Odometry, which is based on hybrid EKF VIO [1]. It is featured by augmenting features with long track length into the filter state of Multi-State Constraint Kalman Filter (MSCKF) [2] by One-Dimensional Inverse Depth Parametrization (1D IDP) to fully utilize their covisibility constraints to provide accurate positioning results. Loop closure is not applied in LARVIO. Two related paper are published [3], [4], and the source code is available at: https://github.com/PetWorm/LARVIO.

LARVIO is basically a member of MSCKF family. However, a difference from traditional MSCKF community is that LARVIO utilized Hamilton quaternion to derive all the formulas from scratch. The flow chart of LARVIO is shown in Figure 1.

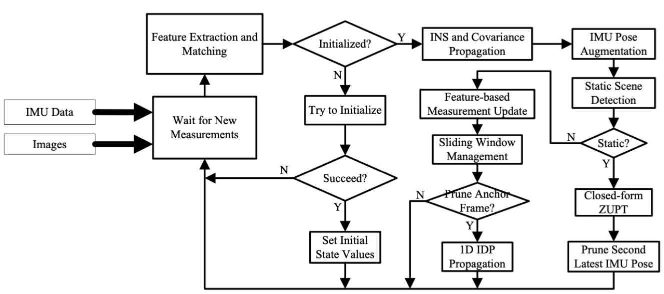


Fig. 1. Flow chart of LARVIO.

Several features are introduced into LARVIO to make it a ready-to-use VIO system:

- *Low cost map points augmentation* - A hybrid EKF architecture is utilized, which is inspired by the work of [1]. It augments features with long track length into the filter state of MSCKF. In LARVIO, 1D IDP is utilized to parametrize the augmented feature (or SLAM feature) state, which is different from the original 3D solution. This novelty improves the computational efficiency compare to the 3d solution. The positioning precision is also improved thanks to the utilization of complete constraints of features with long track length.
- *Online sensors calibration* - LARVIO is capable of online sensors calibration, including IMU intrinsic parameter, IMU-camera extrinsic parameter and timestamp error. All the online calibration functions are optional in configuration file.
- *Automatic initialization* - LARVIO can automatically initialize under either static or dynamic motion. For the static initialization, the static scene can be automatically detected through optical flow analysis, then pitch and roll angles will be recovered from static IMU data. For the dynamic initialization, we utilized the same method as in VINS-MONO [5], based on which we further take care of the timestamp misalignment between IMU and image data. LARVIO would initialize and enter the conventional filter routine as soon as one of the initialization methods succeed. This strategy makes LARVIO feasible to boot in both static and dynamic scenerios without manual intervention.
- *Zero velocity UPdaTe (ZUPT)* - ZUPT is introduced into LARVIO in a closed-form measurement update formulation. LARVIO would try to detect if the camera is static when reading new images. Once the static motion is detected, the ZUPT measurement update will carry out instead of feature based measurement update. This would help to constrain the drift caused by the biased rotation or IMU biases estimation, and through the ZUPT measurement update these biased states would be corrected.
- *Robust visual front-end* - Pyramidal Lucas-Kanade (LK) optical flow tracking [6] is utilized in LARVIO's visual front-end. Since the filter-based VIOs are more vulnerable to outliers compare to the optimization-based solutions, we proposed to utilize several methods trying to eliminate outliers and provide accurate feature tracking. Features are extracted by Shi-Tomasi's method [7], after the forward LK tracking, the inverse LK tracking would be carried out for checking, followed by an ORB descriptor filtering process [3] and finally the RANSAC process. Also, a refinement considering affine warp is utilized after the translation-only forward LK tracking in the edition we used for the competition.
- *Filter Consistency* - First Estimate Jacobian (FEJ) [8] is applied in LARVIO to maintain the filter consistency. This function can be turned on or off in configuration file.

The authors are with the Navigation and Embedded System Labortory from Beihang University, Haidian District, Beijing 100191, China.

II. Algorithm Configuration

Key parameters for LARVIO are kept the same for all sequences in UZH-FPV drone racing dataset [9]. The only exception is that we used a larger threshold to detect static motion in Indoor 45° sequences, however this would only affect the initialization procedure thus is trivial for the overall performance. As a monocular VIO solution, we used IMU data and left camera images from Snapdragon datasets to run LARVIO. The reported results are causal, we recorded the estimation after the measurement update at the first time an IMU pose appears instead of when it exits the sliding window.

A. Visual Front-End Setting

We used the implementation in OpenCV [10] to realize the feature extraction and tracking. Related parameters are listed in Table I. We applied a 2 level pyramidal LK tracker to track maximally 250 features with patch size of 15x15 pixels, the minimal distance between two features is the same as the patch size radius. We applied Contrast Limited Adaptive Histogram Equalization (CLAHE) [11] to equalize the images. The publish frequency parameter controls the frequency of new features extraction from new image and tracked features publication to filter back-end. As the motion in UZH-FPV dataset is highly dynamic, and also considering the computational efficiency of LARVIO, we proposed to use all images for feature extraction and publication. The image frequency reported in Snapdragon calibration file is 30Hz, in case of the timestamp oscillation, we set the publish frequency as 35Hz.

TABLE I
Visual Front-End Parameters Setting.

Parameter	Value
pyramid level	2
patch size radius	15
max iteration	30
track precision	0.01
max feature number	230
min features distance	15
apply CLAHE	true
publish frequency	35

In the edition used for the competition, we utilized affine warp refinement to refine the forward LK tracking results. We used the tracking results of the translation-only forward LK tracking and identity matrix as the initial affine warp, and refine the results at the raw image level. Considering 4 more additional parameters added to the optimization, we enlarge the patch size radius by 2 pixels in the refinement. We believe this would improve the tracking precision as drastic view point changes are more likely to happen under high dynamics as in UZH-FPV dataset, which would introduce nontrivial affine warp to the feature patches.

B. Filter Setting

The filter setting is listed in Table II. We used FEJ to maintain the consistency of the filter. The image plane is divided into 6x8 grids, and we allow 1 feature with the longest tracking length in each grid to be used as a SLAM feature. Also, only the features with tracking length larger than the maximal observation number can be augmented into the filter state, thus there will be 48 SLAM features at most for each time. We set the maximal observation number equal to the sliding window size as 10, and features with observation number smaller than 3 will not be triangularized and used in measurement update.

TABLE II
Filter Setting.

Parameter	Value
use FEJ	true
SLAM features number	48
sliding window size	10
min observation number	3
max observation number	10

C. Online Sensors Calibration

All the online sensors calibration functions, including the IMU-camera extrinsics and timestamp error and the IMU instrinsics, are turned on for UZH-FPV dataset. IMU-camera spatial and temporal calibration are usually turned on in VIO algorithms to improve the performance, while IMU instrinsic parameters need sufficient excition to be observable, which we found is hard to satisfy in common used dataset such as EuRoC [12]. However, we believe the motion in UZH-FPV dataset is dynamic enough to make the IMU instrinsic parameters observable.

III. Experiments And Evaluation

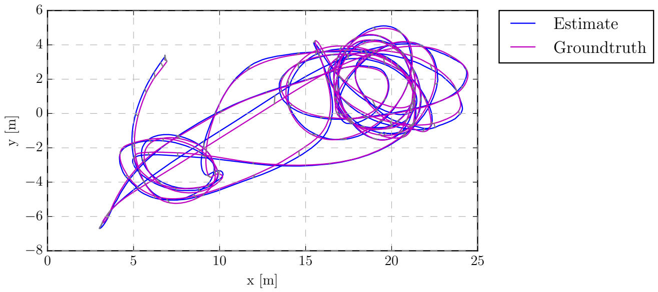
As illustrated in previous sections, the result we report comes from a filter-based monocular VIO solution, recorded trajectories are causal, all the key parameters are kept the same, and loop closure and relocalization are not applied. The experiment was conducted on a Mid 2015 MacBook Pro, with a quad-core Intel Core i7-4770HQ @ 2.2GHz, 16GB DDR3 @ 1600MHz and 256GB Samsung SSD SM0256G. We used a single thread solution reading all the files from the disk and conducting the estimation sequentially. We show some results of the sequences with ground truth in Figure 2, which are drawn by the UZH-RPG trajectory evaluation tool [13].

The total processing time and the dataset duration are listed in Table III.

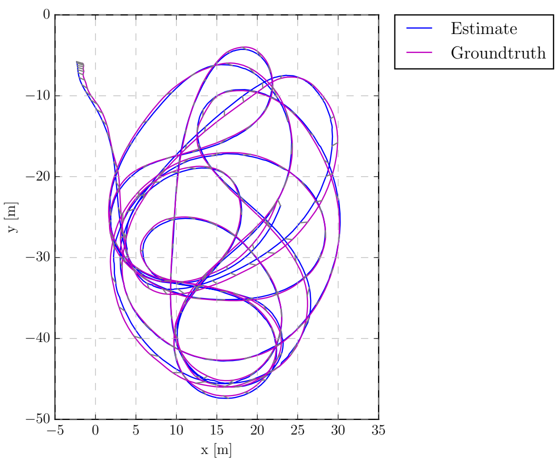
TABLE III
Total Processing Time And Data Duration For Selected Sequences.

Sequence	Processing Time (sec)	Data Duration (sec)
indoor forward 11	66.1	78.6
indoor forward 12	44.5	58.5
indoor 45 3	59.9	76.1
indoor 45 16	37.6	46.3
outdoor forward 9	70.6	88.1
outdoor forward 10	89.0	111.0

The total processing time recorded the time for processing the whole sequence including initialization precedure. Notice



(a) Indoor Forward 7



(b) Outdoor Forward 3

Fig. 2. Estimated trajectories of LARVIO aligned with ground truth in position and yaw.

that we applied a single thread solution to do the job, the processing time could be further reduced if using a multi-thread architecture.

References

[1] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Robotics: Science and Systems*. Berlin Germany, 2013, pp. 241–248.

[2] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *In Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy*, 10-14 April 2007, pp. 3565–3572.

[3] X. Qiu, H. Zhang, W. Fu, C. Zhao, and Y. Jin, "Monocular visual-inertial odometry with an unbiased linear system model and robust feature tracking front-end," *Sensors*, vol. 19, no. 8, p. 1941, 2019.

[4] X. Qiu, H. Zhang, and W. Fu, "Lightweight hybrid visual-inertial odometry with closed-form zero velocity update," *Chinese Journal of Aeronautics*, 2020. [Online]. Available: https://doi.org/10.1016/j.cja.2020.03.008

[5] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[6] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel corporation*, 2000.

[7] J. Shi *et al.*, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.

[8] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended kalman filter based slam," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 473–479.

[9] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6713–6719.

[10] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[11] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 38, no. 1, pp. 35–44, 2004.

[12] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[13] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.