OpenVINS Performance Evaluation on 2020 FPV Drone Racing VIO Dataset

Patrick Geneva and Guoquan Huang

I. SYSTEM OVERVIEW

We leverage the OpenVINS [1] system open sourced by our group, which was developed to fill a gap in the current open sourced visual-inertial navigation systems (VINS). OpenVINS focuses on providing the fundamentals for new researchers and practitioners to allow for users with little background in state estimation to learn and develop new ideas within the VINS research area. We provide the necessary documentation, tools, and theory for filter-based visual-inertial state estimation. The key components of the OpenVINS suite are as follows:

- *ov_core* – Contains 2D image sparse visual feature tracking; linear and Gauss-Newton feature triangulation methods; and fundamental manifold math operations and utilities.
- *ov_eval* – Contains trajectory alignment; plotting utilities for trajectory accuracy and consistency evaluation; Monte-Carlo evaluation of different accuracy metrics; and utility for recording ROS topics to file.
- *ov_msckf* – Contains the extendable modular Extended Kalman Filter (EKF)-based sliding window visual-inertial estimator with on-manifold type system for flexible state representation and its visual-inertial simulator for arbitrary number of cameras and frequencies. Features include: First-Estimates Jacobains (FEJ) [2]–[4], IMU-camera time offset calibration [5], camera intrinsics and extrinsic online calibration [6], standard MSCKF [7], and 3D SLAM landmarks of different representations.

At the core of the system is our on-manifold modular Extended Kalman filter (EKF)-based sliding window visual-inertial estimator. This estimates an inertial state containing the current inertial measurement unit (IMU) position, velocity and biases, along with calibration parameters, stochastic clones, and environmental temporal SLAM features. Keyframing is not used and instead we have a fixed sliding window size that always marginalize the oldest pose from our state vector and bounds the computational complexity. To both model the uncertainty of calibration values and handle imperfect calibration we estimate the time offset between the IMU and camera, along with the camera's intrinsics and extrinsic transform to the IMU. We additionally leverage temporal SLAM features and handled their consistency issues through First-Estimates Jacobains (FEJ) [2]–[4].

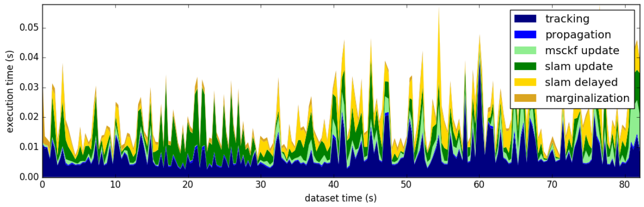

Fig. 1: Example flamegraph on the Indoor Forward 5 dataset. The computational spikes due to increase tracking cost or sudden large number of feature updates can be seen. These are handled by dropping subsequent frames to ensure estimates to not fall behind from the most recent image. Figure best seen in color.

II. DISCUSSION OF DIFFERENCES FROM ORIGINAL SUBMISSION

In what follows we discuss the key changes made to the filter which we found improved performance and overall accuracy on the UZH-FPV drone racing dataset [8]. We note that we use the new 2.0 release version of the codebase at commit 10b33a8. We tuned the system and evaluated different system configurations on a subset of the datasets which provide groundtruth. The key observations and tuning of OpenVINS is as follows:

- *Increase the amount of MSCKF and SLAM features tracked:* As more features are tracked the accuracy of the system increased but would eventually start decreasing. We hypothesize that this was due to the increase of filter update cost causing the system to process images at a lower frequency. This directly impacts the feature track quality due to the increase of feature disparity between update images due to dropped frames.
- *Use SLAM feature single-depth representation:* To allow for inclusion of larger amounts of SLAM features we switched to a single depth representation for all SLAM features estimated. This allowed for the estimation of 150 features vs the upper limit of 50 features if using a global or anchored full 3D representation. The key benefit of using the single depth, at the sacrifice of accuracy, is the allowing for more then half of active features to be SLAM. This also allowed for decreasing of the sliding window size since most features are tracked as SLAM. Details of the single-depth implementation can be found on the OpenVINS website [1].
- *Use monocular image processing:* We additionally found that the benefit of using binocular image sensors to not be as large when including a sufficient amount

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. {pgeneva,ghuang}@udel.edu

TABLE I: Evaluation time for the given datasets. We report the per-frame timing statistics and its standard deviation.

Dataset Name	Total Frame Time (sec)	Deviation (sec)
indoor forward 11	0.0200	0.0096
indoor forward 12	0.0204	0.0101
indoor 45° 3	0.0211	0.0106
indoor 45° 16	0.0199	0.0107
outdoor forward 9	0.0206	0.0105
outdoor forward 10	0.0208	0.0104

of visual features. We found that in some datasets using binocular image tracking the accuracy would increase, while in others, monocular would performed better. Thus we selected monocular to reduce the amount of computation needed for image processing.

- *Publish poses in the IMU clock frame:* The OpenVINS system originally [9] published the IMU pose with the timestamp corresponding to the image measurement timestamp. This was technically incorrect since we estimate the state in the IMU clock frame of reference as compared to the camera clock frame. Thus we publish $^{I}t = {}^{C}t_{img} + {}^{C}t_{I}$ timestamp which uses the current best estimate of the time-offset between the camera-IMU sensor pair. This can cause issues with the groundtruth alignment, but since the groundtruth is at high frequency calculating the error in respect to the closest groundtruth pose should be sufficient.

III. EVALUATION HARDWARE

The OpenVINS system was evaluated on an Intel(R) Xeon(R) CPU E3-1505M v6 @ 3.00GHz Lenovo P51 laptop with 15GB of DDR4 memory and a 1TB Samsung SSD 850 EVO. OpenVINS has very minimal multi-threaded optimization and is limited to just the feature tracking frontend. To process our monocular images we do not have any multi-threading except for extracting features in the grid pattern and OpenCV's [10] internal vectorization in its optical flow method. The rosbags are played back in *realtime* from disk and topics are subscribed to by the estimator. In cases where frames take more then 0.033 seconds to process, the next frame will be dropped due to having a ROS subscriber queue size of one. This ensures that while in the few cases where the estimator "spikes" past the realtime threshold, the next frame is the always the most recent. An example of these spikes can be seen in Figure 1. In lieu of processing the bags in serial we have reported the average time per frame in Table I for each dataset. It is very clear that on average we are able to process at far above the realtime frequency of 0.033 seconds.

IV. ALGORITHM PARAMETERS

All launch parameters are kept the same for all datasets. The system self-initializes after detecting a change in the acceleration from being picked up at the beginning of each dataset. Table II, shows the key parameters used by the OpenVINS algorithm. The time offset between the IMU and cameras was calibrated online along with the camera intrinsics and extrinsic transformations.

TABLE II: Key parameters used for all datasets.

Parameter Name	Value
sliding window size	8
max features	300
max SLAM features	150
max feat in SLAM update	40
max feat in MSCKF update	40
fast threshold	10
fast grid x/y	10/8
min feat. pixel distance	5
raw pixel noise	1.0
acc. white noise	2.0000e-2
acc. random walk	3.0000e-3
gyro. white noise	1.6968e-03
gyro. random walk	1.9393e-05

REFERENCES

[1] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[2] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 19-23 2008, pp. 473–479.

[3] ——, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. of the 11th International Symposium on Experimental Robotics*, Athens, Greece, July 14–17, 2008.

[4] ——, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, Apr. 2010.

[5] M. Li and A. I. Mourikis, "Online temporal calibration for Camera-IMU systems: Theory and algorithms," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, June 2014.

[6] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 409–416.

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

[8] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzhfpv drone racing dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[9] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins performance evaluation on 2019 fpv drone racing vio dataset," Tech. Rep., 2019. [Online]. Available: http://rpg.ifi.uzh.ch/uzh-fpv/IROS2019/reports/Geneva-Delaware.pdf

[10] OpenCV Developers Team, "Open source computer vision (OpenCV) library," Available: http://opencv.org.