OpenVINS Performance Evaluation on 2019 FPV Drone Racing VIO Dataset

Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang

I. SYSTEM OVERVIEW

We leverage the OpenVINS [1] system recently 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; visual-inertial simulator for arbitrary number of cameras and frequencies; 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. 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 which are estimated in an anchored frame with an inverse depth representation. We found that while few SLAM features are tracked during highly dynamic motion, the inward facing turns, take off and landing segments greatly benefited from the inclusion of these features. We handled consistency issues through First-Estimates Jacobains (FEJ) [2]–[4] which have been shown to improve estimator performance.

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

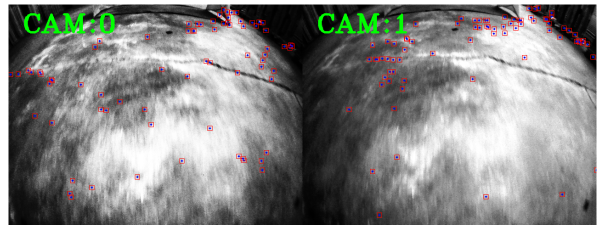

Fig. 1: Extracted fast extractions from the Indoor 45° 14 dataset. These high motion areas with low texture floors prove a challenge for indirect feature extraction. Figure best seen in color.

II. FEATURE TRACKING DISCUSSION

As emphasised in the UZH-FPV drone racing dataset paper [8] the large magnitude of visual optical flow in these datasets present a challenge for feature tracking. OpenVINS supporting both indirect sparse feature KLT tracking [9] and descriptor based methods [10] through the implementations available in OpenCV [11]. We have chose to use KLT as it allows for longer feature tracks which facilitates longer lived temporal SLAM features. Features are first extracted in a uniform grid using FAST detection [12], and as mentioned in the literature, is one of the key weaknesses of indirect-based visual tracking methods in low gradient environments, see Figure 1. We found that the Indoor 45° datasets had particular trouble during the high speed straight segments due to the lack of good feature extractions on the low texture floor regions. The use of direct-based visual feature tracking could possibly help address this. It is also interesting that our stereo implementation, which enforces that all feature tracks are seen in both cameras, does not work due to low amount of features extracted and poor cross camera tracking quality.

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, of which it is limited to just the feature tracking frontend. The rosbags are read in serial from disk to allow for dataset completion timing. The total time taken for a dataset is based on the CPU clock time at initialization of the VIO and until the MAV hits the ground. Table I shows

TABLE I: Evaluation time for the given datasets. Timing from after initialization of the VIO till the MAV hits the ground.

Dataset Name	Eval. Time (sec)	Dataset (sec)
indoor forward 11	54	75
indoor forward 12	32	51
indoor 45deg 3	50	76
indoor 45deg 16	27	46
outdoor forward 9	55	89
outdoor forward 10	72	111

TABLE II: Key parameters used for all datasets.

Parameter Name	Value
sliding window size	15
max features	120
max SLAM features	40
fast threshold	15
fast grid x/y	10/8
min feat. pixel distance	10
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

the time taken on each dataset for the proposed OpenVINS system and the total length of the rosbag processed after initialization. The time for each update is evenly split between visual feature tracking and update, with the average estimation frequency being between 30-40 Hz.

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. We found that in the outdoor datasets uniform extraction of features is more necessary then indoors due to the large quantity of high gradient textures. As noted, we found that requiring all features to have stereo constraints to be unreasonable, and instead processed the left and right images independently in a binocular configuration. The feature tracking parameters in Table II are for each camera. The time offset between the IMU and cameras was calibrated online along with the camera intrinsics and extrinsic transformations.

REFERENCES

[1] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications*, Macau, China, Nov. 2019. [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] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the International Joint Conference on Artificial Intelligence*, Vancouver, BC, 1981, pp. 674–679.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

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

[12] E. Rosten, R. B. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.