A Benchmark Comparison of Three Visual-Inertial Odometry Algorithms

Zhi Fang Tan, Lingyi Song, Kexuan Zhai
Department of Automotive Engineering, Clemson University
International Center for Automotive Research (CU-ICAR)
2020 Spring Course Report: AuE 8930 Machine Perception and Intelligence, Instructor: Dr. Bing Li

Abstract

Cameras and inertial measurement units (IMUs) require less power and payload, so visual inertial odometry (VIO) algorithms are popular choices for state estimation in many scenarios, in addition to their ability to operate without external localization from motion capture or global positioning systems. Our project tries and evaluates three publicly-available VIO pipelines (OKVIS, ROVIO[1-2], and VINS-Mono) in terms of their accuracy.

1. Introduction

1.1.1 The motivation of this work

SLAM has been a very popular perception topic in both robotics and autonomous driving fields. Visual SLAM, or Visual Odometry, has been used in a wide variety of robotic applications.

Visual-Inertial Odometry (VIO), with the combination of camera and IMU, can perform state estimation without the assistance of other sensors, costing less power, load and computing ability of the platforms, the state estimation algorithm must be run under limited resources.

We can get insights of the performance and principles of these algorithms by running and comparing these methods.

1.2.1 The challenges of this work

- Understanding the principle of each algorithm.
- The building of the experiment environment and dependencies
- Getting the output trajectory for evolution. OKVIS and ROVIO both need to modify their codes for getting the

2. Methodology

2.1.1 The evaluation tools

Evo: it is a Python package for the evaluation of odometry and SLAM.

This package provides executables and a small library for handling, evaluating and comparing the trajectory output of odometry and SLAM algorithms. Figure 1 shows visualization functions of Evo. The advantages of the package are:

- There are common tools for different formats.
- The algorithmic options for association, alignment, scale adjustment
- Flexible options for output, plotting or export (e.g. LaTeX plots or Excel tables)
- A powerful, configurable CLI that can cover many use cases.
- Modular core and tools libraries for custom extensions.
- Faster than other established Python-based tools.

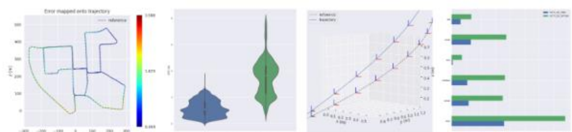

Figure 1: The visualization functions of Evo.

2.2. EuRoC dataset

We used EuRoC dataset as the inputs of the algorithms. EuRoC dataset presents visual-inertial datasets collected on-board a Micro Aerial Vehicle (MAV). The datasets contain stereo images, synchronized IMU measurements, and accurate motion and structure ground-truth. Developed by Autonomous Systems Lab (ASL), ETH Zurich, Switzerland. In our case, data from Machine Hall 01~05 are used. Figure 2 shows the Machin Hall (left), and the MAV for collection of datasets (right).

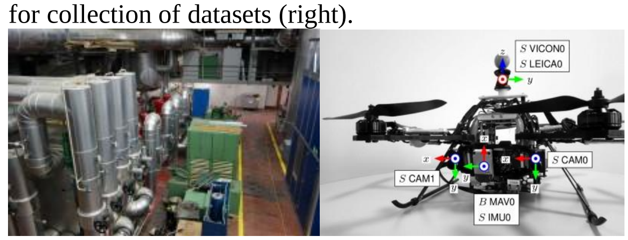

Figure 2: The Machin Hall (left), and the MAV for collection of datasets (right).

Data	Num of poses	Path length(m)	Duration(s)	Difficulty level
MH01	36382	80.626	181.905	Easy
MH02	29993	73.473	149.960	Easy
MH03	26302	130.928	131.505	Medium
MH04	19753	91.747	98.760	Difficult
MH05	22212	97.593	111.055	Difficult

Table 1: The information of the datasets.

2.2. VINS-Mono

2.2.1. Challenges

There are several issues affect the usage of monocular VINS. The first one is rigorous initialization. Due to the lack of direct distance measurements, it is difficult to directly fuse the monocular visual structure with inertial measurements. Also recognizing the fact that VINSs are highly nonlinear, we see significant challenges in terms of estimator initialization. In most cases, the system should be launched from a known stationary position and moved slowly and carefully at the beginning, which limits its usage in practice. Another issue is that the long-term drift is unavoidable for visual-inertial odometry (VIO). In order to eliminate the drift, loop detection, relocalization, and global optimization has to be developed. Except for these critical issues, the demand for map saving and reuse is growing.

2.2.2. Features

To address all these issues, VINS-Mono was proposed by HUKST Aerial Robotics Group. VINS-Mono is a real-time SLAM framework for Monocular Visual-Inertial Systems. It uses an optimization-based sliding window formulation for providing high-accuracy visual-inertial odometry. It features efficient IMU pre-integration with bias correction, automatic estimator initialization, online extrinsic calibration, failure detection and recovery, loop detection, and global pose graph optimization, map merge, pose graph reuse, online temporal calibration, rolling shutter support. VINS-Mono is primarily designed for state estimation and feedback control of autonomous drones, but it is also capable of providing accurate localization for AR applications. VINS-Mono contains following features:

- Robust initialization procedure that is able to bootstrap the system from unknown initial states;
- Tightly coupled, optimization-based monocular VIO with camera–IMU extrinsic calibration and IMU bias correction;
- Online re-localization and four degrees-of-freedom (DOF) global pose graph optimization;
- Pose graph reuse that can save, load, and merge multiple local pose graphs.

2.2.3. Algorithm

Block diagram illustrating the full pipeline of the proposed monocular VINS. The system starts with measurement preprocessing. The initialization procedure provides all necessary values for bootstrapping the subsequent nonlinear optimization-based VIO. The VIO with relocalization modules tightly fuses preintegrated IMU measurements, feature observations, and redetected features from the loop closure. Finally, the pose graph module performs global optimization to eliminate drift and achieve reuse purpose.

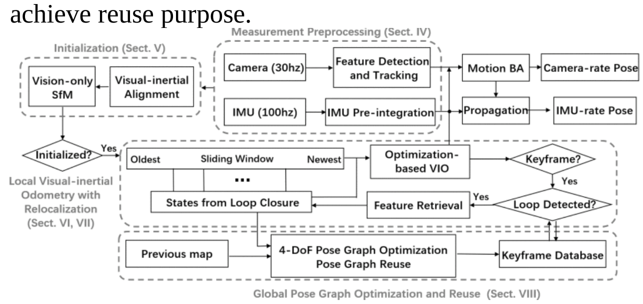

Figure 3: Block diagram of VINS Mono.

2.2.4. Experiment Setup and result

The hardware setup includes: RAM: 8 GB, Processor: Intel Core i7-950H CPU. The software setup includes: Ubuntu 16.04, ROS Kinetic and Ceres Solver. The demo is showed in the following figure.

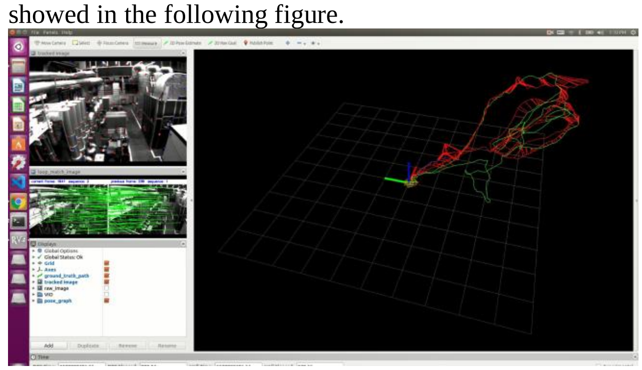

Figure 4: Demo of VINS Mono running EuRoC dataset.

2.3 OKVIS

2.3.1 Introduction

OKVIS is a Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. The output of the okvis library is the pose T_WS as a position r_WS and quaternion q_WS, followed by the velocity in World frame v_W and gyro biases (b_g) as well as accelerometer biases (b_a). See the example application to get an idea on how to use the estimator and its outputs (callbacks returning states). Figure 5 shows the coordinate systems for OKVIS.

OKVIS is a tightly coupled VIO algorithm, same as VINS.

2.3.2 Features

OKVIS combined the IMU and reprojection error terms into a cost function to optimize the system. The old keyframes were marginalized to maintain a bounded-sized optimization window, ensuring real-time operation. As a first step to initialization and matching, they propagated the last pose using acquired IMU measurements to obtain a preliminary uncertain estimate of the states. To avoid repeated constraints caused by the parameterization of relative motion integration, pre-integration was proposed to reduce computation. The pre-integration principle is illustrated in Figure 6.

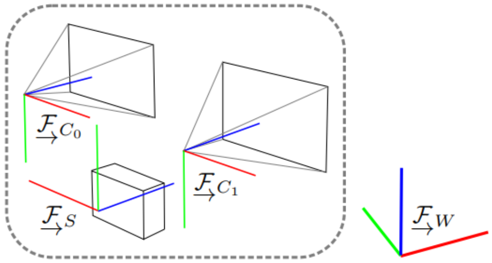

Figure 5: Coordinate frames involved in the hardware setup used: two cameras are placed as a stereo setup with respective frames, $\underrightarrow{F}_{C_i}$, i ∈ {0, 1}. IMU data is acquired in $\underrightarrow{F}_S$. $\underrightarrow{F}_S$ is estimated with respect to $\underrightarrow{F}_W$.

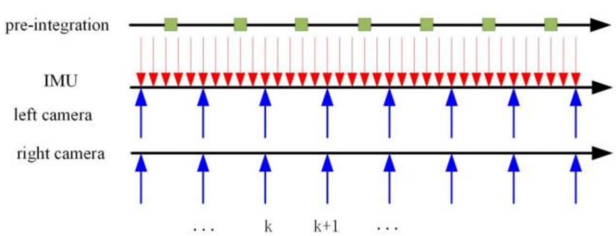

Figure 5: Pre-integration principle

2.3.3 The Front End of OKVIS

It employs a customized multiscale SSE-optimized Harris corner detector combined with BRISK descriptor extraction. The keypoint matching includes the matching between current frame and last keyframe; and between the current frame and last frame. Figure 6 shows the keypoint matching during the running time of OKVIS.

2.3.4 The running of OKVIS

In this project, both monocular and stereo version of OKVIS is been performed. And it turns out there is fewer tracking failures and a better output trajectory for the stereo version. This will be discussed later. Figure 7 and Figure 8 show the running screenshots of monocular and stereo version of OKVIS.

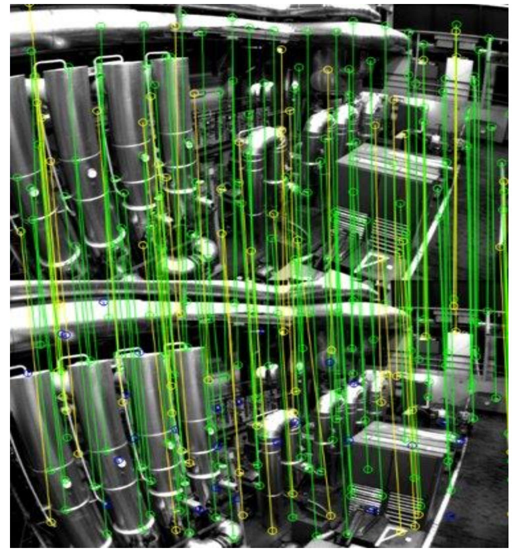

Figure 6: Keypoint matching of OKVIS. The upper frame is the keyframe, the lower frame is the current frame.

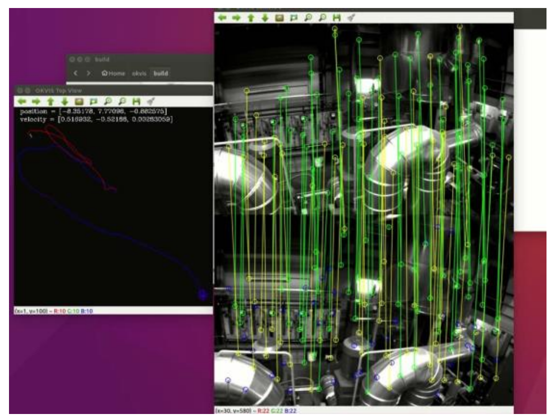

Figure 7: Monocular OKVIS

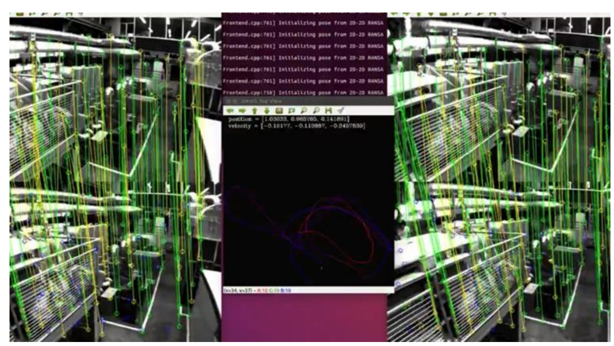

Figure 8: Stereo OKVIS

2.2.1 ROVIO

Similar to the previous approaches, ROVIO [1-2] is a tightly coupled visual inertial odometry method. However, one particular unique thing about this method is that it is based on Iterated Kalman Filter. Implementing Iterated Kalman Filter(IEKF) can prevent potential fail when the initial guess for bearing vector of visual features has large uncertainty. For the visual front end, multilevel patch features are closely coupled to the IEKF by directly using the intensity error during the update step in the filter.

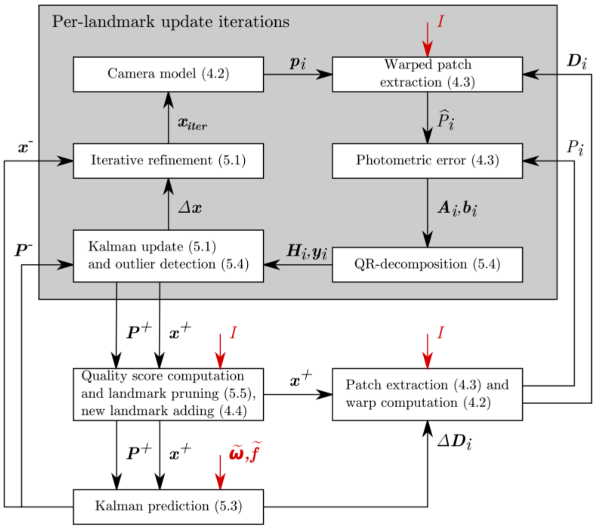

Figure 9: ROVIO Framework

2.3.1 ROVIO Iterated Kalman Filter

From the figure above, the ROVIO framework starts with Kalman prediction using the IMU measurements. Next, the visual features are used in the update step. From there, several iterations are performed to further optimize the state vector as mentioned before. Instead of using reprojection error, the system first predicts a warping matrix for the feature patch to extract the patches on current frame. Next, the photometric error of the patches between these two frames. The Jacobian, A and error term, b of the photometric error is calculated and used to update the state iteratively.

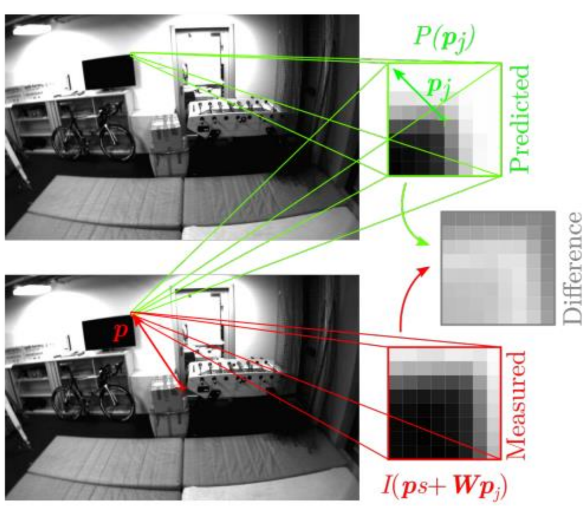

Figure 10: Tracking of Multilevel Patch Features

3. Result comparison

Several changes are made to OKVIS and ROVIO to output the result in TUM format which is represented by x,y,z,qx,qy,qz,and w. VINS-Mono does output odometry in .csv format, but it is arranged as x,y,z,w,qx,qy, and qz. Therefore, some postprocessing on the output data is required to compare them on EVO-Toolkit.

	Coupled Method	Back End Method	Front End Features	Loop Closure
ROVIO	Tightly Coupled	Filter Based	FAST + Optical Flow	No
OKVIS	Tightly Coupled	Optimization Based	Harris + BRISK	No
VINS-Mono	Tightly Coupled	Optimization Based	GFTT + Optical Flow	Yes

Figure 11: VIO Comparison

RMSE	MH_01	MH_02	MH_03	MH_04	MH_05
ROVIO	0.2773	X	0.4036	0.9226	1.1992
OKVIS Mono	0.3699	0.3976	0.3295	0.2972	0.4901
OKVIS Stereo	0.1897	0.1759	0.2147	**0.1722**	0.3146
VINS Mono Loop	**0.0826**	**0.1021**	**0.0773**	0.2031	**0.1927**
VINS Mono No Loop	0.1571	0.1784	0.1950	0.3470	0.3035

Figure 12: Results on EUROC MH_01-05 dataset

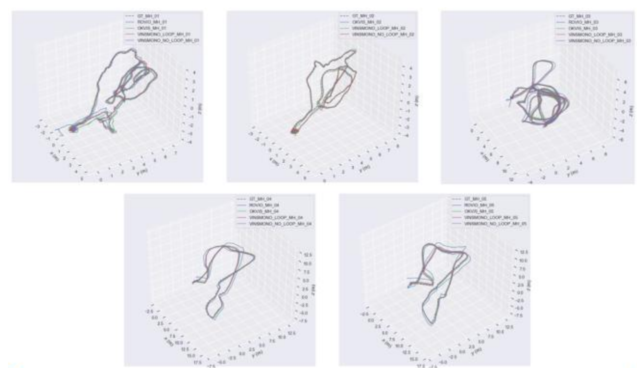

Figure 13: VIO Trajectories Plotted using EVO Toolkit

From the results above, VINS Mono with loop closure enabled has the best accuracy compare to other method. In the MH02 data, ROVIO is not able to converge due to the unstable feature tracking in the beginning of the data. Therefore, there is no result for that particular data in the table. OKVIS Stereo has significantly better result than the monocular version. However, stereo version does require more computational power.

4. Future work

For future work, we would like to experiment on different kinds of localization framework such as RGB-D SLAM, LIDAR SLAM, VO, etc. Another interesting idea for future implementation is to test the results on outdoor data. One of the main challenges of outdoor benchmarking will be ground truth acquisition.

References

[1] Bloesch, Michael, et al. “Robust Visual Inertial Odometry Using a Direct EKF-Based Approach.” 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, doi:10.1109/iros.2015.7353389.

[2] Bloesch, Michael, et al. “Iterated Extended Kalman Filter Based Visual-Inertial Odometry Using Direct Photometric Feedback.” The International Journal of Robotics Research, vol. 36, no. 10, 2017, pp. 1053–1072., doi:10.1177/0278364917728574.

[3] Tong Qin, Shaojie Shen. Online Temporal Calibration for Monocular Visual-Inertial Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, 2018)

[4] Tong Qin, Peiliang Li, Zhenfei Yang, Shaojie Shen, VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator, IEEE Transactions on Robotics.

[5] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 2502-2509.

[6] Leutenegger, Stefan & Lynen, Simon & Bosse, Michael & Siegwart, Roland & Furgale, Paul. (2014). Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization. The International Journal of Robotics Research. 34. 10.1177/0278364914554813.