Analysis of onboard sensor-based odometry for a quadrotor UAV in outdoor environment

Aidar Gabdullin, Grigory Shvedov, Mikhail Ivanou, Ilya Afanasyev
Robotics Institute, Innopolis University
1 Universitetskaya street, 420500 Innopolis, Russia
E-mail: {a.gabdullin, g.shvedov, m.ivanov, i.afanasyev}@innopolis.ru
http://university.innopolis.ru

Abstract

This article introduces a comparative analysis of a quadrotor UAV trajectories evaluated by processing onboard sensors (camera and IMU) with ROS-based monocular visual odometry software. Parrot Bebop drone was launched outdoor within teleoperated closed-loop trajectory along a known perimeter of a square work area, recording sensor's telemetry data for offline processing. We compared UAV monocular SLAM trajectories with Bebop visual-inertial odometry and verified them with the ground truth estimated by an external observer (a hovering quadcopter DJI Mavic, flying at the height of 40m with a camera pointing downwards and tracking near-surface Bebop's flight).

Keywords: monocular visual odometry, GPS-denied environment, quadrotor UAV, SLAM, ROS

1. Introduction

At present, the tasks of autonomous quadrotor UAV localization, mapping and navigation in GPS-denied environment are solved mainly by vision-based sensing and inertial navigation systems (INS), combining visual feature data from a monocular camera with inertial sensor measurements [1-4]. During the last years many studies of GPS-denied indoor navigation have focused on the monocular visual localization applications [5-8] and comparative analysis of different vision-based algorithms [9-11], since visual localization is frequently a natural substitution of a GPS approach. However, there is still a lack of investigations dealt with efficiency and robustness of monocular visual odometry methods for UAV quadrotors in GPS-denied outdoor environment.

When hovering outdoors, especially in a windy condition, a conventional quadrotor UAV with uncalibrated fixed camera actively changes pitch and roll angles, recording a video with a poor quality due to shakes and high-frequency vibrations from motors and propellers. To solve this issue, it is used a camera with a good image quality and robust digital stabilization. In our research, we employ a Parrot Bebop drone outdoor in teleoperated mode, making a closed-loop trajectory along a known perimeter of a work area at a height of 1.5-2m, acquiring sensors telemetry data for offline processing with Robot Operating System (ROS) packages.

The rest of this paper is organized as follows. Section 2 describes system setup. Section 3 presents ROS-based computations of visual odometry trajectories and their verification by an external observer. Section 4 provides outdoor experiment results with comparative accuracy analysis. Section 5 concludes this paper.

2. System setup

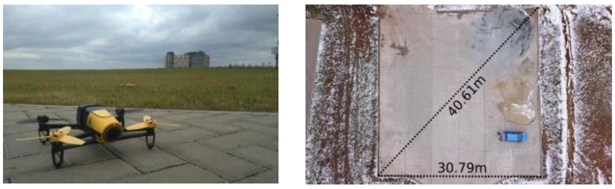
In our outdoor experiments we used Parrot Bebop drone (Fig. 1), as a robust quadrotor UAV, which has a digitally stabilized camera and special flight modes with drone landing in case of collisions[a]. Bebop controls stability and maneuverability of its flight by automatic processing sensors data from 3D inertial measurement unit (IMU), an ultrasound sensor of an 8m range, a pressure sensor and a vertical camera (to track the speed). Bebop quadrotor can record video in a 180° panoramic view by the 14mpx "fisheye" camera with digital image stabilization, taking stable aerial footage independently on UAV maneuvers. Parrot Bebop drone has a firmware with integrated visual-inertial velocity estimation (without GPS), and an open source ROS wrapper[b] to get and process telemetry data.

For a top-view Bebop's trajectory validation (as the ground truth) we used a hovering DJI Mavic quadrocopter [c] as an external observer. It was also teleoperated to hold a position at the height of 40m with a camera pointing downwards and tracking near-surface Bebop's flight. Its camera is actively gyro-stabilized with a brushless gimbal, recording a video with the resolution of 1920*1080 at 60 fps. For observer sensor's data processing we utilized OpenCV 3.0[d] and ROS Kinetic[e].

3. Bebop drone odometry and its verification

3.1. *Bebop visual-inertial odometry*

Parrot Bebop drone has a default option of odometry calculation by a bebop_autonomy driver[f], which allows reading from UAV camera, odometry, GPS and drone states (i.e. on-board data also known as Navdata). The odometry integrates visual-inertial velocity over time (dead reckoning), using video stream from Bebop's front camera with a *bebop_driver* and Bebop's position & velocity in an ENU aligned odometry (which is based on REP-103 specifies an ENU (East, North, Up) coordinate frame for outdoor navigation [g]). Since odometery is calculated from Bebop States (Navdata), the update rate is limited to 5 Hz that can be a reason for growth of trajectory deviation error during Bebop flight (see loop closure displacements in trajectories in Fig. 2a during three cycles of UAV flight along the same work area perimeter in Fig. 1). Visual-inertial odometry seems to provide relevant results in steady-weather conditions, but in strong wind and rapid changes in accelerations, we observed a displacement at ~10m per cycle at a wind of 6m/s (Fig. 2) Therefore, we employed DJI Mavic drone as an external observer for a ground truth.

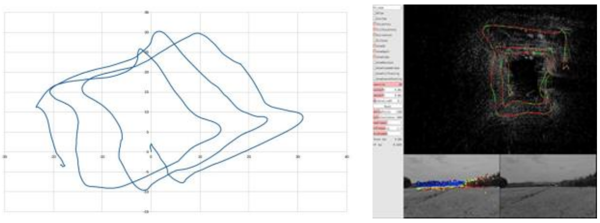


(a) Parrot Bebop drone. (b) Experimental work area

Fig. 1. Outdoor experiments with Parrot Bebop drone

3.2. *Odometry verification by external observer*

As a ground truth (reference trajectory) we used an external observer data taken from a hovering DJI Mavic quadcopter, flying at an altitude of 40m with a camera pointed down and tracking near-surface Bebop's flight. We processed the image footages offline, tracking Bebop with following steps: (1) Computing a dense optical flow with color coding for each video frame [12]; (2) Converting images to grayscale and thresholding; (3) Detection of Bebop contour and its center position *(x,y)*. As additional DJI Mavic camera calibration the landmark size in work area were measured in advance, thus we transformed image pixels to geocoordinates in meters.



(a) Bebop visual-inertial odometry. (b) DSO results

Fig. 2. Difficulties with UAV odometry estimations: (a) Three loop closure displacements in Bebop visual-inertial odometry trajectories (in meters) for UAV flight under the same work area in windy conditions; (b) Divergence of UAV trajectory received with Direct Sparse Odometry (DSO).

3.3. *ROS-based Visual SLAM*

ORB-SLAM: Oriented FAST and Rotated BRIEF[h] [13] in most cases is one of the most accurate and robust

[a] Parrot Bebop Drone description: http://global.parrot.com/

[b] ROS driver for Parrot Bebop drone, based on Parrot's official ARDroneSDK3: http://wiki.ros.org/bebop_autonomy

[c] DJI Mavic drone: https://www.dji.com/mavic

[d] OpenCV 3.0 library: https://opencv.org/opencv-3-0.html

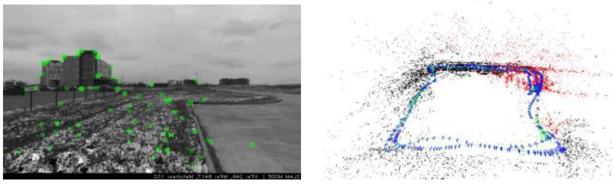
[e] ROS Kinetic Kame middleware : http://wiki.ros.org/kinetic

[f] ROS Driver for Parrot Bebop Drone (quadrocopter) 1.0 & 2.0: http://bebop-autonomy.readthedocs.io/en/latest/index.html

[g] REP-103 provides a reference for the units and coordinate conventions used within ROS: http://www.ros.org/reps/rep-0103.html

[h] ORB-SLAM2 package, github.com/raulmur/ORB SLAM2

SLAM solution that has ROS-based package with bundle adjustment, monocular feature-based observations and camera trajectory estimation. Thus, ORB SLAM includes features from different images in 3D space and visual odometry tracks for unmapped regions (Fig. 3). Finally, we built the UAV trajectory, but while testing we detected some ORB-SLAM difficulties with: (1) random character of initialization (mostly, due to lack of features); (2) adding new points to a map (since ORB-SLAM needs some time for new keyframes); (3) camera rotations (then SLAM tracking tends to get lost features and trajectory); (4) dependence trajectory deviation accuracy on camera calibration and scaling to metric scale. To overcome some problems in our experiment we teleoperated Bebop drone along closed-loop trajectory with slow rotations in the work area corners.

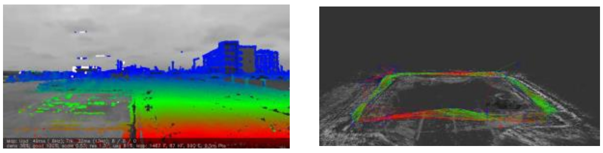


(a) Extracted features.

(b) Point cloud and trajectory

Fig. 3. ORB SLAM point clouds and trajectory visualization

LSD-SLAM: Large-Scale Direct Monocular SLAM[i] [14] performs image-to-image alignment with simultaneous tracking, depth map estimation and optimization (Fig. 4). However, this method is highly demanding of computing resources and sensitive to camera calibration. Despite map optimization, which includes loop closure detection, the resulting trajectory has lower precision than ORB-SLAM one (although LSD-SLAM point cloud is much denser than for ORB-SLAM). In our case, Bebop's camera has a rolling shutter, thus the results were not so accurate as they could be with global shutter camera. Nevertheless, LSD-SLAM demonstrates stable work, excepting the cases of fast camera rotation, and has no problems with initialization.



(a) Depth map with features.

(b) Point cloud and trajectory

Fig. 4. LSD-SLAM featured depth map and UAV trajectory.

DSO: Direct Sparse Odometry[j] [15] combines sparse and direct visual odometry methods, minizing a photometric error evaluated directly from images. DSO is sensitive to camera calibration and suffers from Bebop camera distortions, therefore in our case instead of loop closure we got divergence of UAV trajectories (Fig. 2b).

4. Outdoor Experiment and Results

Experimental dataset was recorded in bag-file during Parrot Bebop drone flight and sensor's telemetry data was processed offline with visual SLAM algorithms. Fig. 5 illustrates the comparison of all UAV odometry trajectories evaluated with ROS-based monocular visual SLAM methods, Bebop visual-inertial odometry and the reference trajectory (ground truth) from the external observer (DJI Mavic drone). For monocular SLAM methods (ORB and LSD) the computed trajectories were properly scaled. For error estimation we computed maximum and average deviations with the formulae:

$$max\ dev. = \max\left(\sqrt{\left(x_i^{traj} - x_i^{gt}\right)^2 + \left(y_i^{traj} - y_i^{gt}\right)^2}\right), \quad (1)$$

$$avg.\ dev. = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(x_i^{traj} - x_i^{gt}\right)^2 + \left(y_i^{traj} - y_i^{gt}\right)^2}. \quad (2)$$

where *x*,*y* are coordinates of odometry estimation and ground truth correspondingly, and N is a number of samples (points) in path. Table 1 illustrates the error estimations (deviations) for UAV odometry trajectories.

Table 1. Estimated max and mean errors of UAV trajectory deviations for UAV sensor-based odometry

Method	**Average deviation, m**	**Maximum deviation, m**
Bebop visual-inertial odometry	2.53	9.85
ORB-SLAM	2.56	8.52
LSD-SLAM	4.45	14.91

[i] LSD-SLAM package, https://github.com/tum-vision/lsd slam
[j] DSO package, https://github.com/JakobEngel/dso

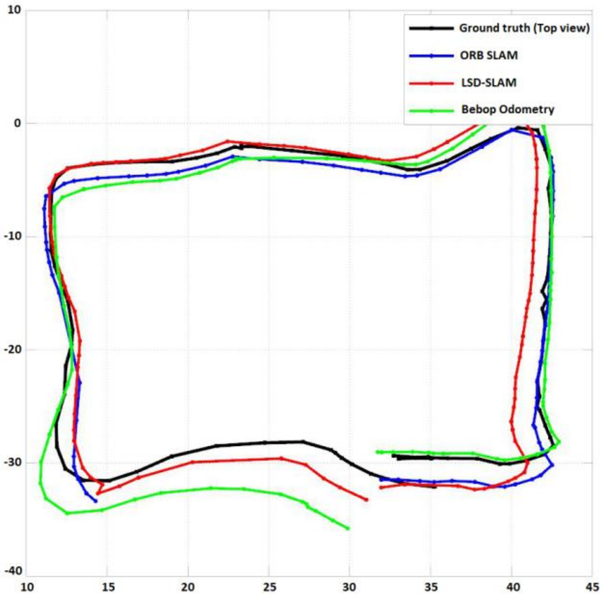


Fig. 5. Comparison of Bebop drone trajectories computed by monocular ORB-SLAM (blue curve), monocular LSD-SLAM (red curve), Bebop Visual-Inertial Odometry (Green curve). Black curve is the ground truth (top view from the external observer). The units for the XY axes are meters.

5. Conclusions

In this paper we studied the experimental results for application of some monocular visual odometry methods to Parrot Bebop drone outdoor closed-loop flight and analyzed UAV trajectories evaluated by offline processing of onboard sensors (camera and IMU) with ROS-based software. We compared UAV monocular SLAM trajectories with Bebop visual-inertial odometry and verified them with the ground truth estimated by an external observer (a hovering quadcopter DJI Mavic, flying at the height of 40m with a camera pointed down and tracking near-surface Bebop's flight).

The main conclusions from our experiment are: (1) There are difficulties with Parrot Bebop drone trajectory recovering for all considered monocular SLAM algorithms (due to possible tracking failure in case UAV rotations, and SLAM sensitivity to Bebop camera geometric distortions and calibration). (2) All monocular SLAM methods need rescaling to metric scale for proper UAV trajectory evaluation (that requires a reliable ground truth). (3) In our experiments, visual odometry based on ORB-SLAM is more accurate and robust to outdoor environment than Bebop visual-inertial odometry and LSD-SLAM, although sometimes it has difficulties with initialization and camera rotations.

Acknowledgements

The research was partially supported by Russian Foundation for Basic Research (RFBR), project ID 15-07-07483, Russian Ministry of Education and Science within the Federal Target Program (research grant ID RFMEFI60917X0100) together with KAMAZ company as the industrial partner, and Innopolis University.

References

1. M.J. Milford et al, Aerial SLAM with a single camera using visual expectation, in *IEEE ICRA* (2011), 2506-2512
2. D. Scaramuzza et al., Vision-controlled micro flying robots, *Robotics & Automation Magazine* 21 (2014)
3. P. Li, M. Garratt, A. Lambert, Monocular Snapshot-based Sensing and Control of Hover, Takeoff, and Landing for a Low-cost Quadrotor, *Field Robotics*, 32 (2015), 984-1003
4. M. Faessler et al., Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *Journal of Field Robotics* 33.4 (2016), 431-450
5. A. Buyval, M. Gavrilenkov, Vision-based pose estimation for indoor navigation of unmanned micro aerial vehicle based on the 3D model of environment, in *MEACS* (2015)
6. K. Yakovlev et al., Distributed control and navigation system for quadrotor UAVs in GPS-denied environments. In *Intelligent Systems*, Springer (2015), 49-56
7. A. Buyval, M. Gavrilenkov, E. Magid, A multithreaded algorithm of UAV visual localization based on a 3D model of environment, in *ICAROB* (2017)
8. A. Bokovoy, K. Yakovlev, Original Loop-closure Detection Algorithm for Monocular vSLAM, *ArXiv preprint arXiv:1707.04771* (2017)
9. A. Buyval, I. Afanasyev, E. Magid, Comparative analysis of ROS-based Monocular SLAM methods for indoor navigation, in *Machine Vision*, SPIE (2017), 103411K
10. M. Sokolov, et al., Analysis of ROS-based Visual and Lidar Odometry for a Teleoperated Crawler-type Robot in Indoor Environment, in *ICINCO* (2017)
11. I. Ibragimov, I. Afanasyev, Comparison of ROS-based Visual SLAM methods in homogeneous indoor environment, in *IEEE WPNC* (2017)
12. G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in *Image analysis* (2003), 363-370
13. R. Mur-Artal, and J.D. Tardós, ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras, *ArXiv preprint arXiv:1610.06475* (2016)
14. J. Engel, T. Schöps, and D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*. Springer, Cham (2014), 834-849
15. J. Engel, V. Koltun, and Daniel Cremers, Direct sparse odometry, *IEEE Trans. PAMI* (2017)