Summary - Comparison of Stereo Visual Inertial Odometry Algorithms for Unmanned Ground Vehicles

Supervisor: prof. Marcello Chiaberge

Candidate: Roberto Cappellaro

Introduction

PIC4Ser is the PoliTO Interdepartmental Centre for Service Robotics. Its objective it to connect and coordinate the activity of researchers with different engineering background, in order to develop new technologies in the complex service robotic field. The research covers different areas and one consists in the development of technologies and programs for indoor localization of UAV and UGV. For this task it is necessary to have a robust and accurate indoor localization and this is where this thesis project came to be. In fact this works aims to study different types of visual inertial algorithms to assess which one is the best choice for indoor localization with the already available commercial off-the-shelf hardware. Despite the fact that the end application is intended for aerial vehicles, a Jackal Unmanned Ground Vehicle (UGV) was used for the tests, since it was the vehicle available and it lowered the risks of damages during the maneuvers. The sensor used was a MYNTYEYE S stereo camera, that comes already built with a inertial measurement unit inside and two infrared lights to project a pattern of points for each monocular camera. One of the stereo visual inertial framework considered is ROVIO, a light-weight filter-based program designed UAV applications, that can work even in rough scenarios, like in almost no light conditions and in very low textured environments. An optimization-based framework, VINS-Fusion, was also considered for its good results in VIO-SLAM mode and its VIO-only module was tested. The last framework considered was OKVIS, also optimization-based, but older then VINS. Its main quality is the use of BRISK to extract robust features and improve the localization accuracy.

Objectives

The main objectives consist in:

- Analyze the camera factory calibration and re-calibrate it if needed,
- Define a test methodology and a metrics to compare the algorithms,
- Determine the best camera settings and parameters settings for each framework, and confront their accuracy in an indoor environment.

Camera calibration

The camera factory calibration was not good enough for the intended use, meaning that the reprojection error should be lower than 0.5 pixels but it was above 1 pixel. The main problem was the camera model used that could not handle such a high field of view that the camera presents. To increase the calibration accuracy it was used another model and the calibration was performed using Kalibr, an open-source toolbox that performs multiple camera extrinsic and intrinsic calibration, and camera-IMU spatial and temporal calibration. In this way it was possible to reach a reprojection error with a maximum of about 0.2 pixels.

Tests

Two cases were tested in two different indoor environments:

- Straight path in a corridor: the robot went straight ahead for about 15 meters, then rotated by 180 degrees and came back to the starting point, without rotating again,
- Pseudo-rectangular path in a room: the robot followed a path of about 16 meters, composed of straight movements and 90 degrees turns, to come back to the initial pose.

The UGV with the camera mounted on it was moved with a controller to follow the two paths marked on the ground with tape. The trial was considered admissible if the robot lateral displacement was less then half of the wheel thickness (2.5 cm) in some predefined control points. Both the UGV and the camera interfaced with ROS. The camera fps were set to 20, because it seemed a good balance between information bandwidth and real-time operability. Different tests were performed with two IMU frequency settings of 200 Hz and 500 Hz, and with the IR point projection turned on or off. Repeating the trial for each scenario the result were 16 files. The recorded data was run offline on a laptop with the VIO algorithms. At the beginning it was used a ROS 3D visualizer to display the recorded data and the estimated trajectory. In this phase the odometery coming from the Jackal was used as ground truth, to optimize the VIO frameworks settings.

Results analysis

For a quantitative analysis of the results, the algorithms were run and the data saved in a ROS bag file, using modified roslaunch files and a bash script. First the ROS bag files were converted in text files for a faster loading, using an adaptation of the "bag to pose" script from the "rpg trajectory evaluation" repository. Then, the files were then loaded in trajectory objects from the "trajectory toolkit" repository and modified with a custom python script.
At the beginning the idea was to use the Jackal odometry as ground truth to evaluate the algorithms performance. It was difficult to start each trial from the exact same pose and the orientation misalignment is especially important, because impacts on the estimated position increasingly with the traveled distance. To account for it, the Jackal odometry trajectory were aligned with each other (in the same environment), considering the linear regression of the points in the first 20 cm of traveled distance. This realignment angle was also applied to the other odometries of the correspondent cases.
Analyzing OKVIS and ROVIO trajectories it was found a common rotation about the vertical axis in all cases, possibly coming from a small tilt angle between the camera and Jackal reference frame. To account for it, all the VIO trajectories were rotated by 2 degrees counterclockwise. VINS-Fusion instead shows a bigger common rotation angle of about 5 degrees, but clearly not resulting from physical rotation, instead maybe caused by an incorrect world frame initialization.
Once the data were plotted for each case, with also the ideal trajectory coming from measurements, it was evident that the Jackal odometry accuracy was in many cases comparable with the VIO ones, thus it was not possible to use it as ground truth.
It were therefore defined two metrics based on the ideal path:

- The relative error on the traveled length: $\delta_d = \dfrac{\mid d_m - d_{VIO}\mid}{d_m}$,
- The difference between initial and final position: $\Delta_p = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$,

where d_m is the traveled distance measured from the ideal path, d_{VIO} is the estimated traveled distance, $[x_i, y_i]^T$ and $[x_f, y_f]^T$ are the initial and final 2D position. To confront the results,

at least in each environment it was defined a score relative to the environment.
First it was computed the average of the results between the pair of trials for each camera setting, then it was defined a score relative to the worst (highest) value for each metric in one environment:

$$s_i^r = \left(\frac{| \delta_{d,i}^r - \delta_{d,max}^r |}{\delta_{d,max}^r} + \frac{| \Delta_{p,i}^r - \Delta_{p,max}^r |}{\Delta_{p,max}^r} \right) \cdot 100,$$
$$s_i^c = \left(\frac{| \delta_{d,i}^c - \delta_{d,max}^c |}{\delta_{d,max}^c} + \frac{| \Delta_{p,i}^c - \Delta_{p,max}^c |}{\Delta_{p,max}^c} \right) \cdot 100,$$

where s_i^r is the score for case i (camera settings and algorithm) in the averaged results of room environment, $\delta_{d,max}^r$ is the maximum value of δ_d^r in the averaged results of the room environment and the rest is self explanatory.
This shows (tab. 1a and tab. 1b, lower value is better) that in the room environment OKVIS performs better, followed by VINS-Fusion and ROVIO. The results different in the corridor, where ROVIO is better, followed by OKVIS and VINS-Fusion. The trajectories corresponding to the best results for each environments are shown in figure 2.

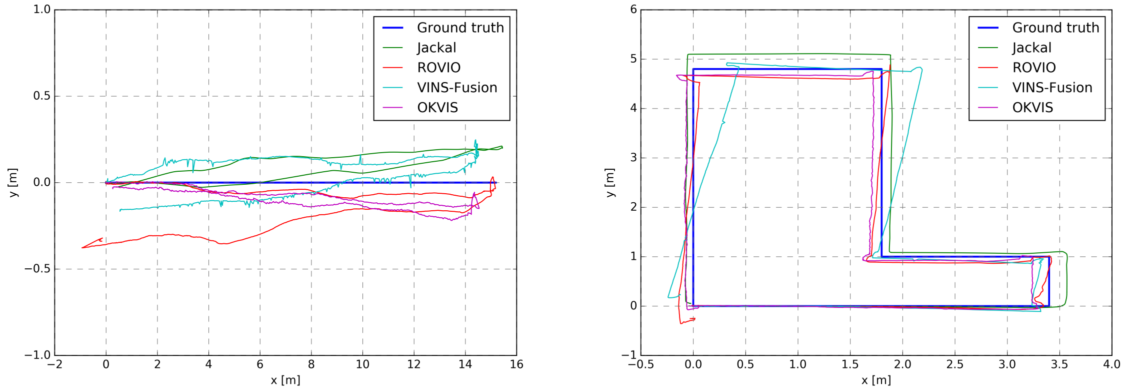
	ROVIO	VINS-Fusion	OKVIS
200 Hz - IR on	**63.0**	81.0	74.0
200 Hz - IR off	127.4	105.5	69.9
500 Hz - IR on	166.9	**69.6**	**63.9**
500 Hz - IR off	194.7	85.9	66.0

(A) Corridor environment.

	ROVIO	VINS-Fusion	OKVIS
200 Hz - IR on	181.2	157.8	58.9
200 Hz - IR off	**142.1**	97.1	**43.6**
500 Hz - IR on	166.1	159.0	47.9
500 Hz - IR off	178.3	**78.7**	45.5

(B) Room enviromet

FIGURE 1: VIO scores comparison. The lowest scores for each algorithm in every environment are highlighted.



(A) Corridor environment.

(B) Room enviromet

FIGURE 2: Trajectory comparison of the best settings for each algorithm.