Project no. VI20172020068

Tools and Methods for Video and Image Processing to Improve Effectivity of Rescue and Security Services Operations (VRASSEO)

Technical report 2018

Experimental flying platform

Alfredo Chávez Plascencia, Vítězslav Beran

Brno University of Technology
Faculty of Information Technology
Božetěchova 1/2
Brno, 612 66, Czechia

December 2018

Contents

1 Introduction 2

2 Platform architecture and devices 4

3 Integrated tested modules 8

4 Simulation and experimental tools 10

Abstract

This technical report describe the design, architecture and realization of the autonomous flight control system based on the NVIDIA Tegra TX2 platform, the ZED stereo camera and the ROS platform software. The purpose of the experimental platform is to enable the development and testing of new applications for efficient drone control using autonomous features, geography and height maps, 3D models and integration of real-world video data from the drone.

1 Introduction

Autonomous flying quadcopter of outdoor environments is a complex task due to the drone has to deal with the interaction and coordination of the different modules of autonomy and also with precise measurements of the environment, in other words the drone must have a good field of view of the environment in turn.

Camera pose estimation and camera map building is an important module in drone autonomous tasks. The drone must build the map and localize itself according to that map, this is called the chicken-egg problem. In order to be able to construct a 2D or a 3D environment and to localize the drone in such environment, a solution is needed. Simultaneous localization and mapping (SLAM) is an algorithm for localization and map making of a 2D and 3D indoor and outdoor environments [1]. In the literature, there are algorithms to solve the SLAM chicken-egg problem. For instance; real time appearance based on mapping (RTAB-Map) [2], large scale direct monocular SLAM (LSD-SLAM) [3] and ORB-SLAM [4]. A comparison method between LSD-SLAM and LSD-SLAM is carry out in [5] showing that LSD-SLAM is more suitable for robot navigation. However the RTAB-Map has better support for ROS and for this reason there is an ease in the integration of the current application.

Full drone autonomy should not be considered due to safety reasons, therefore the operator should be allowed to have as much as control as required during the drone's flight. Thus, a share control strategy is considered in this work. Then, the purpose of this work is to formulate, evaluate and implement a share control strategy algorithms that can allow the operator to be placed on of the top of the hierarchy of the drone's flight. In other words, at any time an at any moment the operator must have full control over the autonomous mode.

To this end, a TBS-Discovery (TBSD) experimental flying platform [1] has been equipped with a control command communication software interface PX4 and with a Nvidia-Jetson TX2, a ZED camera and a Pixhawk devices to carry out the automation algorithms that are part of the robot operating system (ROS) framework.

Moreover, drones do not fly intrinsically stable due to factors as high center of gravity, low moment of inertia, between other factors. Stability of a quadropter has been tackled using control strategies. For instance, [6, 7] use a non linear control strategy more precisely a state feedback linearization. Other control strategies also have been considered for stabilized the quadrotor like; proportional integral and derivative control (PID) [8], robust adaptive-fuzzy control used to minimize sinusoidal wind disturbance

---

[1] `https://github.com/NVIDIA-AI-IOT/redtail/wiki/Skypad-TBS-Discovery-Setup#assembly-2-jetson--j120-connections`

[9], among others. Despite of all control strategies, the PID is the most widespread control technique. And, in this context PX4 uses PID control. Also, there exists a tuning PID gains method for PX4 for all multicopter setups [2]. The method mainly consists in tuning the PID gains to achieve a good stabilization. In our case, the PID PX4 method is chosen as a stabilization method for the TBSD.

---

[2] `https://docs.px4.io/en/config_mc/pid_tuning_guide_multicopter.html`

2 Platform architecture and devices

The system architecture as depicted in Figure 1 consists of the following devices:

1. TBSD frame is a quick and straightforward to build crash resistant multirotor.
2. Lenovo ThinkPad L540 + Intel(R) Core(TM) i7-4712MQ CPU @ 2.30GHz running Ubuntu 16.04.5 LTS and ROS kinetic is used as a ground control station.
3. Pixhawk mini is an independent hardware for open source autopilots that provides a rapid implementation for a hiqh-quality and low-cost autopilot hardware designs for the academic, hobby and developer communities. Pixhawk supports the flight command control software stacks PX4®.
4. Eight channel pulse width modulator (PWM) board that distributes the control velocity commands to the motors.
5. PDB is a regulator board that supplies VDC to the Pixhawk mini.
6. 3DR GPS compass is a module that comes with the u-blox NEO-7 which is a set of positioning modules called global navigation satellite system (GNSS). These modules are receivers and can singly receive and track the following systems: global position system (GPS), global navigation satellite system (GLONASS) and galileo.
7. Nvidia-Jetson TX2 is a supercomputer of the size of a credit card. Its innovative technology makes it suitable for artificial intelligence (AI) and visual computing. Also, its size and power consumption make the module ideal for intelligent devices, like mobile robots, robot arms, drones, among others. The compatible OS are Windows 7, 8, 10 and Linux. It also supports ROS, OpenCV, Matlab as a third-party.
8. Auvidea J120 is a carrier board for the NVIDIA-Jetson TX1/TX2 and has the capability to expand the Nvidia-Jetson TX2 module in a super-mini-computer for desktop usage and for integration into UAVs and drones.
9. ZED camera is a stereo camera that provides high definition 3D video and depth perception of the environment.
10. TBS bulletproof electronic speed controller (ESC) 30A is an electronic device that controls the speed of the motor.
11. TBS discovery 900KV Motors + Graupner propellers 10×5in.

12. MZ12 Graupner (GR) radio controller (RC) is transmitter of signals that has to do with throttle, yaw, pitch and roll movements and also with different flight modes.
13. GR-12L RC device that receives the signals from the RC transmitter.
14. Wifi Antennas that receive wireless signals from the groundcontrol station.
15. 4S lipo battery, 14.8VCD, 3300mAh that supplies with power to the TBSD frame and all the devices.

The architecture shown in Figure 2 consists of three levels:

1. The low level architecture (A_{LL}) is in charge of receiving the speed reference motor signals from the Pixhawk and make the motors to follow them. The A_{LL} includes the motors over the ESCs and the interface with Pixhawk.
2. The middle level architecture (A_{LM}) has the task to interpret the control commands and transform them to velocity references. The A_{LM} consists of acceleromter, gyroscope, imu, gps, Nvidia-Jetson TX2 and the interface with the Pixhawk.
3. The high level architecture (A_{LH}) deals with the generation control commands for different flight modes. The A_{LH} consists of a mavros ROS package and Qgroundcontrol station.

Broadly speaking, the commmunication between the different devices is done using the PX4 software and the mavros. In other words, the communication between ROS-drone is done over the Pixhawk which uses the PX4 software which in turn mainly contains:

1. The flight control software, e.g. position controller, attitude estimator, autonomous flight, output driver for PWM.
2. Drivers, e.g. camera control, gps, imu, RC input, gimbal drivers.

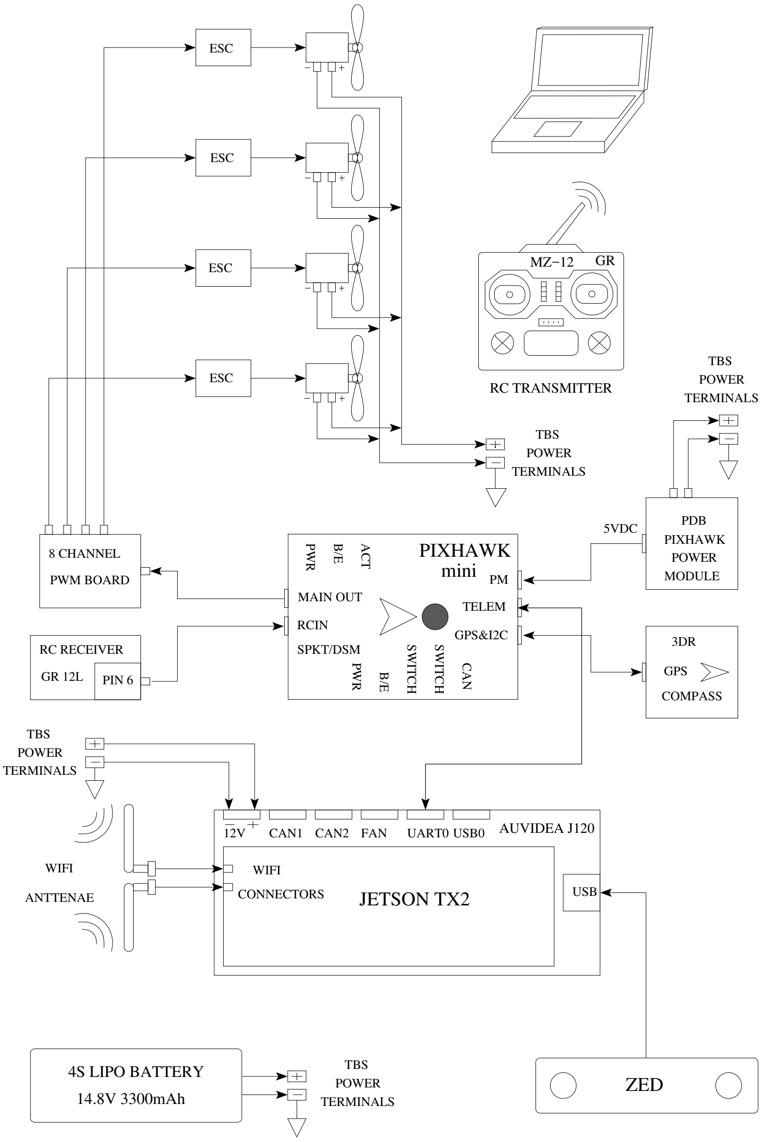


Figure 1: Devices of the architecture.

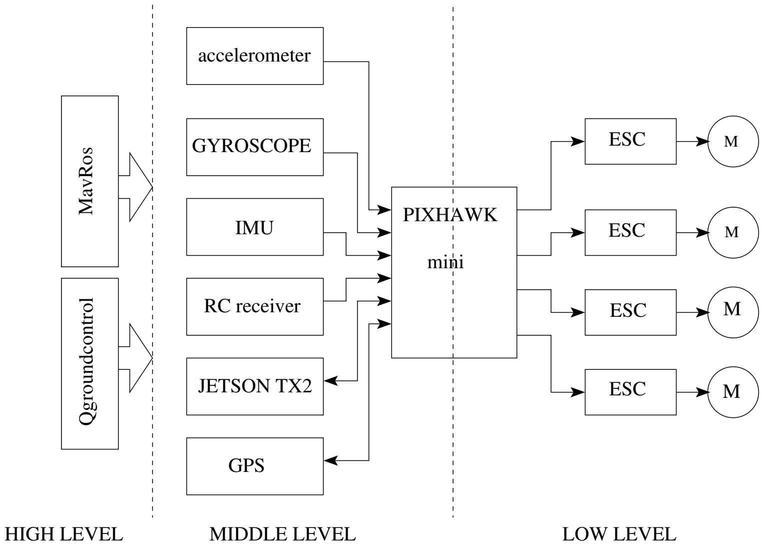


Figure 2: Levels of the architecture.

3 Integrated tested modules

The SDK-ZED from StereoLabs is the official driver for the ZED camera. Nonetheless, an interface is needed to communicate the ZED camera with ROS, in other words, to be able to use the SDK-ZED driver in ROS packages, like mavros. For this purpose, a zed_ros_wrapper ROS package [3] has been developed to wrap the ZED-SDK stereo camera with ROS. This package has been installed on the Nvidia-Jetson TX2 and also visualization of the right, left, stereo, depth images was done using the RVIZ ROS tool.

RTAB-MAP is a RGB-D SLAM approach based on a global loop closure detector with real-time constraints. This algorithm, as it has mentioned previously has been chosen because it has a full ROS support. The ROS warpper package is called rtabmap_ros [4]. Further more, this package has been installed and tested on the Nvidia-Jetson TX2 with the ZED camera. Figure 3 shows the ROS block diagram where the ZED camera is connected to the rtab_ros wrapper and this in turn is connected to the RTAB_MAP which produces the PointCloed2 and the input for the octomap_server.

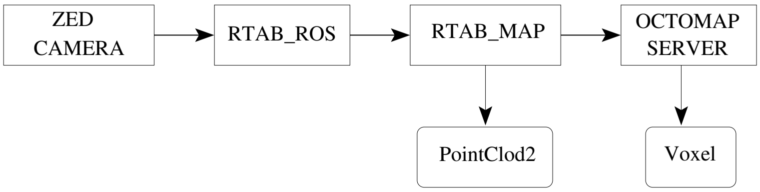


Figure 3: ROS modules.

For visualization, the ROS tool RVIZ has been used to visualized a PointCloud2 as well as a occupied voxels cells. Figure 4 shows the PointCloud2 of a laboratory-office where as Figure 5 shows the voxel occupied cells.

It is worth mention that a further experiment in building a 3D map of some indoor or outdoor environment is needed.

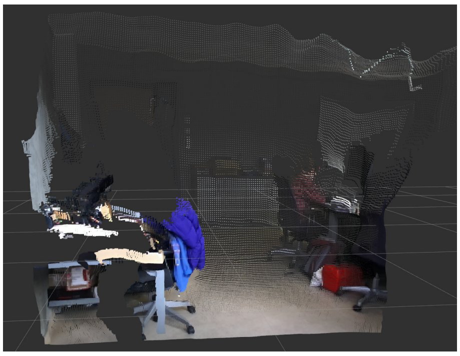


Figure 4: PointCloud2.

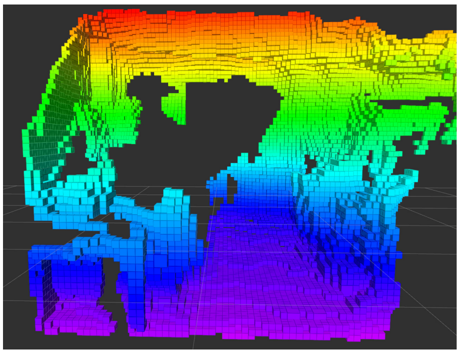


Figure 5: Occupied voxel cells.

---

[3] `http://wiki.ros.org/zed-ros-wrapper`

[4] `http://wiki.ros.org/rtabmap_ros`

In order to develop the share control strategy, automation algorithms based on PX4 and mavros need to be implemented and tested. Avoidance is a ROS package that uses PX4 computer algorithms packed in ROS nodes for depth sensor fusion, obstacle avoidance, path planning and octomap 3D map reconstruction [5]. The package comes with two different implementations; local planer and global planner. The global planner has been run and tested on a laptop. And, Gazebo and RVIZ ROS tools have been used for visualization. In the test, a launch file has been launch that puts the drone in offboard and armed modes, then it puts the drone in hoover mode. Afterwards the 2D-Nav-Goal RVIZ tab has been used to plan a trajectory. Then, the planned path is shown in RVIZ and the drone follow it, updating the map when obstacles are detected.

---

[5] `https://github.com/PX4/avoidance#global-planner`

4 Simulation and experimental tools

ORB-SLAM and LSD-SLAM have been evaluated in [5] using data set. The result of the evaluation have shown that ORB-SLAM is more accurate in rate frame performance than LSD-SLAM. In the other hand, the evaluation had shown that LSD-SLAM is much more suitable as an input for robotic navigation and path planning because the maps are more dense. Also, in that work the experiments are lacking the use of more realistic scenarios with a ground truth included.

Since one of the aims of this work is to simulate a system in realistic scenarios with a ground truth included, a 3D Gazebo-RVIZ virtual laboratory simulation package called ROS_quadrone has been created using and modifying existing ROS-packages. The package in turn uses the rtab_ros for 3D map making and octomap_server for viewing a voxel 3D map on RVIZ. A rapid exploring random tree star (RRT*) path planning ROS package [6] was slightly modified for our purposes of planning a 3D trajectory under a voxel map. The attitude position ROS_quadrotor_simulator package [7] was used to make the drone to follow the 3D path.

Figures 6 shows the virtual laboratory in Gazebo and the virtual drone with a mounted laser for odometry purposes. Figure 7 shows the RVIZ view of the voxel map and also the planned trajectory as green markers, we can also see the drone following the path.

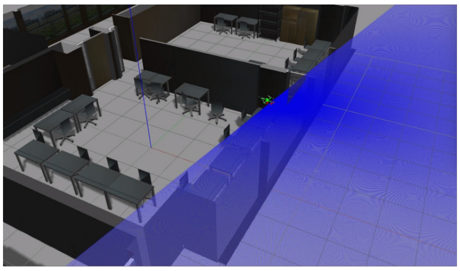


Figure 6: Gazebo.

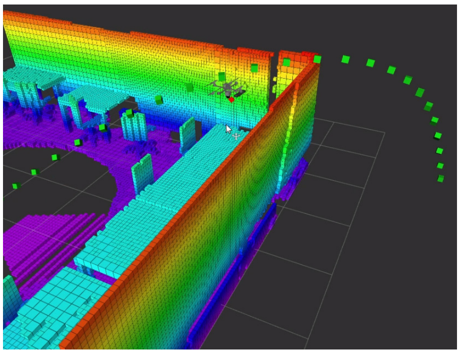


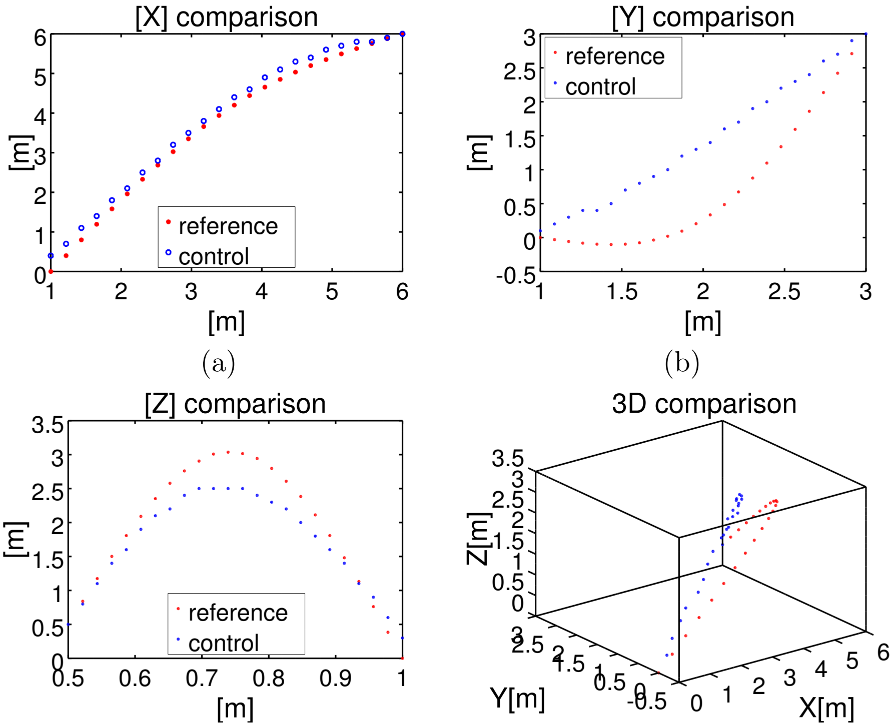
Figure 7: RVIZ.

Figures 8(a), 8(b) and 8(c) shows the comparison in the [X], [Y] and [Z] components respectively of the control and the reference path. Whereas Figure 8(c) shows the comparison 3D path of the control and the planned path. It can be seen that the control in the [X] component is more accurate than the control in the [Y] and [Z] components.

---

[6] `https://github.com/ayushgaud/path_planning`

[7] `https://github.com/wilselby/ROS_quadrotor_simulator/tree/master/quad_control/src/nodes`



(a) (b)

(c) (d)

Figure 8: (a) Comparison between X's components. (b) Comparison between Y's components. (c) Comparison between Zs components. (d) 3D comparison

Table 1 shows the mean and variance of the three components of the planned trajectory and the control. It can be seen the deviation is bigger in the $[Y]$ component compare with the $[X]$ and $[Z]$ components.

	Trajectory (n=24)		Control (n=24)	
	Mean	Variance	Mean	Variance
X	3.6419	3.5324	3.8292	3.3535
Y	0.76088	1.0227	1.4375	0.85027
Z	1.9156	0.89710	1.6917	0.50775

Table 1: The table shows the mean, variance of the planned trajectory as well as the control.

The Table 2 shows the technical data of the hardware software used for the simulation as well as the take took for the drone to achieved the goal.

Processor	Intel$^{\mathrm{R}}$ Core $^{\mathrm{TM}}$ i7-4712MQ CPU @ 2.30GHz × 8
Graphics	Intel$^{\mathrm{R}}$ Haswell Mobile
ROS version	kinetic
Gazebo version	version 7.0.0
Time	48sec

Table 2: Simulation platform.

References

[1] H. D. Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," in *IEEE Robotics and Automation Magazine*, vol. 13, pp. 07–10, 2006.

[2] M. Labbé and F. Michaud, "RTAB Map as an open source lidar and visual SLAM library for large scale and longterm online operation," in *Journal of Field Robotics*, 2018.

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *pringer International Publishing*, pp. 834–849, 2014.

[4] R. M. Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular slam system," vol. 31, pp. 1147–1163, 2015.

[5] V. Beran, M. Šolony, A. Herout, V. Bartl, P. Zemčík, S. Nosko, J. Zendulka, V. Bartík, and V. Fröml, "Selected Video-Processing Methods and System Architecture Design," tech. rep., Brno University of Technology, 12 2017.

[6] Holger, "Nonlinear control of a quadrotor micro-uav using," in *Proceedings of the 2009 IEEE International Conference on Mechatronics. Malaga, Spain, April 2009.*, 2009.

[7] A. Nemati and M. Kumar, "Nonlinear control of a tilting-quadcopter using feedback linearization based on motion control," in *Proceedings of the ASME Dynamic Systems and Control Division, DSCC 2014, San Antonio, USA*, 2014.

[8] N. Ives, R. Pacheco, D. de Castro Ribeiro Resende, and P. A. A. Magalhães, "Stability control of an autonomous quadcopter through pid control law," in *Int. Journal of Engineering Research and Application*, vol. 5, pp. 07–10, 2015.

[9] C. Coza and Macnab, "A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization," in *In Fuzzy Information Processing Society, Annual meeting of the North American*, pp. 454–458, 2006.