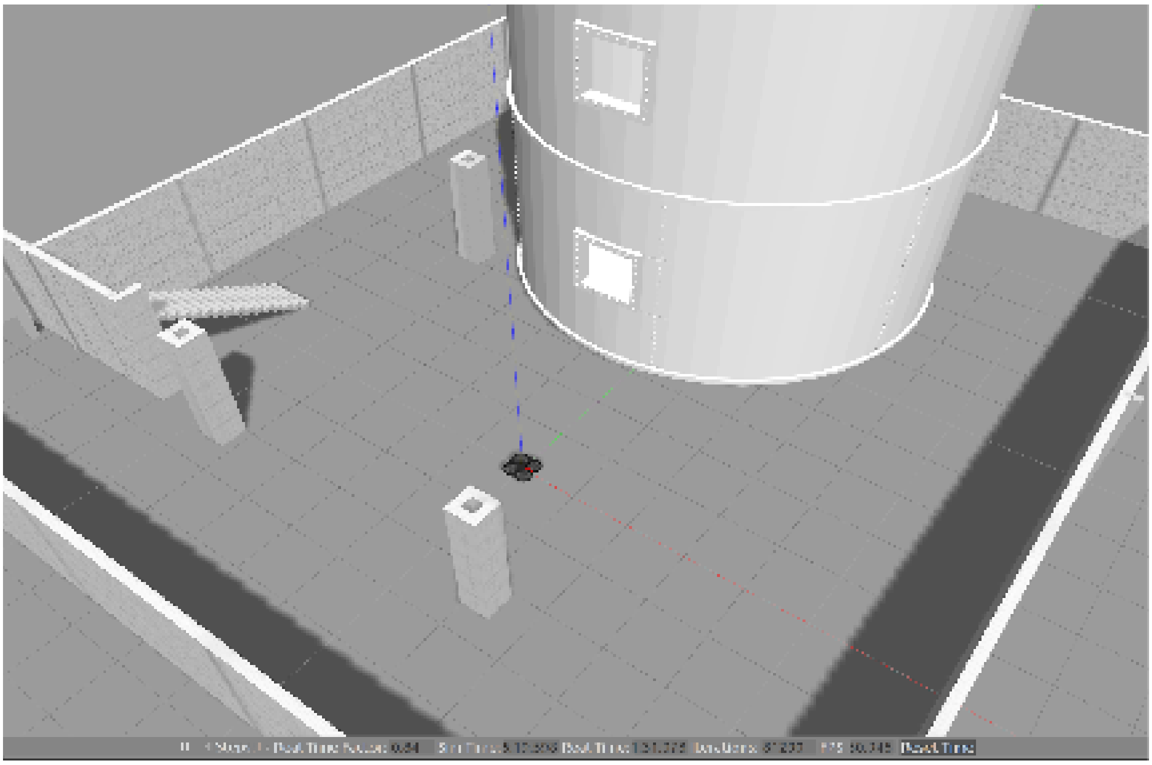
EN.530.707 - Robot System Programming

3D Visual SLAM and Path Planning using AR Drone



Authors: Tianyu Song, Huixiang Li, Wenhao Gu

Instructor: Prof. Louis Whitcomb

Date: May 16, 2018

Introduction

For indoor situations where GPS-based navigation is unavailable, the robot has to learn in an autonomous manner about its surroundings. Sensors commonly used for indoor navigation ranges from low-cost ultrasonic sensors that measure the unidirectional distance to the closest object to much more expensive laser measurement systems (LMS) or LiDARs that are able to produce a complete depth image of the surroundings, but come in at several thousands of dollars. One of the most cost-effective method used by robots to learn about their environment is to use optical cameras. Being light and energy efficient, cameras are ideal when applied to cheap consumer drones with limited payload and battery life. However, mono RGB camera lacks depth information associated with given image, which makes it challenging for the robot to build a 3D map.

On the other hand, Simultaneous Localization and Mapping (SLAM) for Unmanned Aerial Vehicles (UAVs) in the context of rescue and recognition navigation tasks in indoor environments has been a hot topic for several years. Usually, visual SLAM algorithm uses RGBD camera to generate the 3D map. LSD-SLAM [1, 2] is a novel, direct monocular SLAM technique developed by TUM: Instead of using keypoints, it directly operates on image intensities both for tracking and mapping. The camera is tracked using direct image alignment, while geometry is estimated in the form of semi-dense depth maps, obtained by filtering over many pixel-wise stereo comparisons.

With the map data provided by LSD-SLAM, the indoor path planning task can be then achieved. Due to project time limitation and the difficulties in trajectory generation with 6 degrees of freedom of the drone, we assume the drone as a planar joint with only 3 degrees of freedom, meaning that it can only translate and rotate in a plane.

Proposed Goals

This project focuses on a fusion of monocular vision and IMU to robustly track the position of an AR drone using LSD-SLAM (Large-Scale Direct Monocular SLAM) [1, 2] algorithm. The system consists of a low-cost commercial drone and a remote control unit to computationally afford the SLAM algorithms using a distributed node system based on ROS (Robot Operating System). Upon finishing this project, it is expected that we are able to reconstruct the 3D environment around AR drone and localize itself. In addition, using visual cues, the drone will be able to hold a given position despite the random disturbances that could be applied to the drone as well as navigate to a given position or follow a certain path autonomously and safely within the map built with LSD-SLAM. The built map will be displayed on the host computer. The drone then will be able to follow the generated path according to the octomap and avoid obstacles accordingly.

Software

Program architechture

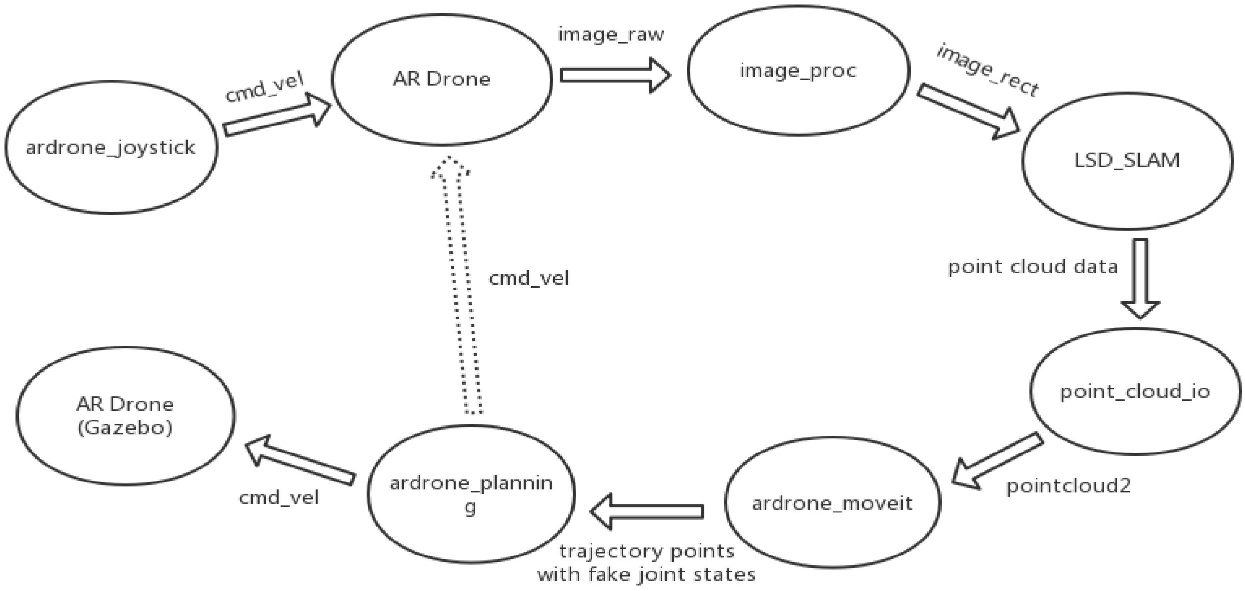


Figure 1. Program Architecture

Existing software employed by this project include the following:

1. ardrone_autonomy: ROS driver for Parrot AR-Drone 1.0 & 2.0 quadrotor. http://wiki.ros.org/ardrone_autonomy
2. LSD_SLAM: It is a novel approach to real-time monocular SLAM. It is fully direct (i.e. does not use keypoints / features) and creates large-scale, semi-dense maps in real-time on a laptop. https://vision.in.tum.de/research/vslam/lsdslam
3. tum_ardrone: It consists of three components: a monocular SLAM system, an extended Kalman filter for data fusion and state estimation and a PID controller to generate steering commands. http://wiki.ros.org/tum_ardrone
4. image_proc: This node rectify the raw image captured by the front camera of Ardrone. By our experiment, this step can significantly reduce the noise in the point cloud map generated by LSD-SLAM.

New software that we designed and coded for this project include the following:

1. cvg_sim_gazebo: Gazebo simulation for Hydro Lab.
2. ardrone_joystick: use Logitech joystick to publish cmd_vel to control the motion of AR Drone.
3. point_cloud_io: publish point cloud topic generated from LSD-SLAM.
4. ardrone_moveit: subscribe point cloud topic and convert point cloud data into octomap for visualization and do path planning, then publish trajectory with fake joint states.
5. ardrone_planning: subscribe trajectory with fake joints states from ardrone_moveit and convert them to drone's pose states, then publish cmd_vel to the AR Drone.

Hardware and Infrastructure

Existing hardware employed by this project include the following:

1. Laptops with ROS Kinetic and Ubuntu 16.04 installed
2. AR Drone: a low cost quadrotor developed by Parrot
3. Logitech Joystick: a low cost multi purpose joystick for teleporated control of the drone

Figure 2. Parrot AR Drone

Figure 3. Logitech Joystick

Sample Data Products

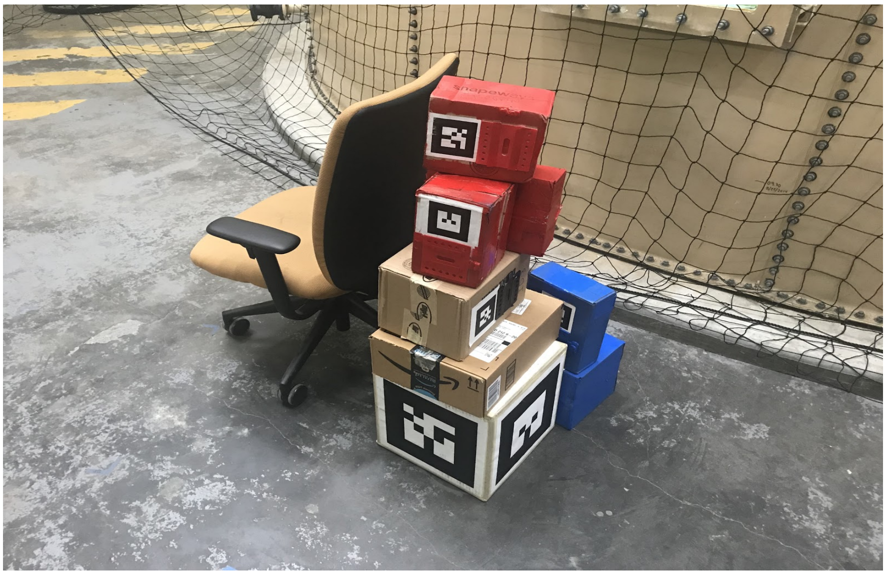


Figure 4. The real environment

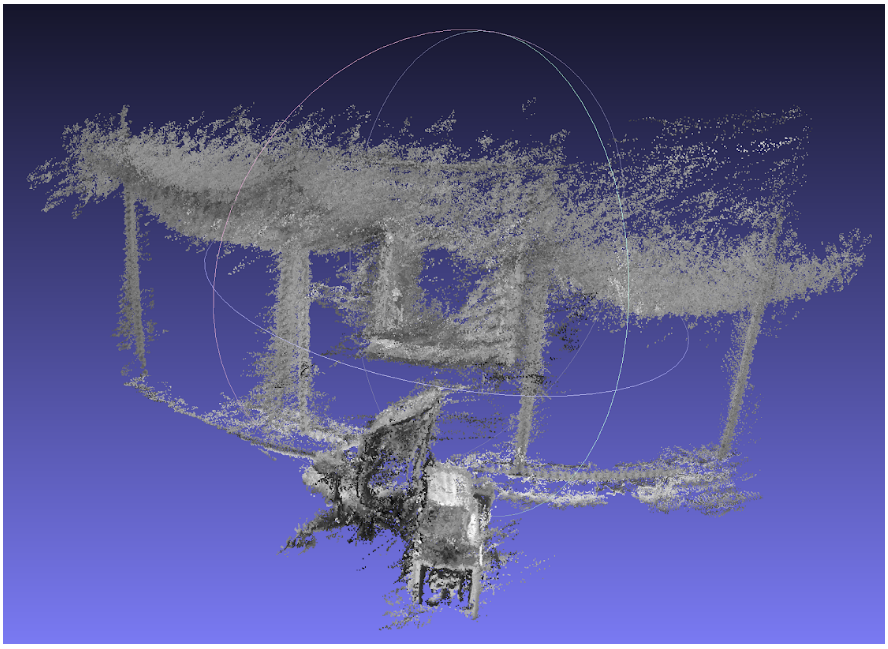


Figure 5. 3D Point Cloud Map Generated by LSD-SLAM

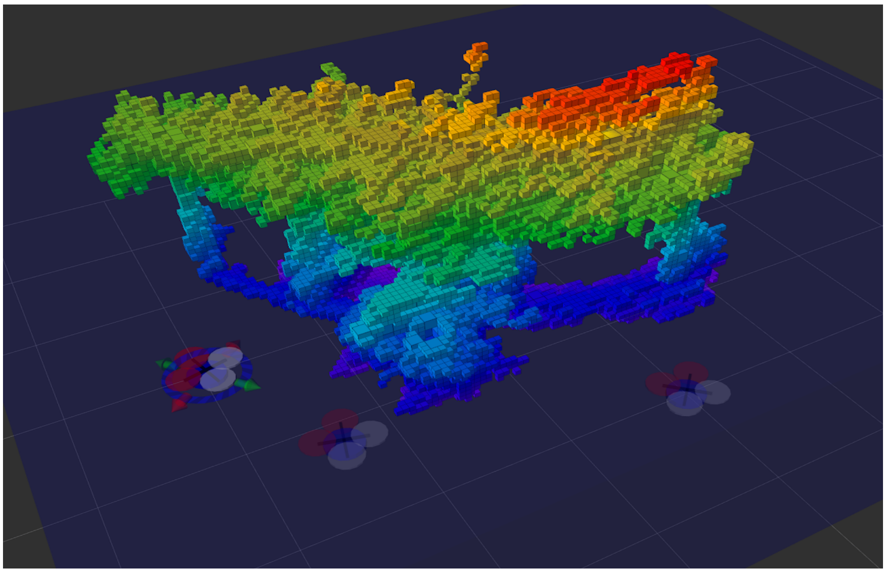


Figure 6. Octomap from Point Cloud for path planning

Figure 5 shows the point cloud data collected by the AR Drone using LSD-SLAM alogrithm in Hydro Lab. The data successfully reconstructed the environment with a chair and a few boxes as obstacles. Figure 6 shows octomap converted from the point cloud in Rviz to use MoveIt! do path planning and visulize the trajectoy.

Results and Demo Video

- Results

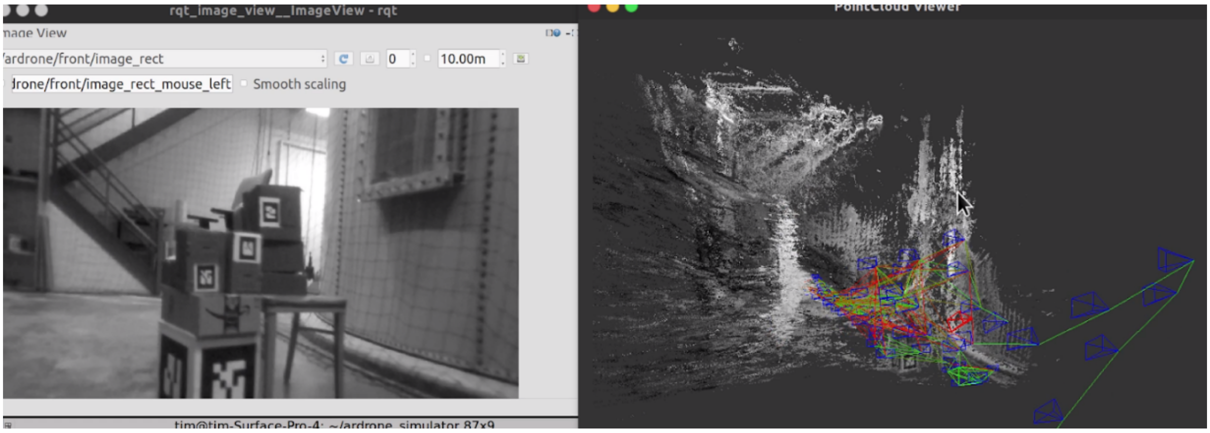

rqt_image_view__ImageView - rqt
nage View
'ardrone/front/image_rect
0
10.00m
lrone/front/image_rect_mouse_left
Smooth scaling
tim@tim-Surface-Pro-4: ~/ardrone_simulator 87x9
PointCloud Viewer

Figure 7. Generating point cloud (right) from monocular camera (left)

A simple environment and the result of map is shown above. The AR Drone is able to map the stairs and obstacles as well as localize itself. For more detailed explanations, you can refer to the attached video in the next section.

- Demo Video

  https://youtu.be/ysaGH9HKCDw

Lessons Learned

- Upgrade Indigo package to Kinetic package and migrate rosbuild file to catkin.
- Convert point cloud data to octomap and plug it into MoveIt! for path planning.
- For 3D path navigation, using MoveIt! would be better than Move_base. Typically, MoveIt! relies on pre-defined action files and action controller file (.yaml file) for translating the multi DOF trajectories produced by MoveIt!.
- MoveIt! does not have a good support for mobile robot. Therefore, we should treat the quadrotor as a multi DOF joint and use fake joint_states when connecting MoveIt! and the AR Drone.
- A server on the quadrotor need to service the move_group client in order to receive control commands output by the move_group node.

Suggestions

- Add a filter to reduce noise of point cloud data generated by LSD-SLAM in real time.
- Update the octomap periodically in Rviz.
- Use both PTAM and LSD-SLAM to improve the precision of pose estimation.

References

[1] J. Engel, T. Schops, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in Computer Vision–ECCV 2014, pp. 834–849, Springer, 2014.

[2] J. Engel, J. Sturm, and D. Cremers, “Camera-based navigation of a low-cost quadrocopter,” in IROS, 2012.

[3]“ardrone_autonomy.” ardrone_autonomy - ardrone_autonomy Indigo-Devel Documentation, ardrone-autonomy.readthedocs.io/en/latest/.

[4]“3D Perception/Configuration Tutorial.” Move Group Interface Tutorial, docs.ros.org/indigo/api/moveit_tutorials/html/doc/pr2_tutorials/planning/src/doc/perception_configuration.html.