

Indoor Environment Mapping Tutorials

Autonomous Mapping

Required ROS Packages

- For using Realsense R200 camera in Gazebo
 1. realsense_gazebo_plugin^[2] - [1]
- For registering the depth image to colour image frame
 1. depth_image_proc - [1]
 2. image_common - [1]
- For generation of point clouds from colour and registered depth image
 1. rtabmap_ros - sudo apt-get install ros-indigo-rtabmap-ros
- For frontier based exploration
 1. autonomous_exploration - [1]
- For path planning
 1. move_it - sudo apt-get install ros-indigo-moveit
- For keyboard control of the ARDrone
 1. cvg_sim_gazebo - [1]

Required Files

- `ardrone_realsense.launch` (`realsense_gazebo_plugin/launch`) - Launch Gazebo7 and load the simulated world
- `register.launch` (`realsense_gazebo_plugin/launch`) - Launch a nodelet to register the depth image to the colour image frame
- `pub_camera_info.py` (`realsense_gazebo_plugin/scripts`) - Run a script to publish fake depth and color camera info for the simulated cameras
- `rtabmap_pcl.launch` (`realsense_gazebo_plugin/launch`) - Launch nodelet to convert depth and color image to pointcloud
- `ardrone_get_odometry.py` (`cvg_sim_gazebo/scripts`) - Run a script to fetch pose of the ardrone in Gazebo and publish the tf
- `moveit.launch` (`roslaunch move_it/launch`) - Launch the MoveIt! path planner
- `send_goal.py` (`realsense_gazebo_plugin/scripts`) - Start the script to send goals to MoveIt!
- `server.py` (`realsense_gazebo_plugin/scripts`) - Start the actionlib server to execute the waypoints from MoveIt!
- `client.py` (`realsense_gazebo_plugin/scripts`) - Start the actionlib client to execute the waypoints from MoveIt!
- `keyboard.py` (`cvg_sim_gazebo/scripts`) - Start the keyboard tele-op to control the drone in Gazebo
- `fbet.launch` (`realsense_gazebo_plugin/launch`) - Launch the node to generate goals for autonomous mapping

Procedure

1. Launch `ardrone_realsense.launch` present in `realsense_gazebo_plugin` package to start the simulated world along which has the ARDrone with a Realsense R200 camera mounted on it
2. Launch `register.launch` present in `realsense_gazebo_plugin` package to register the depth image stream to the colour image stream
3. Run the script `pub_camera_info.py` to publish fake camera metadata for the simulated Realsense R200 camera
4. Launch `rtabmap_pcl.launch` present in `realsense_gazebo_plugin` package to generate point clouds from the depth and colour images
5. Run the script `ardrone_get_odometry.py` present in `cvg_sim_gazebo` package to fetch pose of the ardrone in Gazebo and to publish the corresponding transform (tf)
6. Launch `moveit.launch` present in `move_it` package to start the MoveIt! path planner
7. Run the scripts `server.py` and `client.py` present in `realsense_gazebo_plugin` package to start the actionlib server and client to execute the waypoints from MoveIt!
8. Launch `fbet.launch` present in `realsense_gazebo_plugin` package to generate goals for autonomous mapping
9. Start the ARDrone in simulation using the keyboard tele-op script (`keyboard.py`) present in `cvg_sim_gazebo` package

Appendix

1. Github Link for packages - https://github.com/eYSIP-2017/eYSIP-2017_Indoor-Environments-Mapping-using-UAV
2. Link to install Gazebo7 - https://github.com/eYSIP-2017/eYSIP-2017_Indoor-Environments-Mapping-using-UAV/blob/master/bash_scripts/install_gazebo7.sh
3. Bash Script to launch all nodes and scripts for autonomous mapping - https://github.com/eYSIP-2017/eYSIP-2017_Indoor-Environments-Mapping-using-UAV/blob/master/bash_scripts/autonomous_mapping.sh
4. Full video link - <https://youtu.be/kXyV3OpbWo8>
5. Video for explanation of the autonomous mapping algorithm - <https://youtu.be/Ow4pZIDPhkY>

THANK YOU!