

Record a dataset for RGB-D SLAM

In order to record a dataset for RGB-D SLAM, it is important to install the freenect drivers! Therefore, follow '**Kinect camera ROS driver installation instructions**' first! Because of the dependency issues explained in these instructions, I will use the freenect driver that was installed on my laptop.

- Power the Erle-Copter and connect the Kinect camera to the Erle-Copter battery. I soldered a DC-jack to the battery and adapted the Kinect power cable in order to do so. The Kinect USB cable has to be connected to your laptop.
- Connect your laptop to the Erle-Copter WiFi.
- Set the Erle-Copter as ROS Master. In this portfolio at '**/Tech report/Code/Shell scripts**', you can find the '**set_ros_master.sh**' script that does this! Run this script in a terminal, close this terminal and open a new one.

- Set the ROS IP on your laptop to 0.0.0.0:

export ROS_IP=0.0.0.0

- Launch the freenect driver on your laptop, by running:

roslaunch freenect_launch freenect.launch

- I mounted the Kinect camera upside down, below the Erle-Copter base. Also, the camera is tilted downwards with an angle of 23 degrees (0.40 radians). We will update the ROS /tf tree, so that our map in RGB-D SLAM would not be upside down. For this purpose, I created the **kinect_transformer** ROS package, which can be found at **/Tech report/Code/ros_catkin_ws/src**. Copy this folder into the **src** folder in your own ROS workspace and build it by going to your ROS workspace and running:

catkin_make --pkg kinect_transformer

After building the package, run the transformation:

roslaunch kinect_transformer transformer

This launchfile transforms the /camera_link frame relative to the /base_link frame. You can change the 'x y z roll pitch yaw' translations in the **transformer.launch** file if needed. As RGB-D SLAM subscribes to /tf by default, the algorithm will get fused sensor data of the visual odometry and the Erle-Copter odometry. This will result in a more accurate trajectory estimate. You can visualize the /tf tree by running

roslaunch tf view_frames

- Now, we are ready to record a dataset. Start recording a rosbag by executing:

rosbag record -b 2048 /tf /camera/rgb/image_raw /camera/rgb/camera_info /camera/depth/image_raw /camera/depth/camera_info

- As we are using visual SLAM algorithms, we need to keep as many visual features in the image as possible while recording a dataset. I found that the best way to do this is spinning around 360 degrees in the middle of the room while holding the Erle-Copter high up. After that, do a coastal navigation of the room. Repeat this when entering a new room.
- When you are done recording, go to the rosbag terminal and type CTRL+C