2. Handheld LiDAR Mapping

It is recommended to use our provided virtual machine, as it may not run due to environment differences.

1. Install dependency libraries. Taking the ros-melodic version as an example, enter the following in the terminal:

```
sudo apt install ros-melodic-moveit ros-melodic-moveit-visual-tools ros-melodic-kdl-* ros-melodic-joint-state-publisher-gui ros-melodic-trac-ik liborocos-kdl-dev ros-melodic-teleop-twist-keyboard ros-melodic-moveit-resources ros-melodic-navigation ros-melodic-gmapping ros-melodic-hector-slam ros-melodic-slam-karto ros-melodic-robot-state-publisher ros-melodic-geographic-msgs ros-melodic-libuvc-* ros-melodic-rtabmap-ros libavformat-dev libavcodec-dev libswresample-dev libswscale-dev libavutil-dev libsdl1.2-dev ros-melodic-pointcloud-to-laserscan ros-melodic-mbf-msgs ros-melodic-mbf-costmap-core ros-melodic-costmap-converter ros-melodic-bfl ros-melodic-serial ros-melodic-teleop-twist-joy ros-melodic-laser-proc ros-melodic-rosserial-arduino ros-melodic-rosserial-python ros-melodic-rosserial-server ros-melodic-rosserial-client ros-melodic-rosserial-msgs ros-melodic-amcl ros-melodic-map-server ros-melodic-urdf ros-melodic-xacro ros-melodic-interactive-markers ros-melodic-octomap* ros-melodic-joy* ros-melodic-dwa-local-planner ros-melodic-multirobot-map-merge python-catkin-tools python3-dev python3-catkin-pkg-modules python3-numpy python3-yaml build-essential ros-melodic-imu-tools ros-melodic-cartographer*
```

2. Starting the Mapping Command

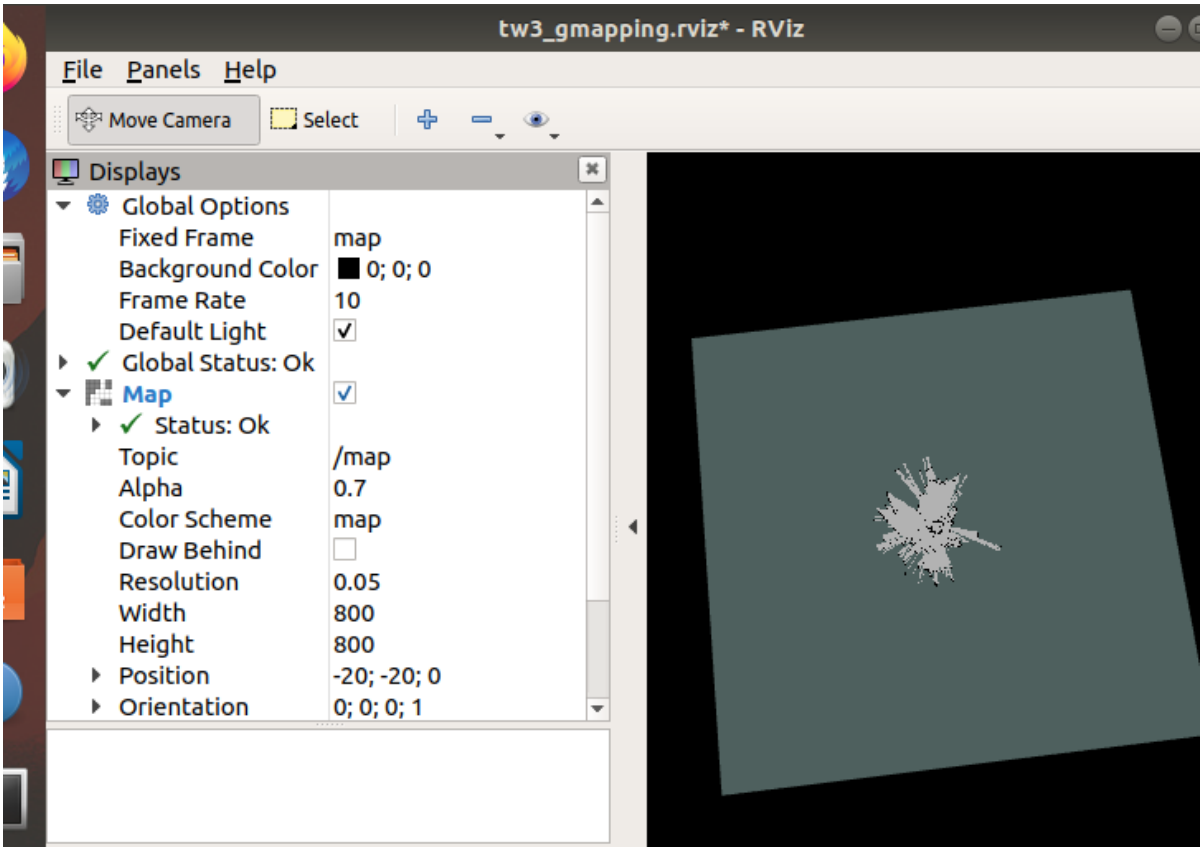
Taking gmapping as an example, enter the following in the terminal:

```
roslaunch gmapping_test tw3_gmapping.launch
```

It's normal to see this error message at the beginning of the program, because the TensorFlow conversion failed initially but will be successful later.

```
process[ydlidar_lidar_publisher-1]: started with pid [8254]
[ INFO] [1764574971.978471900]: YDLIDAR ROS Driver Version: 1.0.2
process[base_footprint_to_laser-2]: started with pid [8255]
process[rf2o_laser_odometry-3]: started with pid [8260]
process[slam_gmapping-4]: started with pid [8267]
process[rviz-5]: started with pid [8274]
[ INFO] [1764574972.005584084]: Initializing RF2O node...
[ERROR] [1764574972.023507046]: "base_footprint" passed to lookupTransform arg
ment target_frame does not exist.
[2025-12-01 15:42:52][info] SDK initializing
[2025-12-01 15:42:52][info] SDK has been initialized
[2025-12-01 15:42:52][info] SDK Version: 1.2.19
[ INFO] [1764574973.024094682]: Listening laser scan from topic: /scan
[ WARN] [1764574973.192457839]: Waiting for laser_scans....
[ WARN] [1764574973.358999010]: Waiting for laser_scans....
[2025-12-01 15:42:53][info] Connect elapsed time 1401 ms
[2025-12-01 15:42:53][info] Lidar successfully connected [/dev/ttyUSB0:230400]
[ WARN] [1764574973.525580729]: Waiting for laser_scans....
[ WARN] [1764574973.692486506]: Waiting for laser_scans....
[ WARN] [1764574973.858930294]: Waiting for laser_scans....
[ WARN] [1764574974.025570978]: Waiting for laser_scans....
[2025-12-01 15:42:54][info] Lidar running correctly! The health status good
[2025-12-01 15:42:54][info] Current Lidar Model Code 152
[2025-12-01 15:42:54][info] Baseplate device info
```

The following screen indicates successful operation.



3. Saving the Map

Taking gmapping as an example, the command to save the map is:

```
rosrun map_server map_saver -f ~/ydlidar_ws/src/gmapping_test/maps/test_map
```

```
yahboom@yahboom-virtual-machine:~$ rosrun map_server map_saver -f ~/ydlidar_ws/
src/gmapping_test/maps/test_map
[ INFO] [1764577156.289590163]: Waiting for the map
[ INFO] [1764577156.544235233]: Received a 800 X 800 map @ 0.050 m/pix
[ INFO] [1764577156.544299463]: Writing map occupancy data to /home/yahboom/ydl
idar_ws/src/gmapping_test/maps/test_map.pgm
[ INFO] [1764577156.557233266]: Writing map occupancy data to /home/yahboom/ydl
idar_ws/src/gmapping_test/maps/test_map.yaml
[ INFO] [1764577156.557473605]: Done
```

The map will be saved to the folder /home/yahboom/ydlidar_ws/src/gmapping_test/maps/, consisting of a PGM image and a YAML file.

test_map.yaml

```
image: /home/yahboom/ydlidar_ws/src/gmapping_test/maps/test_map.pgm
resolution: 0.050000
origin: [-20.000000, -20.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

Parameter Explanation:

- image: Path to the map file, can be an absolute or relative path.
- resolution: Map resolution, meters per pixel.
- origin: 2D pose (x, y, yaw) of the bottom left corner of the map. Here, yaw represents a counter-clockwise rotation (yaw=0 indicates no rotation). Many parts of the current system ignore the yaw value.
- negate: Whether to invert the meaning of white/black and free/occupied (the interpretation of the threshold is unaffected).
- occupied_thresh: Pixels with an occupation probability greater than this threshold are considered fully occupied.
- free_thresh: Pixels with an occupation probability less than this threshold are considered completely free.

4. Algorithm information reference website

4.1. Graphing algorithm

Gmapping: http://wiki.ros.org/gmapping/

hector_slam: http://wiki.ros.org/hector_slam

hector_slam/Tutorials: http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot

hector_mapping: http://wiki.ros.org/hector_mapping

karto: http://wiki.ros.org/slam_karto

Cartographer: https://google-cartographer.readthedocs.io/en/latest/

Cartographer ROS: https://google-cartographer-ros.readthedocs.io/en/latest/

4.2 Autonomous Mapping and Navigation

rrt_exploration: http://wiki.ros.org/rrt_exploration

rrt_exploration/Tutorials: http://wiki.ros.org/rrt_exploration/Tutorials

4.3 Saving Maps

map_server: https://wiki.ros.org/map_server