Lidar basics(For SLAM lidar)

Slam lidar tutorial materials: http://www.slamtec.com/cn/Support

Lidar technology email address: support@slamtec.com

Lidar wiki: http://wiki.ros.org/rplidar

LiDAR SDK: https://github.com/Slamtec/rplidar_sdk

LiDAR ROS: https://github.com/Slamtec/rplidar_ros

Lidar tutorial: https://github.com/robopeak/rplidar_ros/wiki

- For different models of radar

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker
tutorial
~/run_docker.sh
```

Before using lidar, you need to declare the [RPLIDAR_TYPE] variable in advance in the [.bashrc] file according to different radar models. Open the [.bashrc] file

```
sudo vim ~/.bashrc
```

If there is no sentence below in [.bashrc], you need to add it manually according to the purchased radar model. If there is this sentence, modify the radar model directly. For example: SLAM a1 lidar

```
export RPLIDAR_TYPE=a1 # a1, a2, a3, s1, s2
```

Note: For rosmaster series cars equipped with s2l radar, the radar startup method is the same as s2. Just change it to s2 here.

After modification, refresh the environment variables

```
source ~/.bashrc
```

1.1. Overview

Single-line lidar refers to a radar whose line beam emitted by the laser source is a single line. It can be divided into triangular ranging and TOF lidar. It is mainly used in the field of robotics. It has fast scanning speed, strong resolution and high reliability. Compared with multi-line lidar, single-line lidar responds faster in angular frequency and sensitivity, so it is more accurate in measuring distance and accuracy of obstacles.

1.2. SLAM radar components

Taking SLAM Technology's single-line lidar as an example, it is mainly composed of four core components: laser, receiver, signal processing unit and rotating mechanism.

RPLIDAR adopts a coordinate system that follows the left-hand rule. The front of the sensor is defined as the x-axis of the coordinate system. The origin of the coordinate system is the rotation center of the ranging core. The rotation angle increases with clockwise rotation. The specific coordinate system definition is shown in the figure below: (For details, please see the supporting manual)

A1	A2, A3
:---------- -:	:---------- ---:
S1	**S2**

1.2.1. Laser

The laser is the laser emitting mechanism in lidar. During operation, it lights up in a pulsed manner. SLAM Technology's RPLIDAR A3 series radar lights up and goes out 16,000 times per second.

1.2.2. Receiver

After the laser emitted by the laser hits an obstacle, the reflected light will be concentrated on the receiver through the lens group through reflection from the obstacle.

1.2.3. Signal processing unit

The signal processing unit is responsible for controlling the emission of the laser and processing the signals received by the receiver. Based on this information, the distance information of the target object is calculated.

1.2.4. Rotating mechanism

The above three components constitute the core components of measurement. The rotating mechanism is responsible for rotating the above-mentioned core components at a stable speed to scan the plane and generate real-time floor plan information.

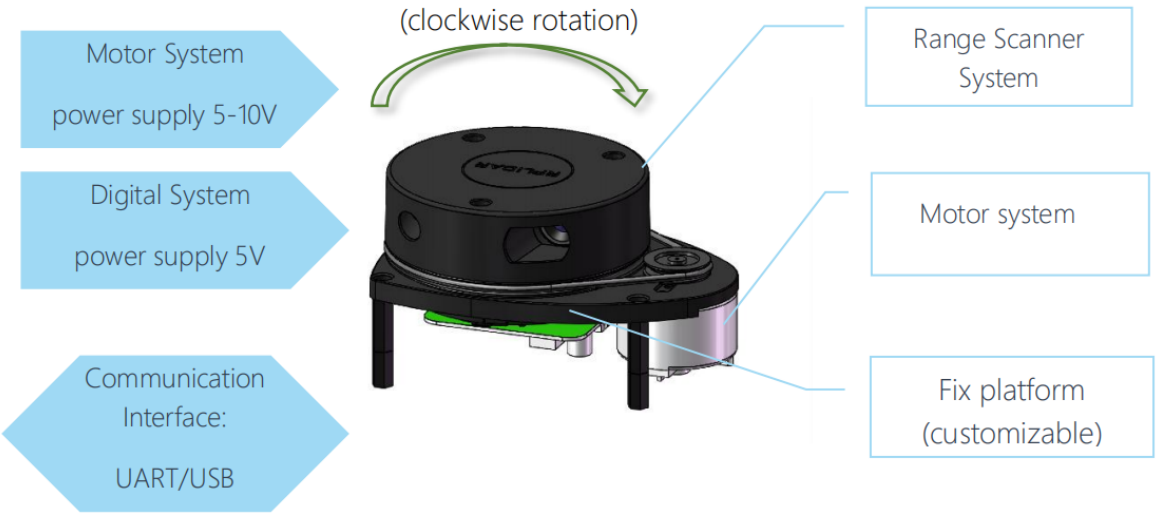
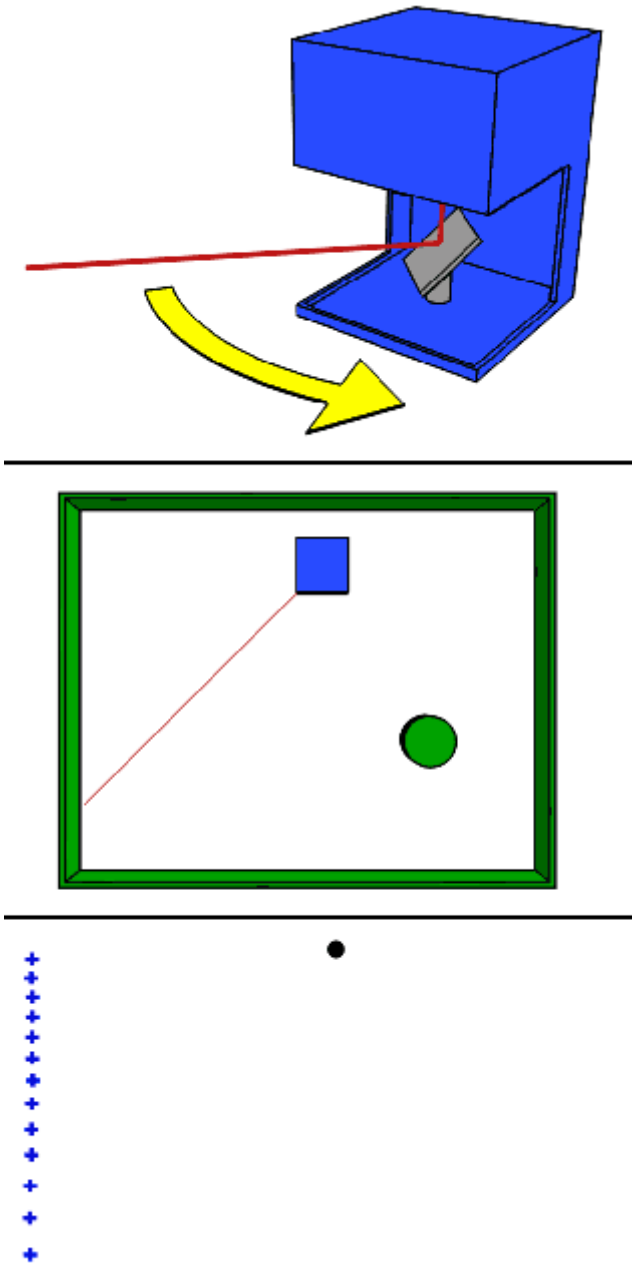


Figure 1-1 RPLIDAR A1 System Composition

1.3. Principle of single-line lidar

The working principle of the single-wire mechanical rotating radar is as shown in the figure below:

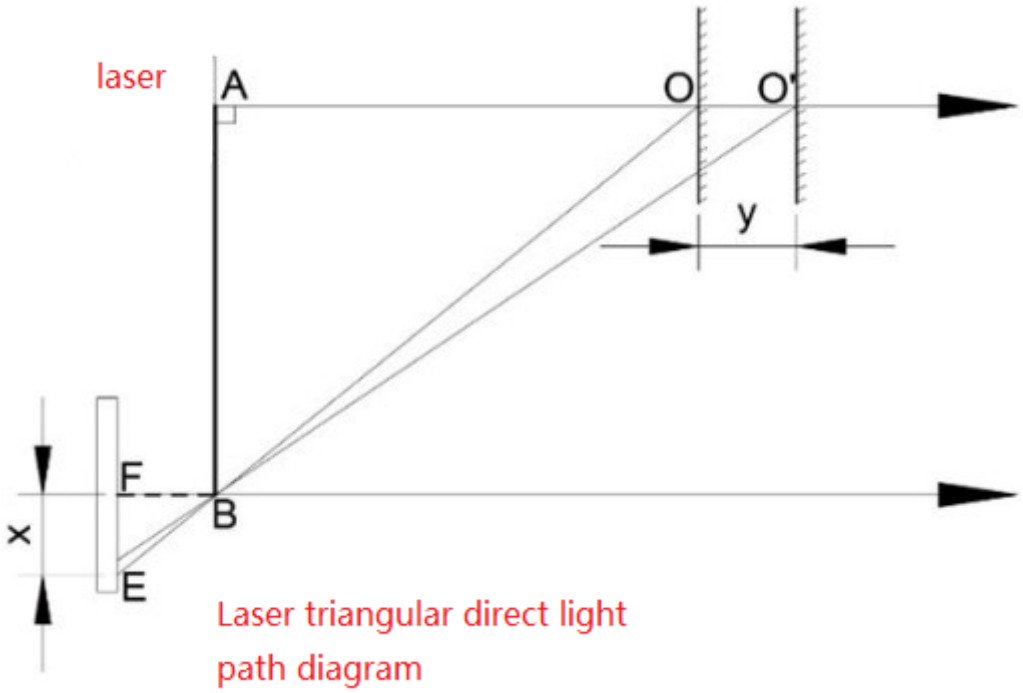


1.3.1. Trigonometric ranging method

The laser triangulation ranging method mainly uses a beam of laser to illuminate the measured target at a certain incident angle. The laser is reflected and scattered on the target surface. At another angle, a lens is used to converge and image the reflected laser. The spot is imaged on the CCD (Charge-coupled Device, photosensitive coupling component) on the position sensor. When the measured object moves along the direction of the laser, the light spot on the position sensor will move, and its displacement corresponds to the movement distance of the measured object. Therefore, the distance between the measured object and the baseline can be calculated from the light spot displacement distance through algorithm design. value. Since the incident light and the reflected light form a triangle, the geometric triangle theorem is used to calculate the spot displacement, so this measurement method is called the laser triangulation ranging method.

According to the angular relationship between the incident beam and the normal line of the surface of the measured object, the laser triangulation ranging method can be divided into two types: oblique type and direct type.

1. Direct type

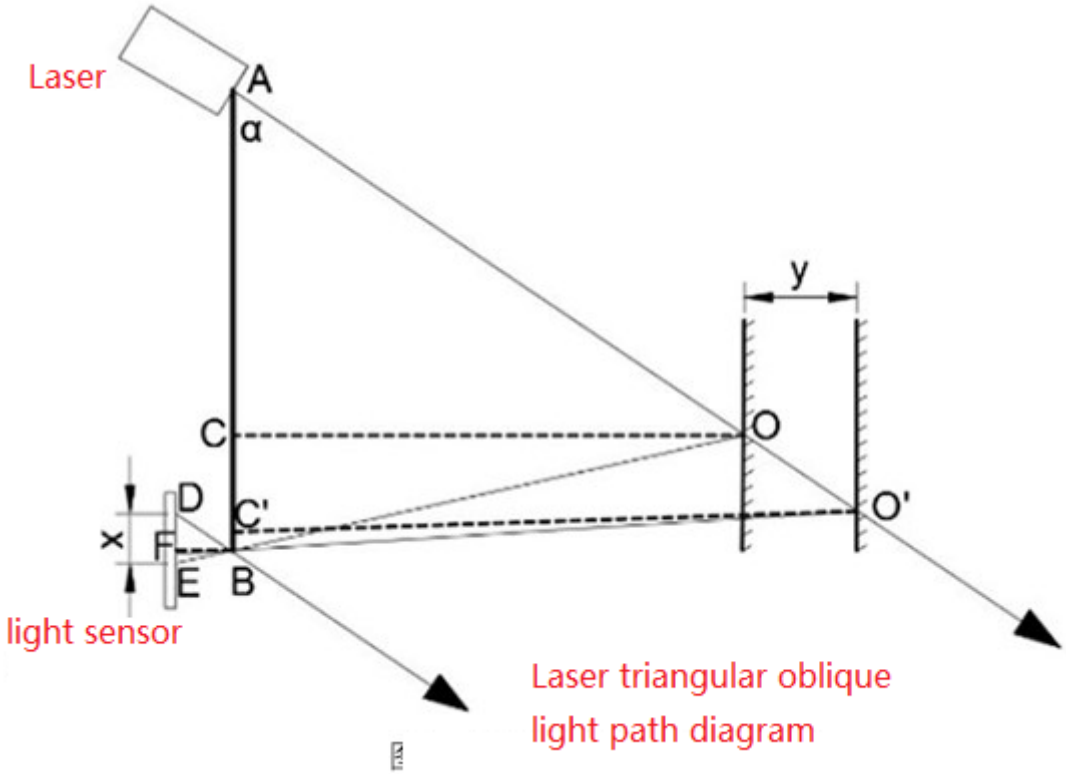


Laser triangular direct light path diagram

As shown in Figure 1, when the laser beam is vertically incident on the surface of the object to be measured, that is, when the incident light is collinear with the normal line of the surface of the object to be measured, it is a direct laser triangulation method.

2. Oblique shot type

When the angle between the incident laser beam and the normal line of the surface of the object being measured is less than 90° in the optical path system, the incident mode is oblique. The optical path diagram shown in Figure 2 is a laser triangulation oblique optical path diagram.



Laser triangular oblique light path diagram

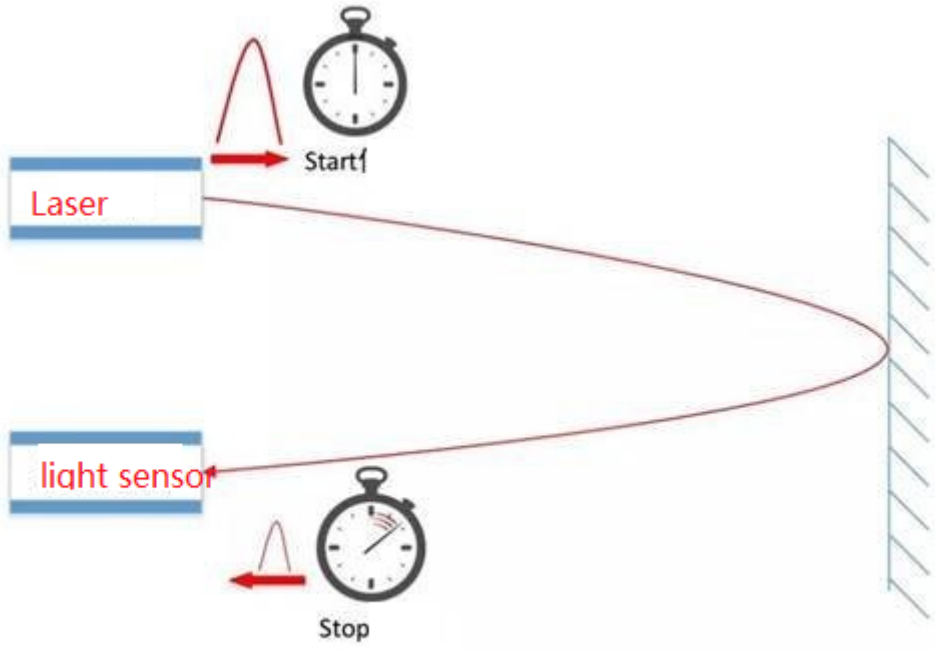
The laser emitted by the laser is incident on the surface of the object being measured at a certain angle with the normal line of the object surface. The reflected (scattered) light is concentrated through the lens at B and is finally collected by the photosensitive unit.

Whether it is the direct or oblique laser triangulation ranging method, it can achieve high-precision, non-contact measurement of the measured object, but the resolution of the direct type is not as high as that of the oblique type.

SLAM Technology's RPLIDAR series lidar also uses the oblique laser triangulation ranging method. During each ranging process, the RPLIDAR series lidar will emit a modulated infrared laser signal. The reflection generated by the laser signal after hitting the target object will be received by the RPLIDAR visual acquisition system, and then processed by the DSP embedded inside the RPLIDAR. The device solves the problem in real time, and the distance value between the illuminated target object and the RPLIDAR and the current angle information will be output from the communication interface. Driven by the motor mechanism, the ranging core of RPLIDAR will rotate clockwise, thereby achieving 360-degree all-round scanning and ranging detection of the surrounding environment.

1.3.2. TOF time-of-flight ranging method

TOF lidar is based on measuring the flight time of light to obtain the distance of the target. Its working principle is mainly as follows: a modulated laser signal is emitted through a laser transmitter. The modulated light is received by the laser detector after being reflected by the object being measured. The distance to the target can be calculated by measuring the phase difference between the emitted laser and the received laser. .



Working principle diagram of TOF laser

Under the condition of distant objects, its measurement accuracy remains accurate and stable. At the same time, TOF radar is not inferior in its ability to resist light interference due to its ultra-short light pulse characteristics. It can achieve stable ranging and high-precision mapping even under strong light of 60Klx outdoors.

Generally speaking, triangular ranging lidar and TOF lidar have their own difficulties in implementation. In principle, TOF radar has a longer ranging distance. In some occasions where distance is required, TOF radar is the most common, while The manufacturing cost of triangular ranging lidar is relatively low, and its accuracy can meet most industrial-grade civilian requirements, so it has also attracted much attention in the industry.

1.4. Lidar A1M8

- For Model A1M8 Only

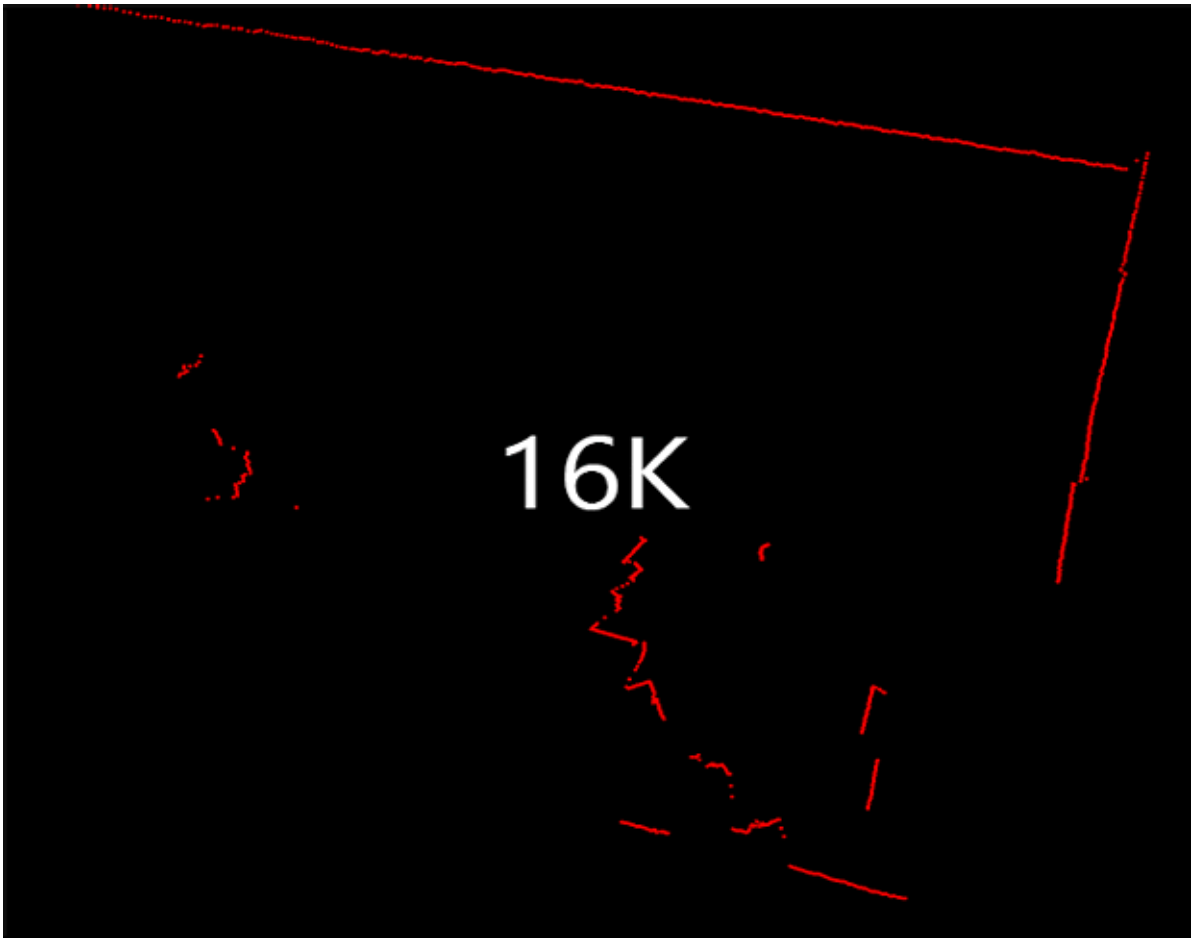
Item	Unit	Min	Typical	Max	Comments
Distance Range	Meter(m)	TBD	0.15-12	TBD	White objects
Angular Range	Degree	n/a	0-360	n/a	
Scan Field Flatness	Degree	-1.5		1.5	
Distance Resolution	mm	n/a	<0.5 <1% of the distance	n/a	<1.5 meters All distance range*
Angular Resolution	Degree	n/a	≤1	n/a	5.5Hz scan rate
Sample Duration	Millisecond(ms)	n/a	0.125	n/a	
Sample Frequency	Hz	n/a	≥8000	8010	
Scan Rate	Hz	1	5.5	10	Typical value is measured when RPLIDAR A1 takes 360 samples per scan

Figure 2-1 RPLIDAR A1 Performance

As can be seen from the figure above, parameters such as measurement radius, sampling speed, rotation speed, and angular resolution are important indicators of radar performance.

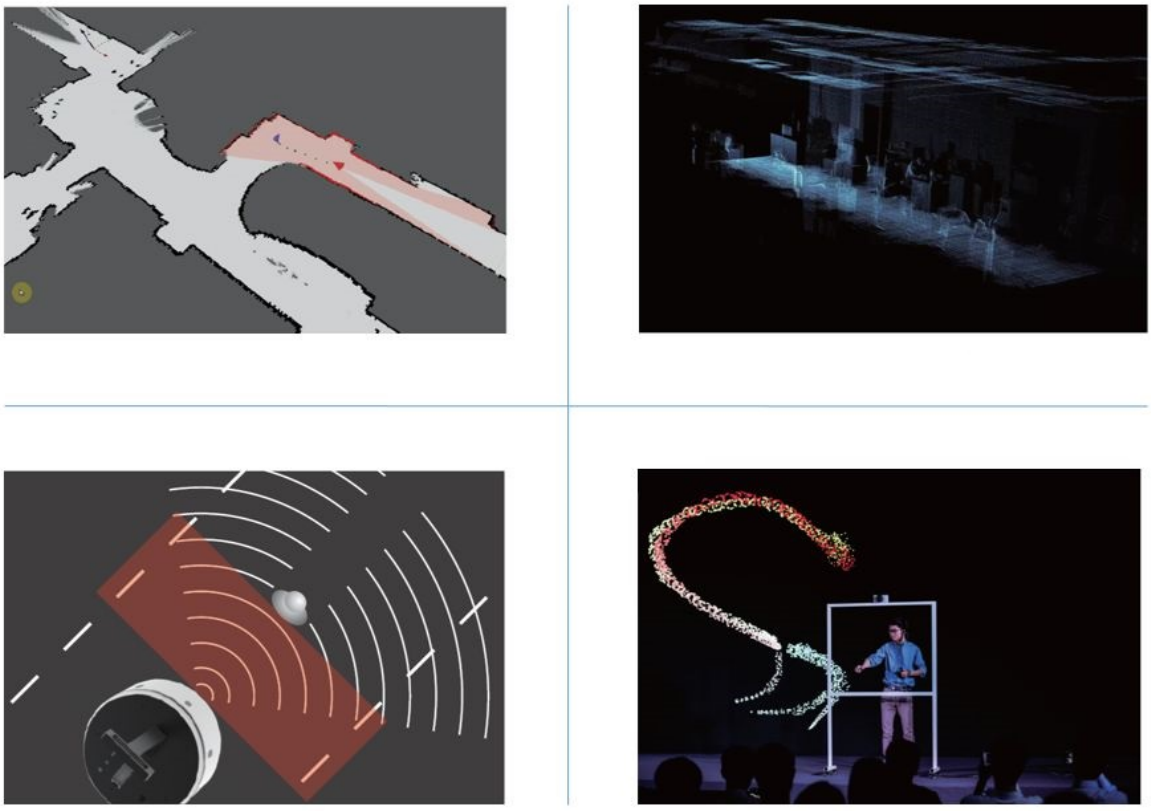
Indicator	**Description**
Ranging radius	Radar measuring distance range
Ranging sampling rate	How many ranging outputs are performed in one second
Scanning frequency	How many times the radar scans in one second
Angular resolution	Angle step between two adjacent ranging
Measurement resolution/accuracy	Minimum distance change can be perceived

A higher **scanning frequency** can ensure that the robot equipped with lidar can move at a faster speed and ensure the quality of map construction. However, increasing the scanning frequency is not as simple as accelerating the rotation of the lidar's internal scanning motor. Correspondingly, it is necessary to increase the ranging sampling rate. Otherwise, when the sampling frequency is fixed, faster scanning speed will only reduce the angular resolution. In addition to ranging distance and scanning frequency, parameters such as measurement resolution and mapping accuracy are equally important to lidar performance. These are important parameters to ensure that the robot can have stable performance.



1.5. Application scenarios

Thanks to the advancement of lidar technology, the measurement radius, ranging frequency, distance resolution and angular resolution of lidar have been greatly improved, which can help various applications obtain larger scenes and richer contour information. It plays an indispensable and important role in many fields such as robot autonomous positioning and navigation, space environment surveying and mapping, and security and defense.



1.6. Function package rplidar ros

Clone this project into the src folder of your workspace and run catkin_make to build rplidarNode and rplidarNodeClient.

1.6.1. Remap USB serial port

When starting the radar function package, you may encounter that the serial port permissions are not executable. There are two solutions: 1) Add permissions directly; 2) Remap the USB serial port

1. Directly add permission method

This method only works this time.

Check the permissions of rplidar serial port:

```
ls -l /dev |grep ttyUSB
```

Add write permission: (such as /dev/ttyUSB0)

```
sudo chmod 777 /dev/ttyUSB0
```

2. Remapping USB serial port method

This method works long term.

In the rplidar_ros function package path, install USB port remapping

```
./scripts/create_udev_rules.sh
```

Re-plug the LiDAR USB interface and use the following command to modify the remapping:

```
ls -l /dev | grep ttyUSB
```

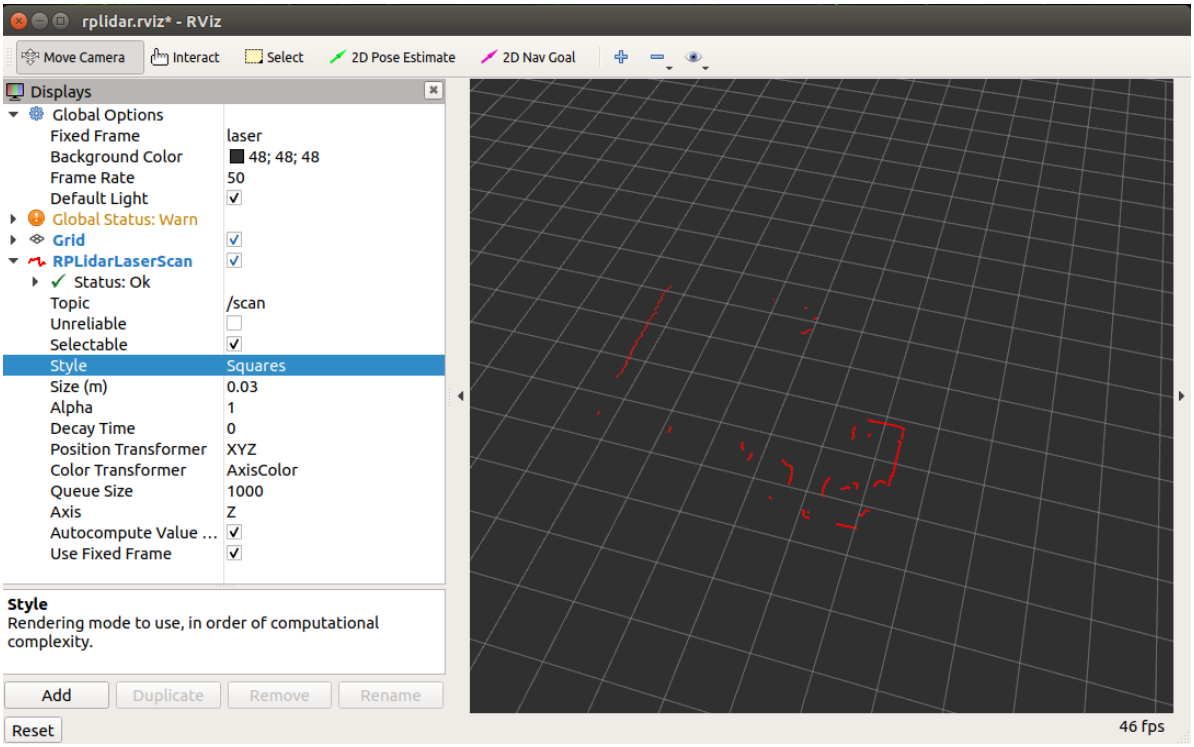
```
lrwxrwxrwx  1 root   root           7 Sep 14 15:04 rplidar -> ttyUSB0
crwxrwxrwx  1 root   dialout 188,   0 Sep 14 15:04 ttyUSB0
jetson@jetson-yahboom:~$
```

1.6.2. Code testing

- Run the rplidar node and view it in rviz

```
roslaunch rplidar_ros view_rplidar.launch
```

You should see rplidar's scan results in rviz.



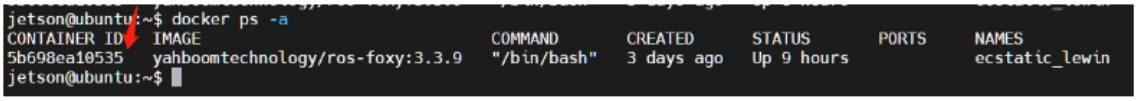
- Run the rplidar node and see with the test application

```
roslaunch rplidar_ros rplidar.launch # Launch radar
```

<PI5 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```



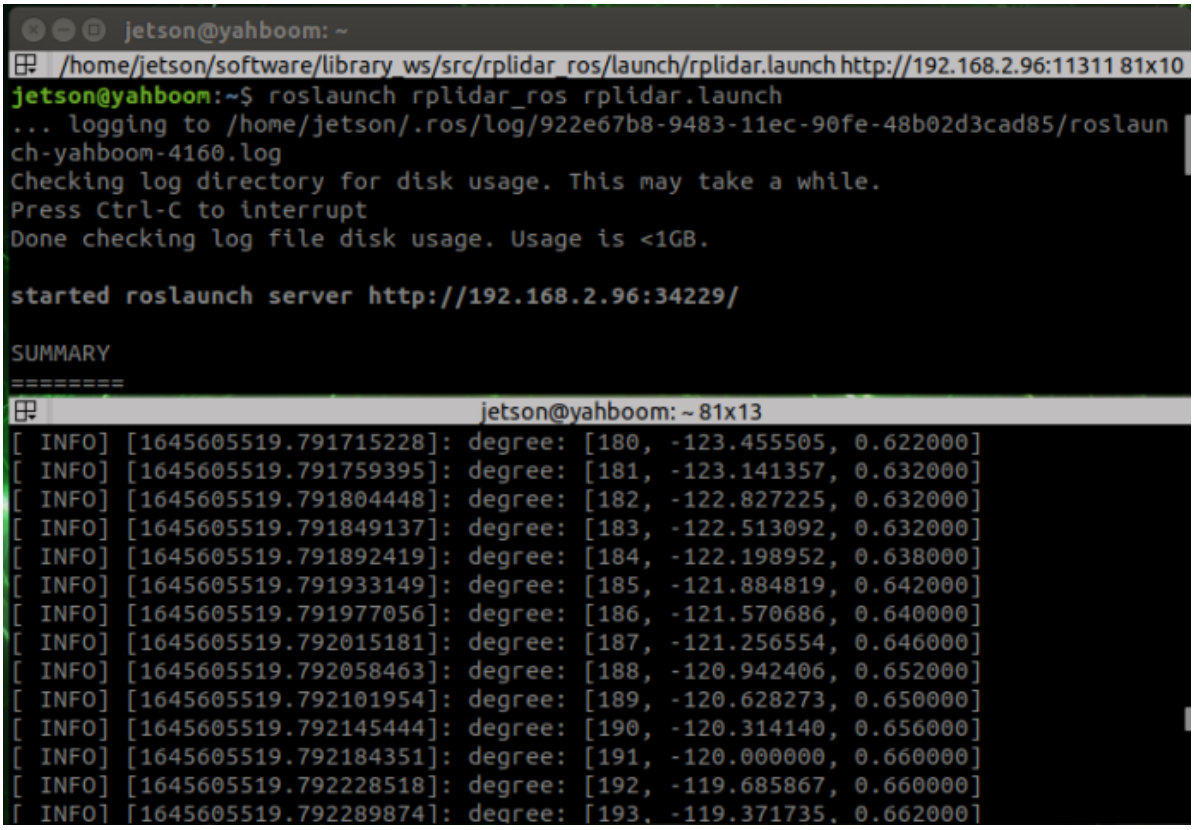
2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```

After successfully entering the container, you can open countless terminals to enter the container.

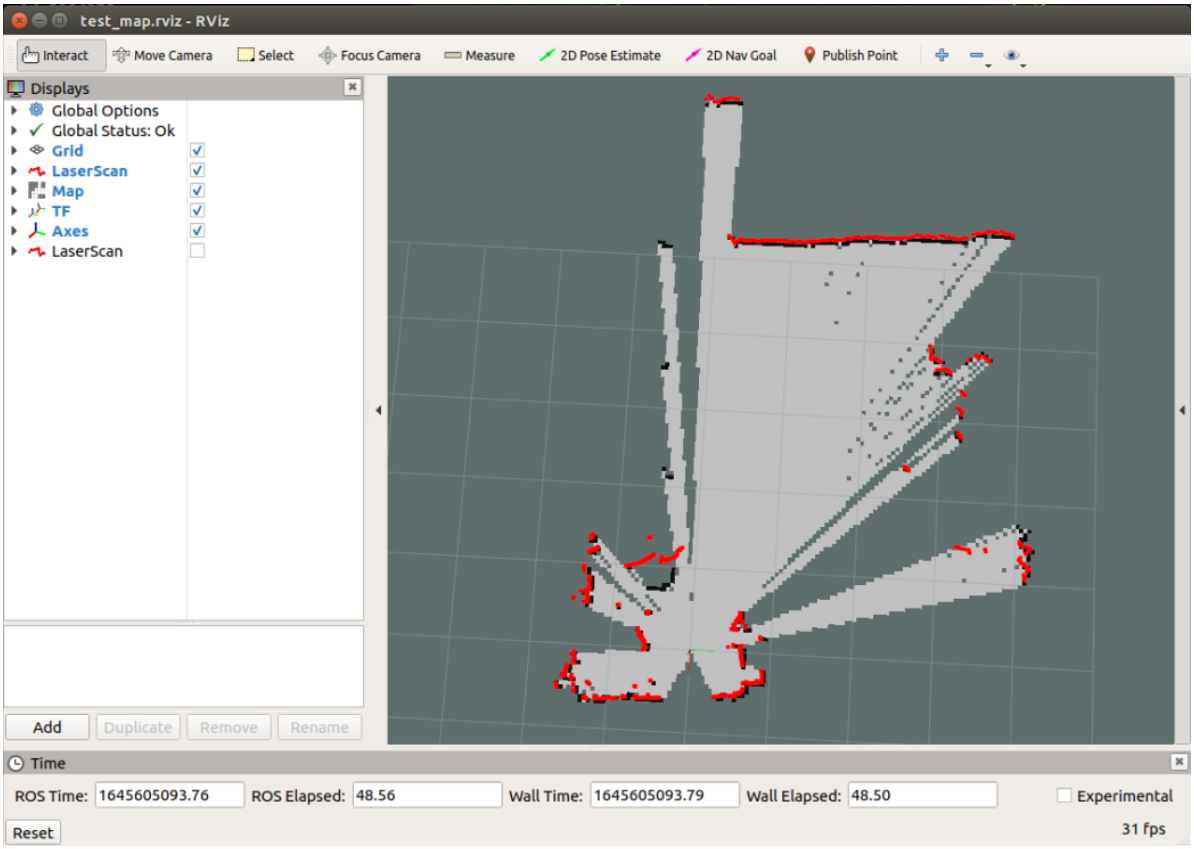
```
rosrun rplidar_ros rplidarNodeClient # Get and print radar data
```

You should see rplidar's scan results in the console

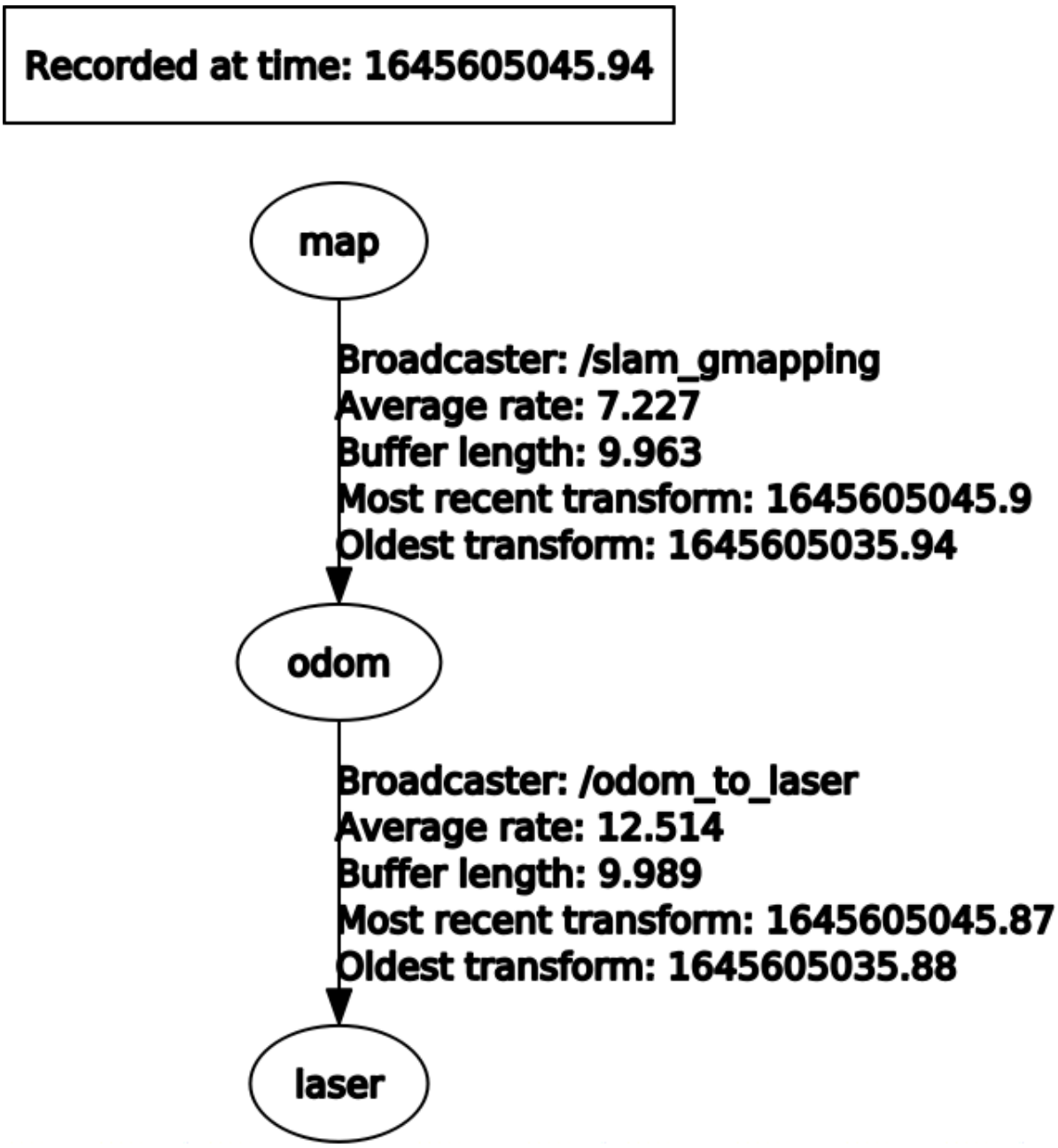


1.6.3. Map construction test

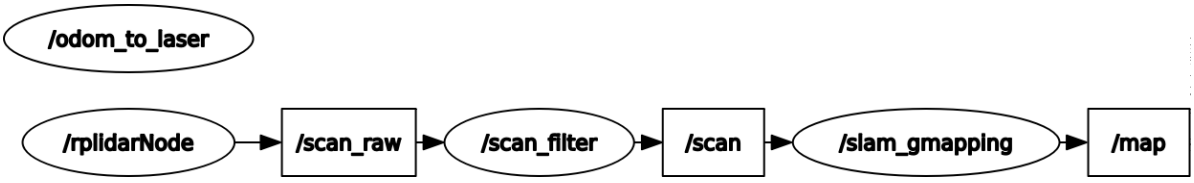
```
roslaunch rplidar_ros test_gmapping.launch
```




```
rosrun rqt_tf_tree rqt_tf_tree
```




```
rosrun rqt_graph rqt_graph
```



1.7. Source code analysis

From the test in section [1.6], it can be seen that the lidar data does not have 360° data. There is a small gap. That is because the data behind the lidar is blocked.

rplidar.launch file

```
<launch>
     <arg name="lidar_type" value="$(env RPLIDAR_TYPE)" doc="lidar_type type
[a1,a2,a3,s1,s2]"/>
     <arg name="frame_id" default="laser"/>
     <arg name="shielding_angle" default="30"/>
```

```
    <!-- scan filtering node -->
    <node name="scan_filter" pkg="rplidar_ros" type="scan_filter.py"
output="screen" respawn="true">
        <param name="shielding_angle" type="double" value="$(arg
shielding_angle)"/>
    </node>
    <node name="rplidarNode" pkg="rplidar_ros" type="rplidarNode"
output="screen" respawn="true">
        <param name="serial_port" type="string" value="/dev/rplidar"/>
        <param name="serial_baudrate" type="int" value="115200" if="$(eval
arg('lidar_type') == 'a1')"/>
        <param name="serial_baudrate" type="int" value="115200" if="$(eval
arg('lidar_type') == 'a2')"/>
        <param name="serial_baudrate" type="int" value="256000" if="$(eval
arg('lidar_type') == 'a3')"/>
        <param name="serial_baudrate" type="int" value="256000" if="$(eval
arg('lidar_type') == 's1')"/>
        <param name="serial_baudrate" type="int" value="1000000" if="$(eval
arg('lidar_type') == 's2')"/>
        <param name="frame_id" type="string" value="$(arg frame_id)"/>
        <param name="inverted" type="bool" value="false"/>
        <param name="angle_compensate" type="bool" value="true"/>
        <param name="scan_mode" type="string" value="Sensitivity" if="$(eval
arg('lidar_type') == 'a3')"/>
        <param name="scan_mode" type="string" value=" " unless="$(eval
arg('lidar_type') == 'a3')"/>
        <remap from="scan" to="scan_raw"/>
    </node>
</launch>
```

- shielding_angle parameter: the angle for shielding radar data, range [0, 360], which can be adjusted according to the actual situation.
- gmapping is only applicable to points where the number of two-dimensional laser points in a single frame is less than 1440. If the number of laser points in a single frame is greater than 1440, then problems such as [[mapping-4] process has died] will occur. Therefore, when using S2 lidar, the number of S2 points needs to be diluted.

If you do not need to filter radar data, comment or delete the following content in the [rplidar.launch] file

```
    <!-- scan filtering node -->
    <node name="scan_filter" pkg="rplidar_ros" type="scan_filter.py"
output="screen" respawn="true">
        <param name="shielding_angle" type="double" value="$(arg
shielding_angle)"/>
    </node>
```

Then modify the [rplidarNode] node

```
<remap from="scan" to="scan_raw"/> <!-- Delete -->
<remap from="scan" to="scan"/> <!-- Add -->
```

Users can handle it according to the actual situation.