Odin1重定位地图获取手册

Odin1提供重定位功能，已实现与MANIFOLD手持扫描仪的数据联动。目前我们提供3个方案：

- 1.使用Odin采集olx数据，通过Mindcloud处理后使用“导出地图”功能导出bin文件用于Odin的重定位；
- 2.使用留形手持扫描仪（留形机，如：Q9000）采集lx数据，通过Mindcloud处理后使用“导出地图”功能导出bin文件用于Odin的重定位；
- 3.使用Odin SLAM模式生成bin文件用于Odin的重定位。详细操作如下。

1. Odin录制olx文件（选择1）

1.1 Odin录制olx文件（Odin固件版本要求0.10.0以上，驱动版本0.9.0以上）

1.1.1 修改/catkin_ws/src/odin_ros_driver/config/control_command.yaml文件中参数，打开recorddata: 1 选择custom_map_mode: 0 (odom模式)：

代码块

```
recorddata: 1          # 0: off; 1: on
***
custom_map_mode: 0     # 0: Odometry mode 1: SLAM mode 2: Relocalization mode
```

1.1.2 运行驱动

代码块

```
cd ~/catkin_ws
# 以下以ros2为例，ros1请输入对应的指令
source install/setup.bash
ros2 launch odin_ros_driver odin1_ros2.launch.py
```

1.1.3 采集地图

采集完成后Ctrl+c结束驱动，即可在/catkin_ws/src/odin_ros_driver/recorddata中找到对应的olx文件。将整个文件夹拷贝到装有MindCloud（版本≥0.2.8）的电脑上进行第3步处理。

2. Q9000录制 lx 数据（选择2）

2.1 Q9000录制lx文件

- 采集数据

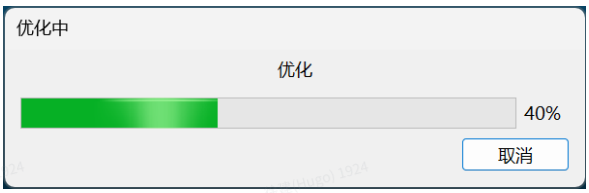
- 将录制的lx文件拷贝到装有MindCloud（版本≥0.2.8）的电脑上进行第3步处理

2.2 导入MindCloud Studio并保存pcd & bin文件（需要在windows系统）

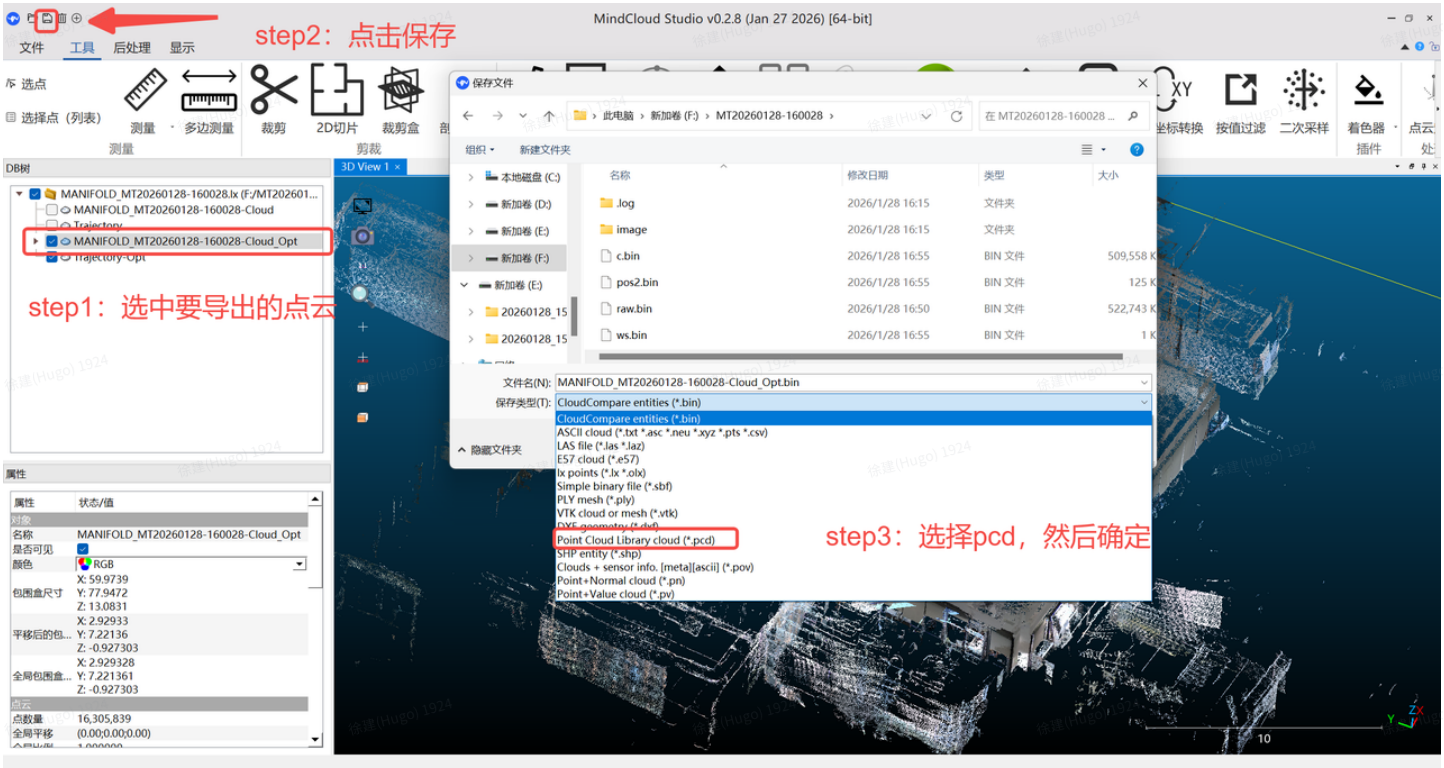
3. MindCloud使用

3.1 mindcloud处理 olx/lx数据

3.1.1 等待处理完成



3.1.2 保存Pcd点云图



3.1.3 保存bin地图

3.1.4 将生成的pcd点云地图和bin文件地图拷贝到ubuntu电脑上

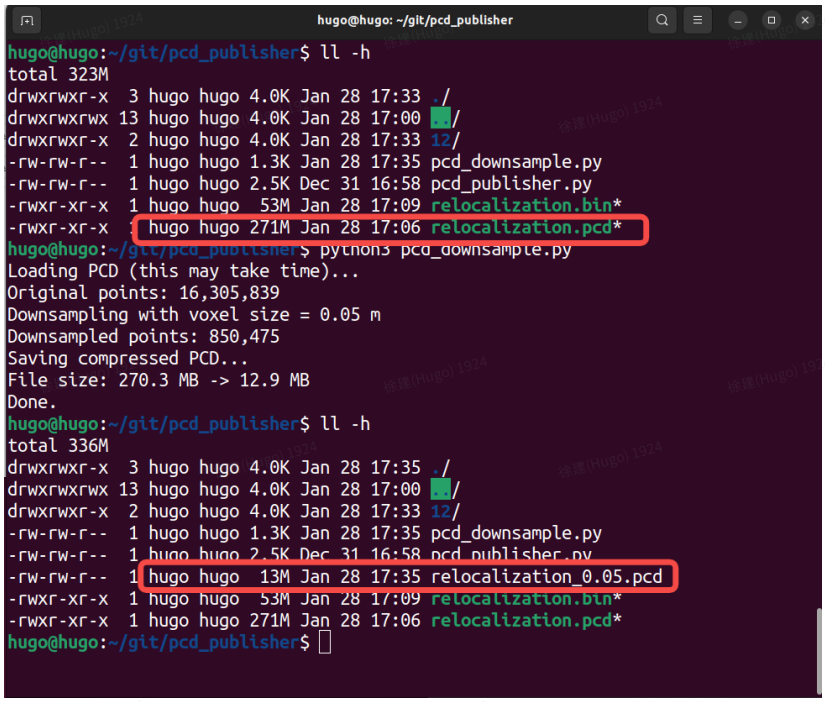
3.2 pcd点云抽稀

- 由于原始点云很大如果想在rviz中显示，需要进行点云抽稀
- 使用如下代码，建议将pcd文件数据量抽稀至20M以内（保证rviz流畅运行即可）

代码块

```python
#!/usr/bin/env python3
"""
超大 PCD 降采样工具（适合 RViz）
"""

###################### 可修改区域 ######################
INPUT_PCD  = "/home/hugo/git/pcd_publisher/relocalization.pcd.pcd"          # 原始 4.3G PCD
OUTPUT_PCD = "/home/hugo/git/pcd_publisher/relocalization_0.05.pcd"        # 输出压缩后
VOXEL_SIZE = 0.05                                  # 体素大小（单位：米）
########################################################

import open3d as o3d
import os

def main():
    print("Loading PCD (this may take time)...")
    pcd = o3d.io.read_point_cloud(INPUT_PCD)

    print(f"Original points: {len(pcd.points):,}")

    print(f"Downsampling with voxel size = {VOXEL_SIZE} m")
    pcd_ds = pcd.voxel_down_sample(voxel_size=VOXEL_SIZE)

    print(f"Downsampled points: {len(pcd_ds.points):,}")

    print("Saving compressed PCD...")
    o3d.io.write_point_cloud(
        OUTPUT_PCD,
        pcd_ds,
        write_ascii=False,    # binary
        compressed=True       # open3d 压缩
    )

    orig = os.path.getsize(INPUT_PCD) / 1024 / 1024
    out  = os.path.getsize(OUTPUT_PCD) / 1024 / 1024

    print(f"File size: {orig:.1f} MB -> {out:.1f} MB")
    print("Done.")

if __name__ == "__main__":
    main()

```

- 抽稀后



3.3 运行重定位同时发布pcd点云

3.3.1 设置重定位模式，输入正确的bin文件路径：

代码块

```
    custom_map_mode: 2      # 0: Odometry mode 1: SLAM mode 2: Relocalization mode
    custom_init_pos: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
    relocalization_map_abs_path: "/home/hugo/git/odin_0.8.0/map/relocalization.bin" # must be set for Relocalization mode or will fail
```

3.3.2 运行驱动

3.3.3 使用如下代码将抽稀后的pcd发布到ros中

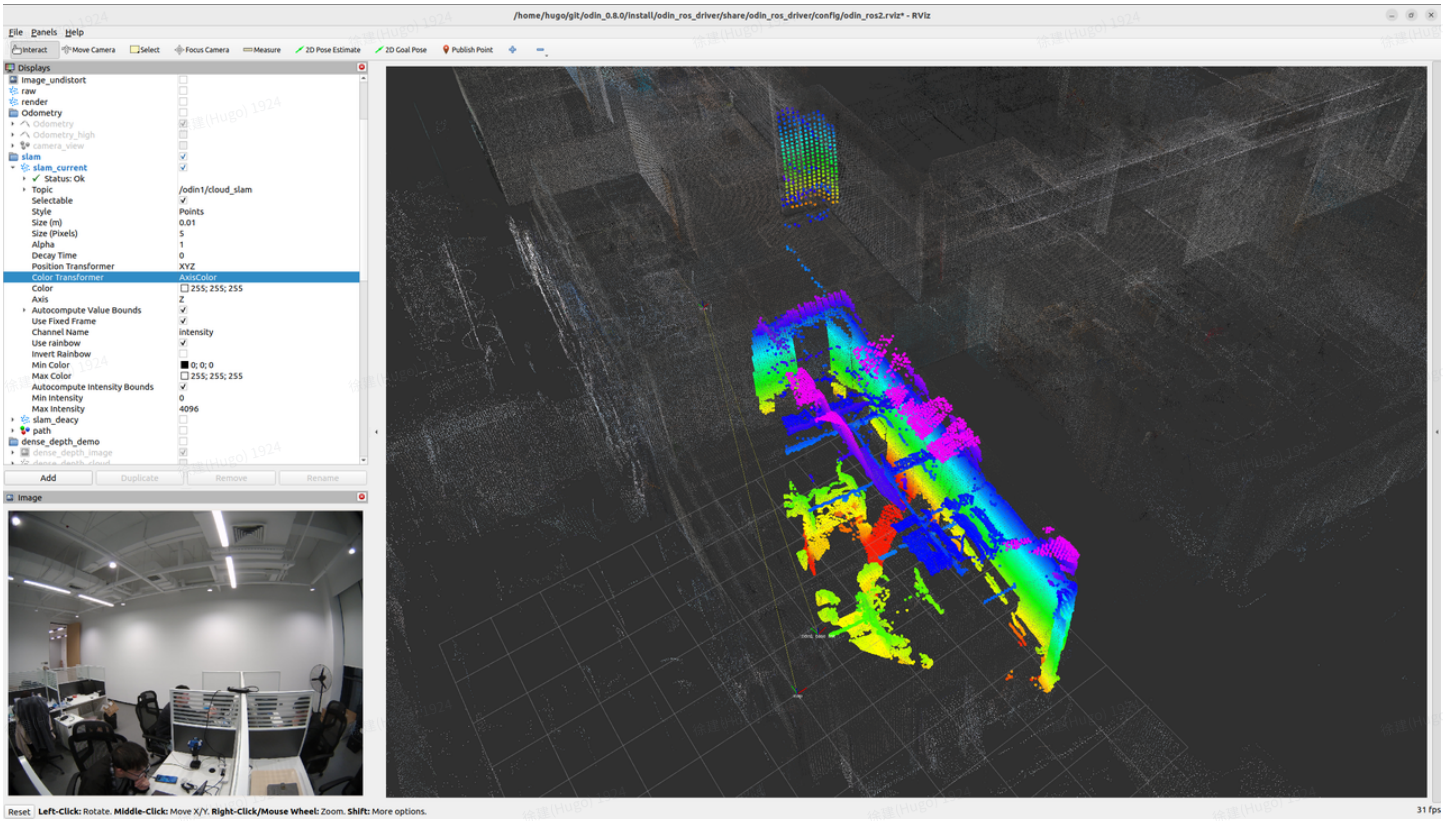
代码块

```

#!/usr/bin/env python3
"""
PCD (XYZRGB) -> ROS2 PointCloud2
使用 open3d，直接 python3 运行
"""

######################## 可修改区域 ########################
PCD_PATH  = "/home/hugo/git/pcd_publisher/relocalization_0.05.pcd"   # <<< 改这里
FRAME_ID = "map"
TOPIC    = "/pcd_points"
```

```
PUB_HZ   = 1.0
#########################################################

import rclpy
from rclpy.node import Node

import numpy as np
import open3d as o3d
import struct

from std_msgs.msg import Header
from sensor_msgs.msg import PointCloud2, PointField
from sensor_msgs_py import point_cloud2

def rgb_to_float(r, g, b):
    """RGB(uint8) -> packed float32"""
    rgb_uint32 = (int(r) << 16) | (int(g) << 8) | int(b)
    return struct.unpack("f", struct.pack("I", rgb_uint32))[0]

class PCDPublisher(Node):
    def __init__(self):
        super().__init__("pcd_rgb_publisher")

        self.pub = self.create_publisher(PointCloud2, TOPIC, 10)
        self.timer = self.create_timer(1.0 / PUB_HZ, self.timer_cb)

        self.cloud_msg = self.load_pcd(PCD_PATH)
        self.get_logger().info(f"Loaded PCD with RGB: {PCD_PATH}")

    def load_pcd(self, path: str) -> PointCloud2:
        pcd = o3d.io.read_point_cloud(path)

        points = np.asarray(pcd.points, dtype=np.float32)
        colors = np.asarray(pcd.colors, dtype=np.float32)  # [0,1]

        cloud_data = []
        for i in range(points.shape[0]):
            r = int(colors[i, 0] * 255)
            g = int(colors[i, 1] * 255)
            b = int(colors[i, 2] * 255)
            rgb_f = rgb_to_float(r, g, b)

            cloud_data.append((
                points[i, 0],
                points[i, 1],
                points[i, 2],
                rgb_f
```

```
            ))

        header = Header()
        header.frame_id = FRAME_ID

        fields = [
            PointField(name="x",   offset=0,  datatype=PointField.FLOAT32,
count=1),
            PointField(name="y",   offset=4,  datatype=PointField.FLOAT32,
count=1),
            PointField(name="z",   offset=8,  datatype=PointField.FLOAT32,
count=1),
            PointField(name="rgb", offset=12, datatype=PointField.FLOAT32,
count=1),
]

        return point_cloud2.create_cloud(header, fields, cloud_data)

    def timer_cb(self):
        self.cloud_msg.header.stamp = self.get_clock().now().to_msg()
        self.pub.publish(self.cloud_msg)

def main():
    rclpy.init()
    node = PCDPublisher()
    rclpy.spin(node)
    node.destroy_node()
    rclpy.shutdown()

if __name__ == "__main__":
    main()
```

3.3.4 效果如下：



4. 使用Odin录制重定位地图（选择3）

4.1 录制bin格式地图

- 设置control_command.yaml文件中customer_map_mode: 1，recorddata: 0

代码块

```
recorddata: 0           # 0: off; 1: on
***
custom_map_mode: 1      # 0: Odometry mode 1: SLAM mode 2: Relocalization mode
```

- 运行驱动进行数据采集

代码块

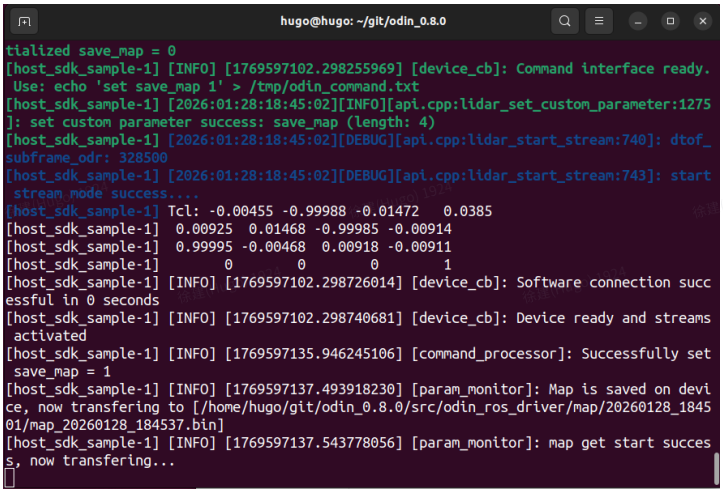
```
cd ~/catkin_ws
# 以下以ros2为例，ros1请输入对应的指令
source install/setup.bash
ros2 launch odin_ros_driver odin1_ros2.launch.py
```

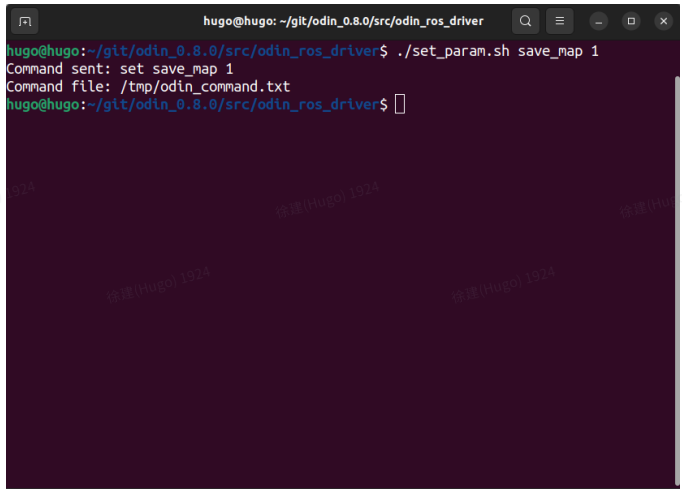
- 采集地图结束后请勿直接Ctrl+c关闭驱动，需要新建终端，进入到驱动路径中，运行./set_param.sh save_map 1保存地图。

代码块

```
cd ~/catkin_ws/src/odin_ros_driver
```

```
./set_param.sh save_map 1
# 等待驱动终端提示保存完成后再结束驱动
```





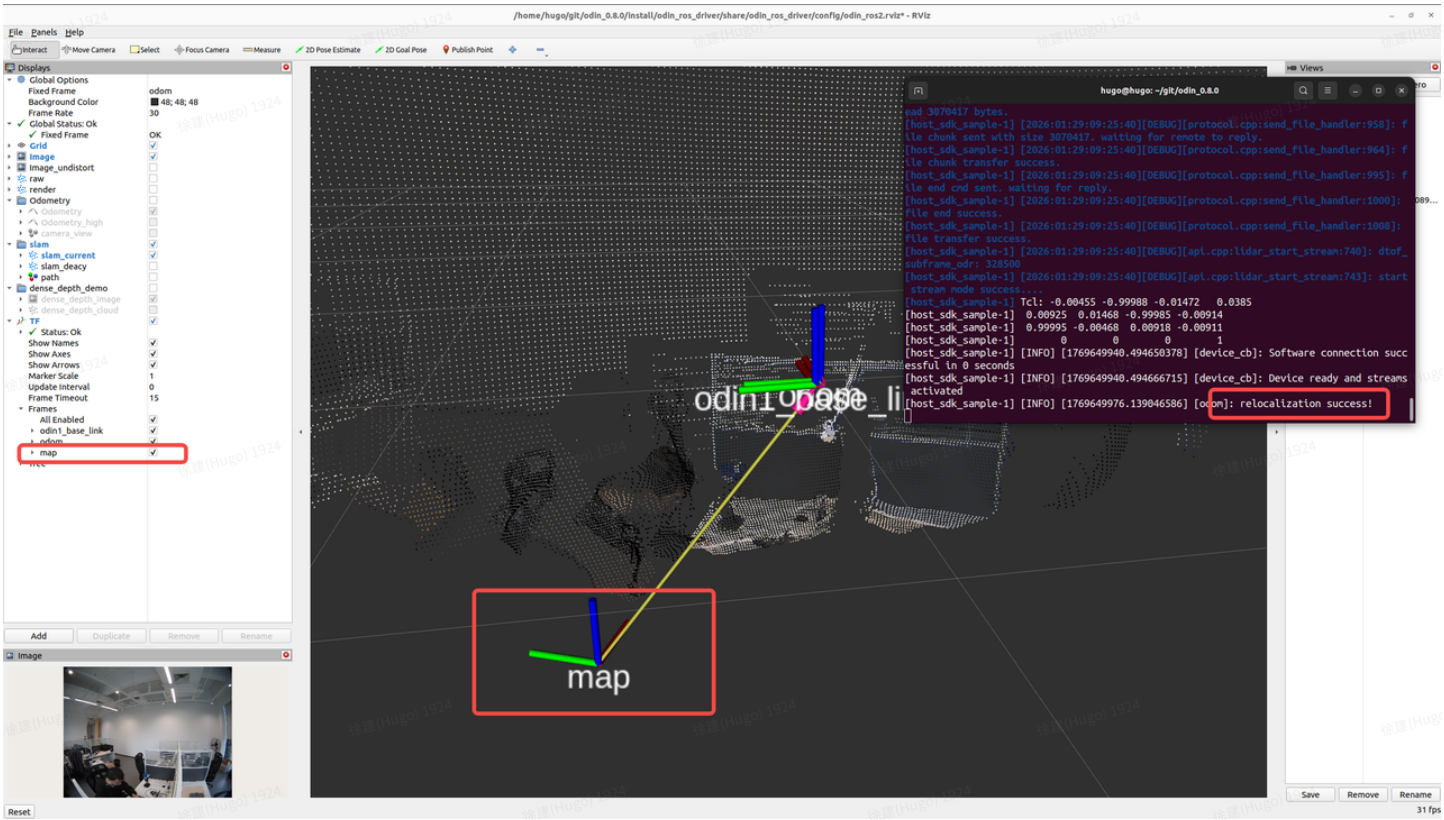
4.2 直接运行重定位

- 设置control_command.yaml文件中customer_map_mode: 2，配置bin文件所在路径：

代码块

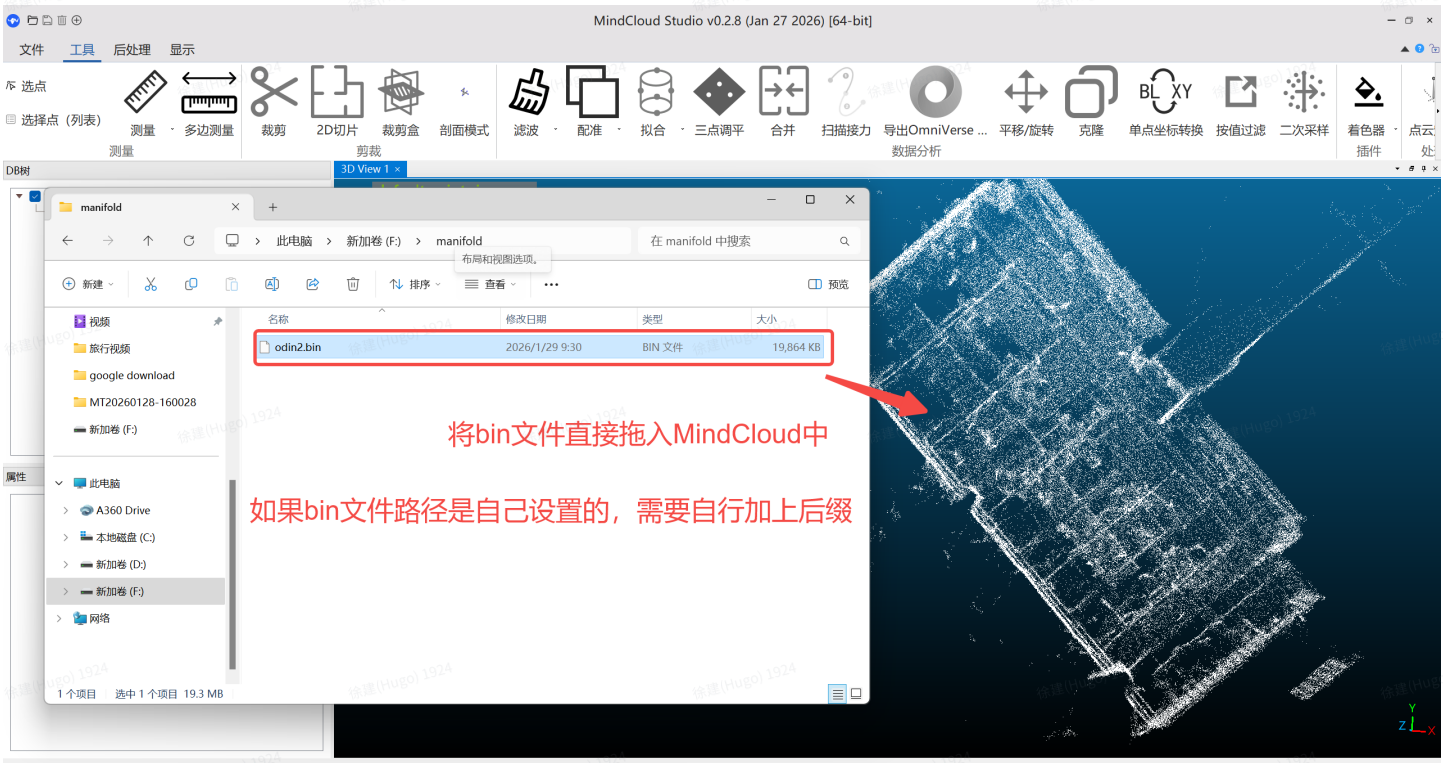
```
custom_map_mode: 2       # 0: Odometry mode 1: SLAM mode 2: Relocalization mode
custom_init_pos: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
relocalization_map_abs_path:
"/home/hugo/git/odin_0.8.0/src/odin_ros_driver/map/20260128_184501/map_20260128_184537.bin" # must be set for Relocalization mode or will fail
```

- 运行驱动
- 定位成功后显示如下：rviz显示map系，驱动终端提示重定位成功!

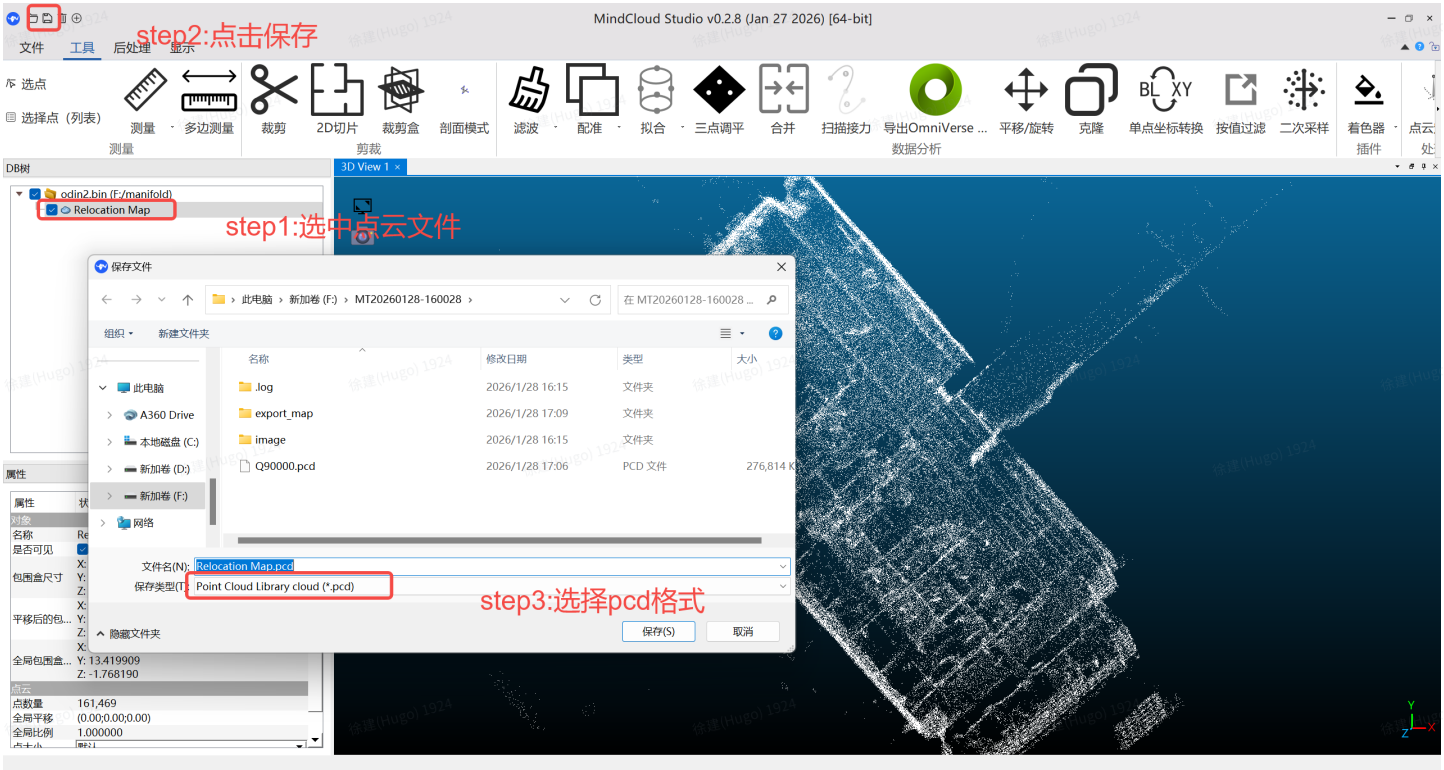


4.3 bin文件导出pcd

- 拷贝Odin导出的bin文件到安装Mindcloud0.2.8版本的PC上
- 将bin文件直接拖入MindCloud中



- 导出pcd格式点云



4.4 rviz显示pcd点云

- 将pcd拷贝到ubuntu（驱动所在）电脑上
- 同1.5一样进行加载但是要注意bin导出的pcd没有rgb信息，也可以依据自己需求书写发布代码

注：上述提及代码均为AI生成的demo，暂不技术支持。