

- 1. Steps to reproduce the results
  - 1.1. Need docker?
    - 1.1.1. Download the docker image by
    - 1.1.2. Run the docker
  - 1.2. ROS
    - 1.2.1. coloradar
    - 1.2.2. reproduce the groundtruth
    - 1.2.3. demo
    - 1.2.4. rqt\_graph
- 2. Code interpretation
  - 2.1. what is frame
- 3. inherit from vins-mono
  - 3.1. reducevector
  - 3.2.
- 4. Sensor Characteristic/Front end
  - 4.1. Radar
    - 4.1.1. /radar\_frame
  - 4.2. what is row\_step
  - 4.3.
  - 4.4. IMU
- 5. Back end/Optimization
- 6. Output Data Extraction
  - 6.1. map
    - 6.1.1. /sub\_map\_frame
    - 6.1.2. /tracking\_frame
      - 6.1.2.1. Is /tracking\_frame the overlap of the current frame with sub\_map?
      - 6.1.2.2. Is radarFeatureFactor.pointRelation contains each point from the current frame, where the point without match turns out to be zeros-match?
  - 6.2. Pose
    - 6.2.1. /estimated\_pose (world to inertial frame)
    - 6.2.2. rio
- 7. RIO frame management
- 8. Issue
  - 8.1. ERROR: Unable to start XML-RPC server, port 11311 is already in use (just occasionally occurs)
  - 8.2. z error is very large
  - 8.3. Bug
    - 8.3.1. the rio msg is strange, always being jumping
    - 8.3.2. the header timestamp of radar\_frame is wrong
    - 8.3.3. the rviz is dark
- 9. Xu Yang questions

# 1. Steps to reproduce the results

---

## 1.1. Need docker?

### 1.1.1. Download the docker image by

```
bash docker.sh -b
```

(then, it start download the file and build

```
function build() {
    docker build \
    -t rio \
    -f $SCRIPT_DIR/Dockerfile \
    $SCRIPT_DIR/. .
}
```

```

)
ubuntu@ubuntu-Legion-Y9000P-IRX9:~/assignments/RIO/src/RIO/docker$ bash docker.sh -b
[+] Building 902.4s (9/12)
aule
==> [internal] load build definition from Dockerfile
0.1s
==> == transferring dockerfile: 4.69kB
0.0s
==> [internal] load metadata for docker.io/library/ubuntu:20.04
1.8s
==> [internal] load .dockerignore
0.1s
==> == transferring context: 2B
0.0s
==> [1/9] FROM docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b
1.2s
==> == resolve docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b
0.0s
==> == sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b 6.69kB / 6.69kB
0.0s
==> == sha256:e5a0aeeef391a8a9bdaee3deeb28f393837c479d8217324a2340b64e45a81e0ef 424B / 424B
0.0s
==> == sha256:6013ae1a63c2ee58a8949f03c366a3ef6a2f386a7db27d86de2de965e9f450b 2.36kB / 2.36kB
0.0s
==> == sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f110e11cd99592daf261e9cd2 27.51MB / 27.51MB
1.7s
==> == extracting sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f110e11cd99592daf261e9cd2
[+] Building 1048.0s (13/13) FINISHED
==> [internal] load build definition from Dockerfile
==> == transferring dockerfile: 4.69kB
==> [internal] load metadata for docker.io/library/ubuntu:20.04
==> [internal] load .dockerignore
==> == transferring context: 2B
==> [1/9] FROM docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b
==> == resolve docker.io/library/ubuntu:20.04@sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b
==> == sha256:8e5c4f0285ecbb4ead70431d29b576a530d316ddf73ec44affc1cd27555141b 6.69kB / 6.69kB
==> == sha256:e5a0aeeef391a8a9bdaee3deeb28f393837c479d8217324a2340b64e45a81e0ef 424B / 424B
==> == sha256:6013ae1a63c2ee58a8949f03c366a3ef6a2f386a7db27d86de2de965e9f450b 2.36kB / 2.36kB
==> == sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f110e11cd99592daf261e9cd2 27.51MB / 27.51MB
==> == extracting sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f110e11cd99592daf261e9cd2
==> [2/9] RUN apt-get update && apt-get install -y build-essential libgtk2.0-dev lsb-release net-tools cmake libgoogle-glog-dev libgflags-dev libatlas-base-dev libeigen3-d
==> [3/9] RUN sh -c 'echo "deb http://mirrors.ustc.edu.cn/ros/ubuntu/ lsb-release -cs" main" > /etc/apt/sources.list.d/ros-latest.list' && curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.
==> [4/9] RUN apt-get update && apt-get install -y ros-noetic-navros ros-noetic-navros-extras ros-noetic-navlink && apt-get clean && wget https://raw.githubusercontent
==> [5/9] RUN apt-get update && apt-get install -y python3-pip && apt-get clean && python3 -m pip install --upgrade --no-binary :egg: && python3 -m pip install packaging
==> [6/9] RUN apt-get update && apt-get install -y ca-certificates gpg wget software-properties-common && wget -O - https://apt.kitware.com/keys/kitware-archive-latest.asc 2>/dev/null | g
==> [7/9] RUN mkdir -p /root/.3rdparty && cd /root/.3rdparty && wget -O opencv.zip https://github.com/Itseez/opencv/archive/4.6.0.zip && wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/
==> [8/9] RUN cd /root/.3rdparty && wget http://ceres-solver.org/ceres-solver-2.2.0.tar.gz && tar xzf ceres-solver-2.2.0.tar.gz && mkdir ceres-bin && cd ceres-bin && cmake ../ceres-solver-2.2.
==> [9/9] RUN cd /root/.3rdparty && git clone https://github.com/bombela/backward-cpp.git && cd backward-cpp && mkdir -p build && cd build && cmake .. && make -j$(nproc) && make instal
==> == exporting layers
==> == writing image sha256:30b93e04e9f3afe20b388a97c02c49679a25bfbd0f2159404644db1088b22
==> == naming to docker.io/library/rio
What's Next?
1. Sign in to your Docker account -> docker login
2. View a summary of image vulnerabilities and recommendations -> docker scout quickview

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
rio	latest	38b93e04e9f3	13 minutes ago	5.8GB
registry.cn-shanghai.aliyuncs.com/shenlanxueyuan/sensor-fusion-workspace	bionic-cpu-vnc-20240621	1542fa39af03	6 months ago	10.9GB
registry.cn-shanghai.aliyuncs.com/shenlanxueyuan/sensor-fusion-workspace	bionic-cpu-vnc-20240609	eb035d4c3212	6 months ago	8.23GB
registry.cn-shanghai.aliyuncs.com/shenlanxueyuan/sensor-fusion-workspace	bionic-cpu-vnc-20240608	9b1f7f8945db	6 months ago	7.63GB
hello-world	latest	d2c94e258dcb	20 months ago	13.3kB
registry.cn-shanghai.aliyuncs.com/shenlanxueyuan/sensor-fusion-workspace	bionic-cpu-vnc	a60e774709f3	3 years ago	6.21GB

### 1.1.2. Run the docker

```
bash docker.sh -r
```

*This starts the Docker container and attaches your terminal to it. Once the container starts successfully, the terminal will show inside the Docker container.*

(mapping the current folder to docker

```
function run() {
    docker run -it --rm \
    --network host \
    --privileged \
    -v /dev:/dev \
    -v $SCRIPT_DIR/../../ws/src \
    -e DISPLAY=$DISPLAY \
    -v /tmp/.X11-unix:/tmp/.X11-unix \
    rio \
    /bin/bash
}
```

)

## 1.2. ROS

bash 1

```
roscore &
rviz -d rio/config/RI0.rviz
```

bash 2

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/ars548.yaml -
d /ws/src/dataset/exp/Sequence_3.bag -r 1 -p 1
```

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/ars548.yaml -
d /ws/src/dataset/exp/Sequence_3_modified_gt_horizontal.bag -r 1 -p 1
```

### 1.2.1. coloradar

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/coloradar.yaml -d
dataset/coloradar_trim/colo_trim_outdoors_run_0_modified_gt2.bag -r 1 -p 1
```

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/coloradar.yaml -d
dataset/coloradar_trim/colo_trim_outdoors_run_0_modified_gt2.bag -r 1 -p 1
```

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/coloradar.yaml -d
dataset/coloradar_trim/classroom3.bag -r 1 -p 1
```

```
python3 /ws/src/docker/run.py -a -n rio -c /ws/src/rio/config/coloradar.yaml -d
dataset/coloradar_trim/classroom3_modified_gt_horizontal.bag -r 1 -p 1
```

```
evo_ape bag ./rio_output_seq3_2025-02-09-13-43-21_0.2toend.bag /estimated_pose /lidar_ground_truth -
va --plot_mode xy --plot -t_max_diff 0.05
```

The screenshot shows a ROS2 environment with a terminal window and a RViz visualization.

**Terminal Output:**

```

roscore http://ubuntu-Legion-Y9000P-8X9J:11311/
root@ubuntu-Legion-Y9000P-8X9J: ~/src
./src/rio/launch/TestRadar.launch http://localhost:11311

Done checking log file disk usage. Usage is 1GB.
started rosLaunch server http://ubuntu-Legion-Y9000P-8X9J:37953/

SUMMARY
=====

PARAMETERS
* /rio/SignalAzmuth: 0.00290888
* /rio/SignalElevation: 0.00458178
* /rio/SignalRange: 0.05
* /rio/dopplerResidualWeight: 1
* /rio/frameSubTopic: /raw
* /rio/ImuSubTopic: /navir
* /rio/mavlinkInfoGain: 2000
* /rio/mavlinkInfoGain: 1500
* /rio/nav_gstSubTopic: /p
* /rio/navStatus: 3
* /rio/observationThreshold: 1
* /rio/poseSubTopic: /est
* /rio/poseSubTopic: /radar_frame
* /rio/radarType: 0
* /rio/radarSubTopic: /raw
* /rio/trackingPubTopic: /tracking_frame
* /rio/useDopplerResidual: 1
* /rio/useImuInfoGain: 1
* /rio/useLightedResidual: 1
* /rio/rostdrv: meetic
* /rosversion: 1.17.0

NODES
  /relay (topic_tools/relay)
  /rio (rio)
  /rosbag_play (rosbag/play)

ROS_MASTER_URI=http://localhost:11311/
process[rio-1]: started with pid 1000
process[rosbag_play-2]: started with pid 1001
process[relay-3]: started with pid 1002
useLightedResiduals: 1
useSfFilter: 1
observationThreshold: 4
useDopplerResidual: 1
useImuInfoGain: 1
SignalRange: 0.05
SignalAzmuth: 0.00290888
SignalElevation: 0.00458178
SignalDoppler: 0
mavlinkInfoGain: 2000
mavlinkInfoGain: 1500
dopplerResidualWeight: 1500
pointResidualWeight: 1000
radarSubTopic: /radar_frame
ImuSubTopic: /navir
geometry_gstSubTopic: /p
poseSubTopic: /estimated_pose
pathSubTopic: /estimated_path
framePubTopic: /radar_frame
trackingPubTopic: /tracking_frame
subMapPubTopic: /sub_map_frame
radarType: 0
0:
-0.173502
0.326126
-9.79858

```

**RViz Visualization:**

The RViz window displays a 3D point cloud of a scene. The 'Global Options' panel on the left shows the following settings:

- Fixed Frame: world
- Background Color: 48, 48, 48
- Frame Rate: 30
- Default Light: On
- Global Status: Warn
- Grid: On
- Odometry: On
- Status: OK
- Topic: /pos\_vel\_mocap/odom...
- Unpublishable: Off
- Queue Size: 10
- Position Tolerance: 0.1
- Angle Tolerance: 0.1
- Keep: 1
- Slope: Auto
- Covariance: On
- PointCloud2: On
- Status: OK
- Topic: /tracking\_frame
- Unpublishable: Off
- Queue Size: 10
- Selectable: On
- Style: Flat Squares
- Size (m): 0.1
- Line: 1
- Decay Time: 0
- Position Transfer: XYZ
- Color Transformer: FlatColor
- Color: 255, 0, 0
- PointCloud2: On
- Pass: On
- Path: On

The 'Displays' panel in the center shows the following displays:

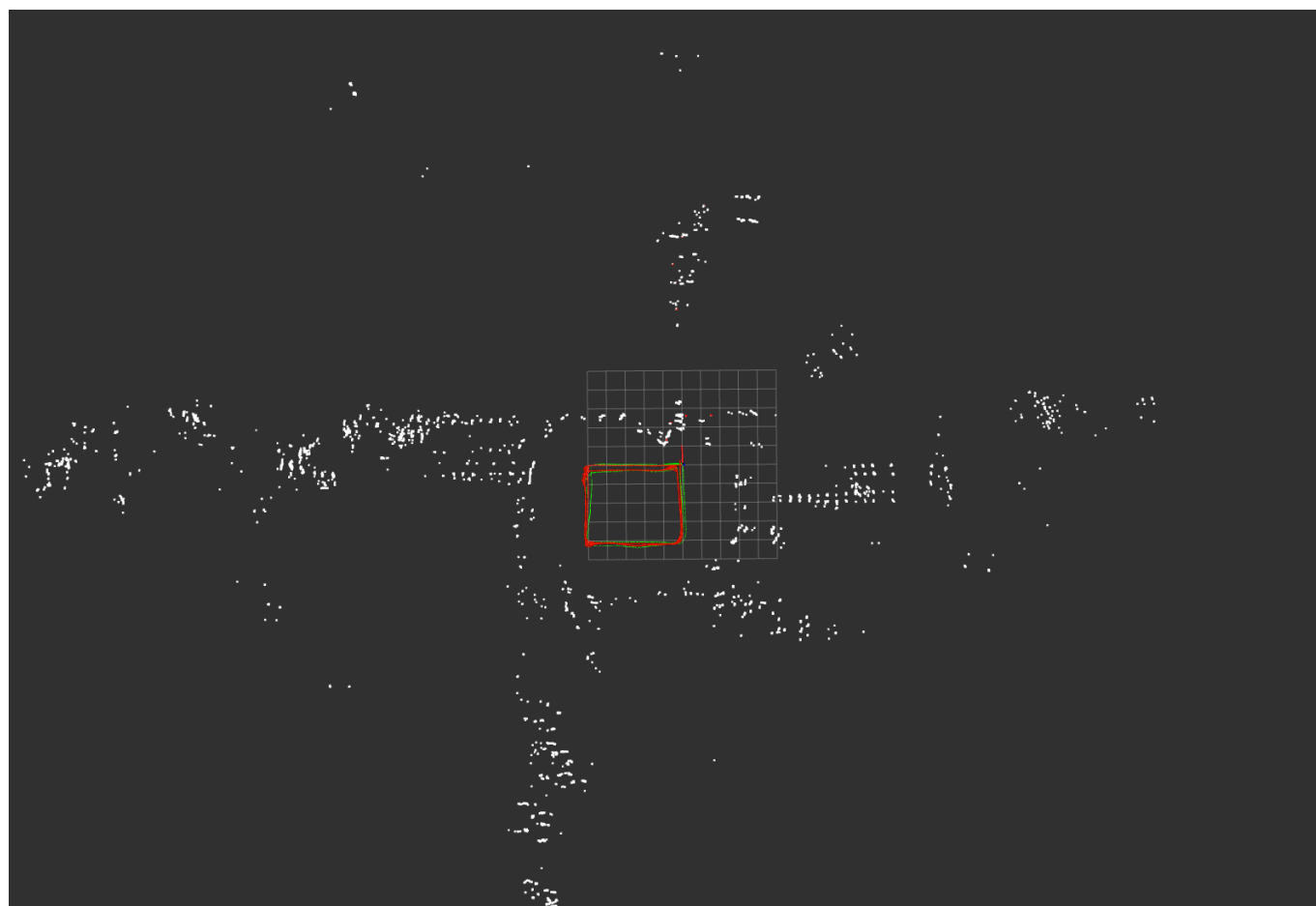
- Global Options
- Fixed Frame
- Odometry
- Status
- Topic
- Unpublishable
- Queue Size
- Selectable
- Style
- Size (m)
- Line
- Decay Time
- Position Transfer
- Color Transformer
- Color
- PointCloud2
- Pass
- Path

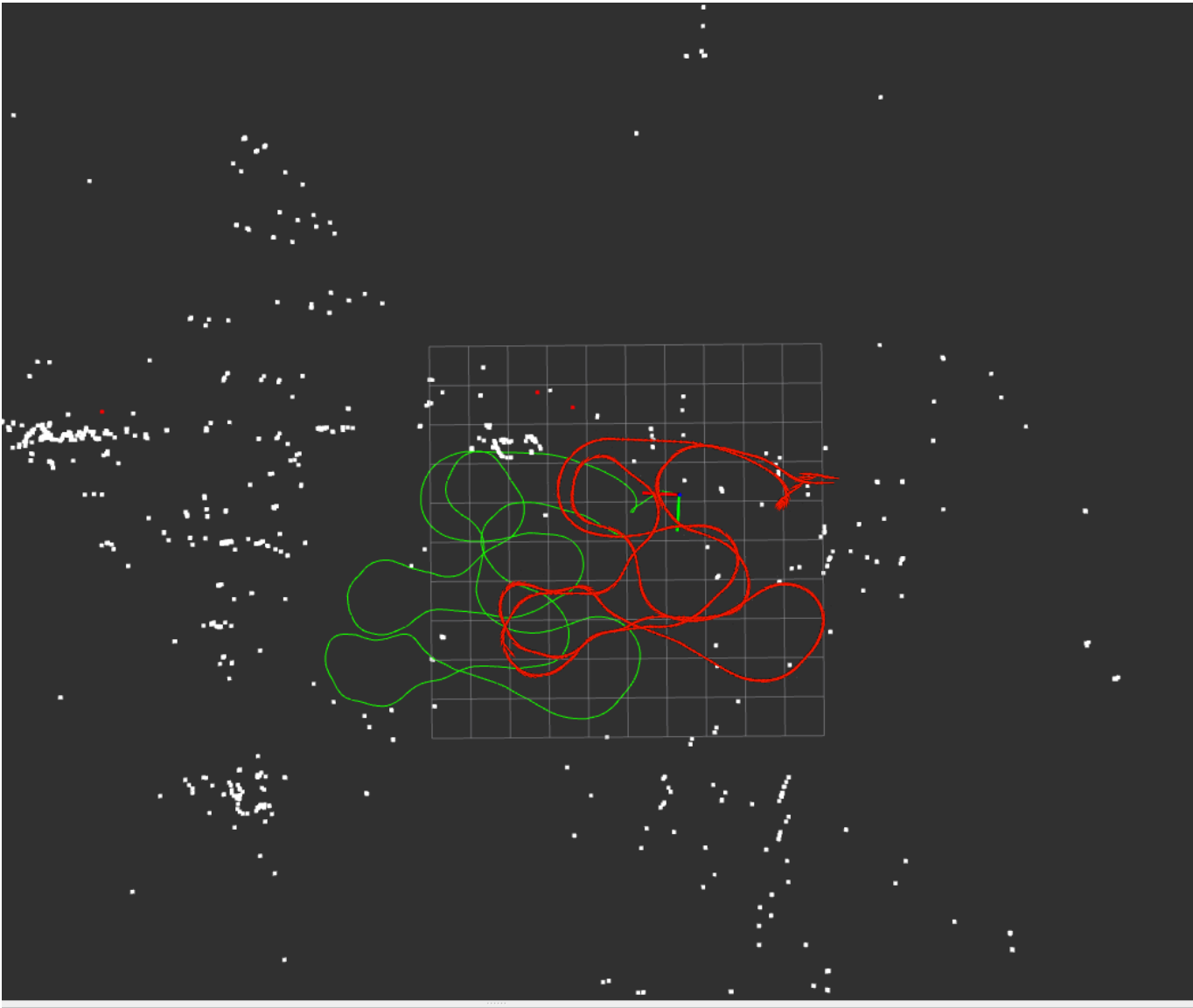
The 'Views' panel on the right shows the following views:

- Current View: Orbit (rviz)
- Orbit (rviz)
- Invert Z Axis: Off
- Fixed Frame: world
- Distance: 36.3704
- Focal Shape: Off
- Yaw: 2.1382
- Pitch: 1.4298
- Field of View: 0.765398
- Focal Point: 0.63778, -1.4206, ...

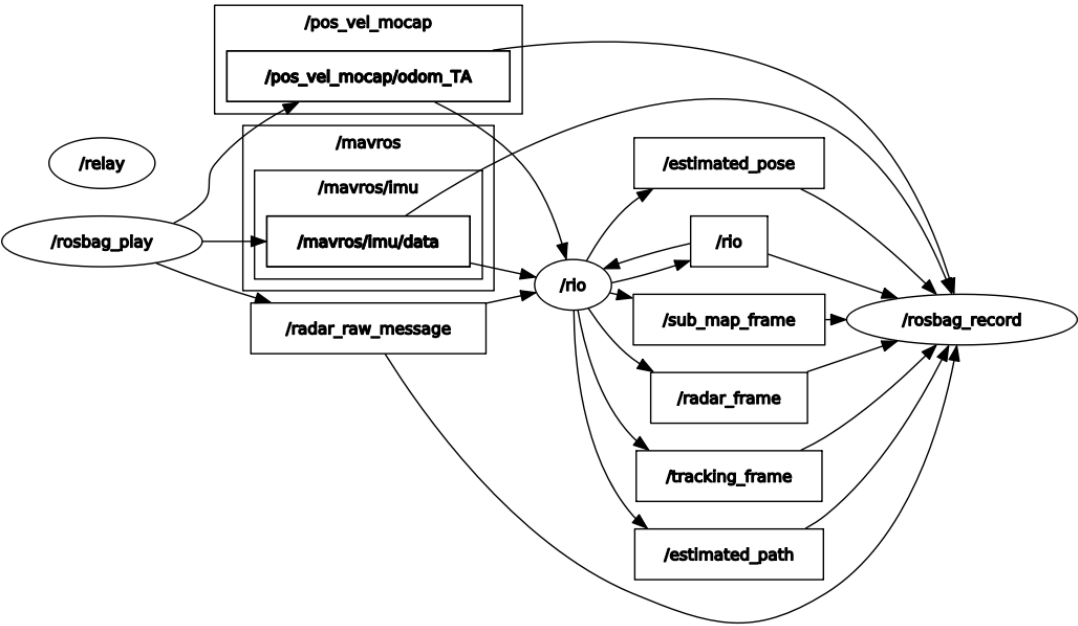
The bottom status bar shows the following information:

- Time: 1735210740.55
- ROS Time: 1735210740.55
- ROS Elapsed: 43.81
- Wall Time: 1735210740.58
- Wall Elapsed: 43.74
- 31 fps





1.2.4. rqt\_graph



## 2. Code interpretation

### 2.1. what is frame

```
frame = scan2scanTracker.trackPoints(frameRadarData, timeStamp);

radarData.data.emplace_back(frame);
```

## 3. inherit from vins-mono

### 3.1. reducevector

```
template <typename Derived>
static void reduceVector(vector<Derived> &v, vect
{
    int j = 0;
    for (int i = 0; i < int(v.size()); i++)
        if (status[i])
            v[j++] = v[i];
    v.resize(j);
}

template <typename T>
inline void reduceVector(std::vector<T> &vector, std::vector<uchar> status) {
    int size = 0;
    for (int i = 0; i < int(vector.size()); i++)
        if (status[i]) vector[size++] = vector[i];
    vector.resize(size);
}
```

### 3.2.

## 4. Sensor Charactoristic/Front end

### 4.1. Radar

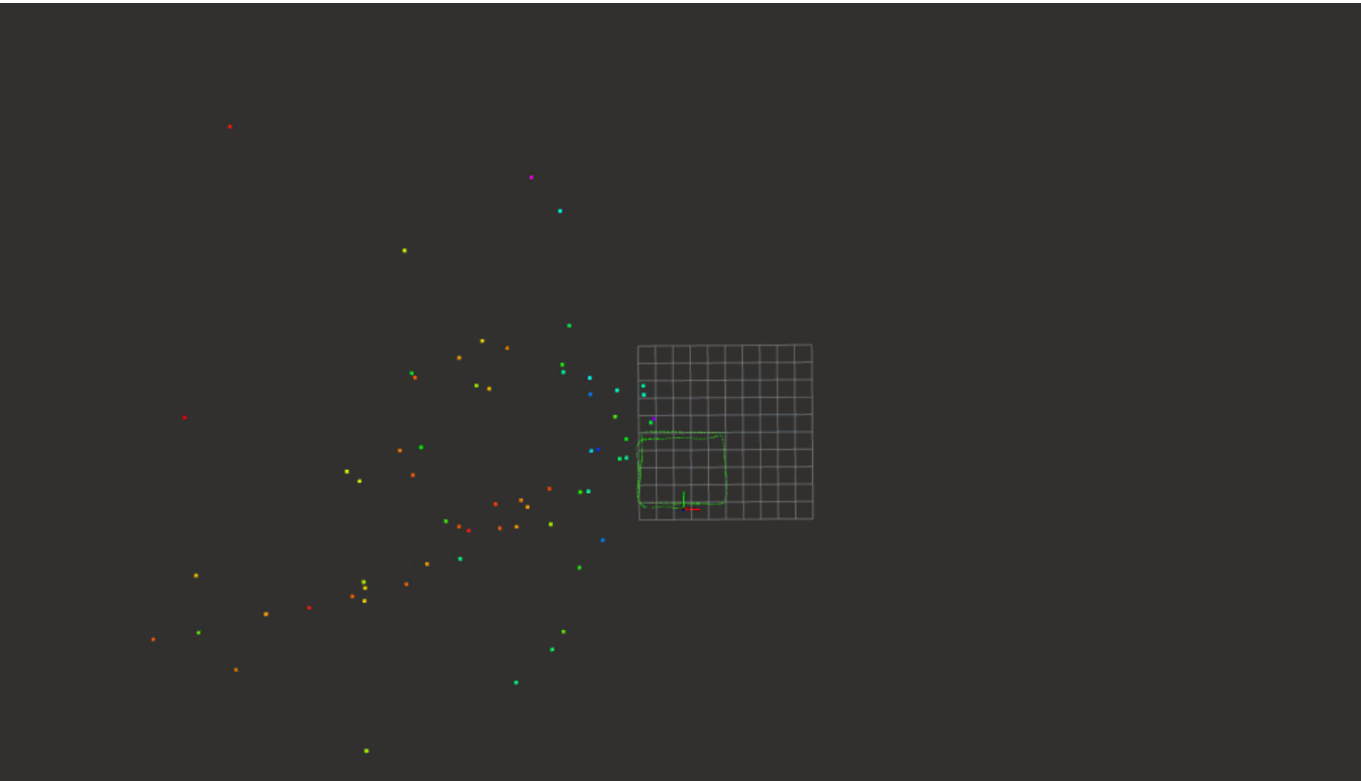
#### 2. Structure of Binary Data:

The binary data likely looks like this (for a single point):

Field Name	Size (Bytes)	Type	Offset
azimuth	4	float	offsetAzimuth
azimuthSTD	4	float	offsetAzimuthSTD
elevation	4	float	offsetElevation
elevationSTD	4	float	offsetElevationSTD
range	4	float	offsetRange
rangeSTD	4	float	offsetRangeSTD
velocity	4	float	offsetVelocity
velocitySTD	4	float	offsetVelocitySTD
rsc	1	int8_t	offsetRCS

The offsets for each field correspond to the `offset` specified in the `fields` metadata. For instance:

4.1.1. /radar\_frame



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
time	field_header_seq	field_header_stamp	field_header_frame_id	field_height	field_width	field_fields0_name	field_fields0_offset	field_fields0_datatype	field_fields0_count	field_fields1_name	field_fields1_offset	field_fields1_datatype	field_fields1_count	field_fields2_name	field_fields2_offset	field_fields2_datatype	field_fields2_count
1.73522568279073E+018	18	0	world	1	84x		0	7	1y		4	7	1z		8	7	1w
1.73522568280311E+018	19	0	world	1	84x		0	7	1y		4	7	1z		8	7	1w
1.73522568289412E+018	20	0	world	1	79x		0	7	1y		4	7	1z		8	7	1w
1.73522568293601E+018	21	0	world	1	82x		0	7	1y		4	7	1z		8	7	1w
1.73522568296275E+018	22	0	world	1	75x		0	7	1y		4	7	1z		8	7	1w
1.73522568300674E+018	23	0	world	1	74x		0	7	1y		4	7	1z		8	7	1w
1.73522568309019E+018	24	0	world	1	85x		0	7	1y		4	7	1z		8	7	1w
1.73522568313606E+018	25	0	world	1	84x		0	7	1y		4	7	1z		8	7	1w
1.73522568319605E+018	26	0	world	1	80x		0	7	1y		4	7	1z		8	7	1w
1.73522568323456E+018	27	0	world	1	86x		0	7	1y		4	7	1z		8	7	1w
1.73522568326712E+018	28	0	world	1	87x		0	7	1y		4	7	1z		8	7	1w
1.73522568332939E+018	29	0	world	1	86x		0	7	1y		4	7	1z		8	7	1w
1.7352256833835E+018	30	0	world	1	72x		0	7	1y		4	7	1z		8	7	1w
1.73522568342713E+018	31	0	world	1	86x		0	7	1y		4	7	1z		8	7	1w
1.73522568347969E+018	32	0	world	1	84x		0	7	1y		4	7	1z		8	7	1w
1.73522568352313E+018	33	0	world	1	79x		0	7	1y		4	7	1z		8	7	1w
1.73522568360111E+018	34	0	world	1	81x		0	7	1y		4	7	1z		8	7	1w

angle seems to be [-60, 60]x[-20, 20]

4.2. what is row\_step

it is how many points there are in a single frame.

4.3.

4.4. IMU

5. Back end/Optimization

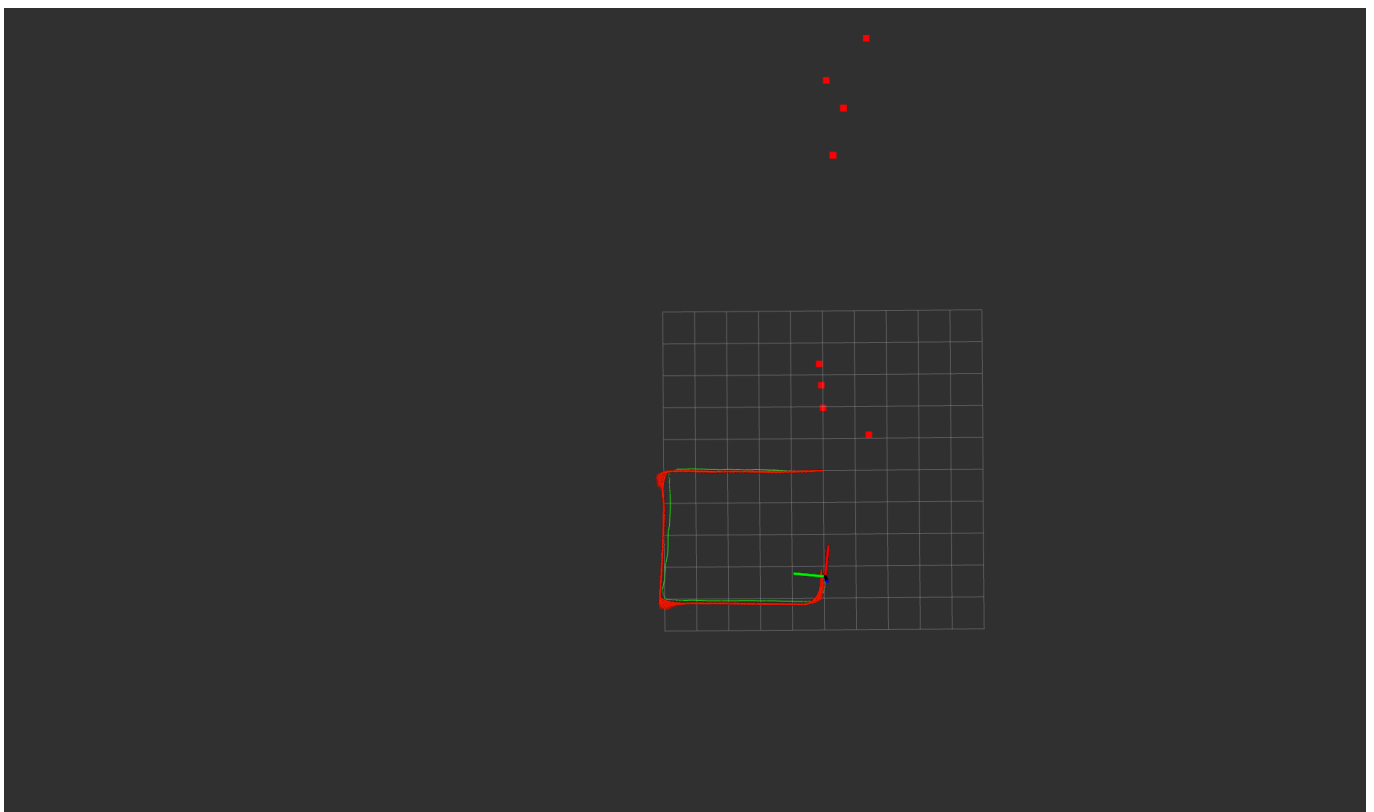
6. Output Data Extraction

6.1. map

6.1.1. /sub\_map\_frame



### 6.1.2. /tracking\_frame



#### 6.1.2.1. Is /tracking\_frame the overlap of the current frame with sub\_map?

Yes

#### 6.1.2.2. Is radarFeatureFactor.pointRelation contains each point from the current frame, where the point without match turns out to be zeros-match?



Likely

## 6.2. Pose

6.2.1. /estimated\_pose (world to inertial frame)

6.2.2. rio

seems to be groundtruth, the same name as node name

## 7. RIO frame management

---

all of the states uses inertial frames as body frames

## 8. Issue

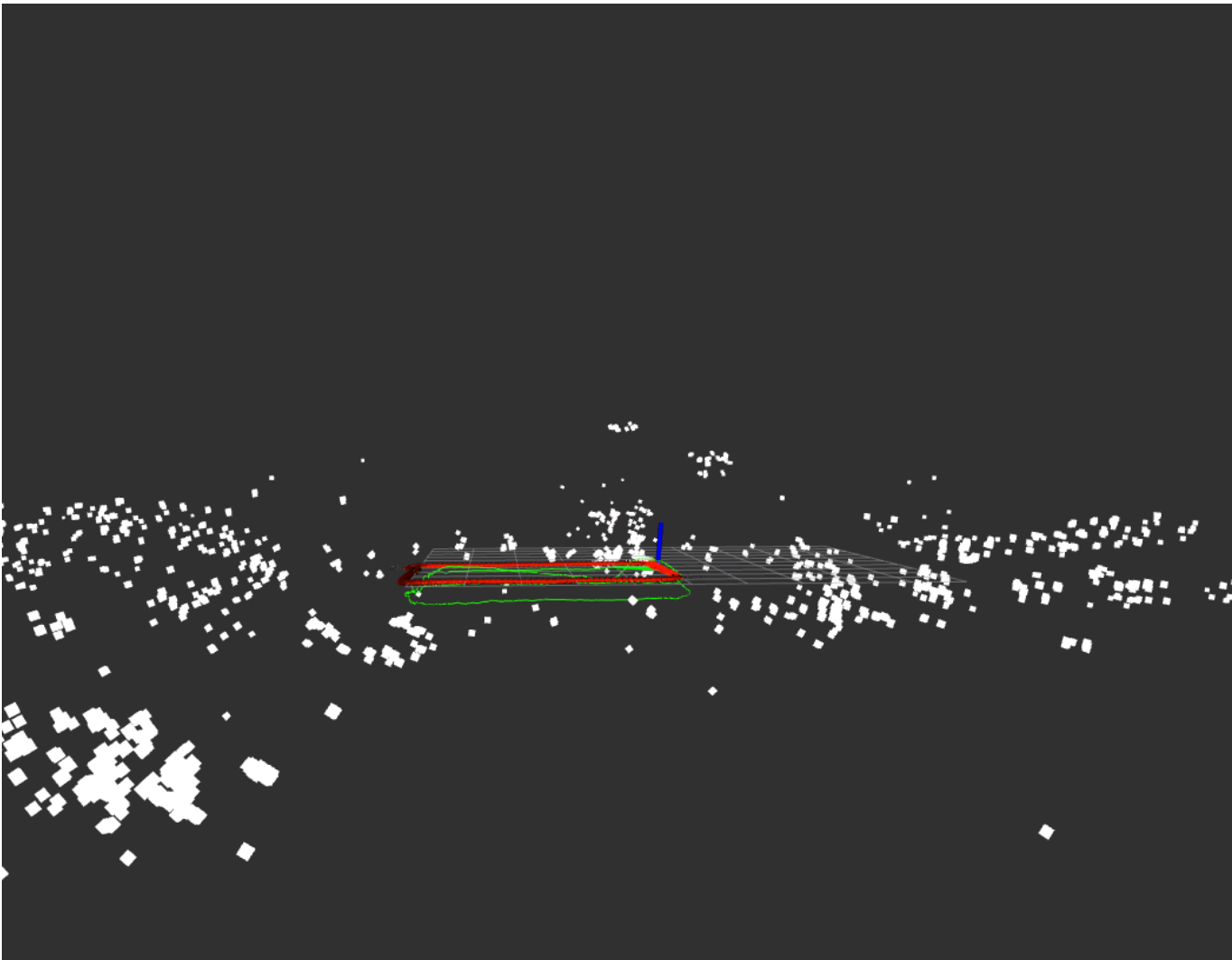
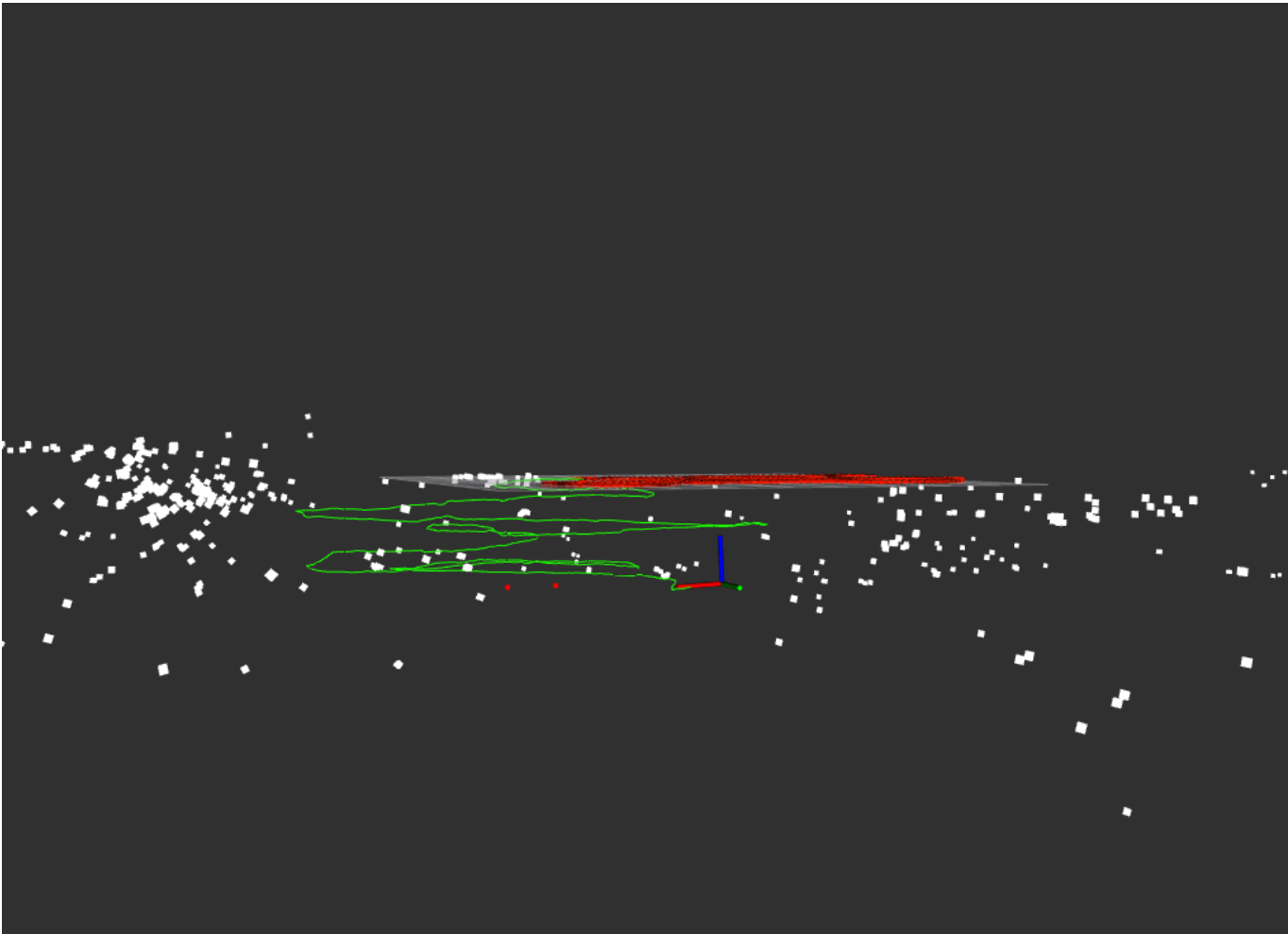
---

8.1. ERROR: Unable to start XML-RPC server, port 11311 is already in use (just occasionally occurs)

```
roscore -p 11312
```

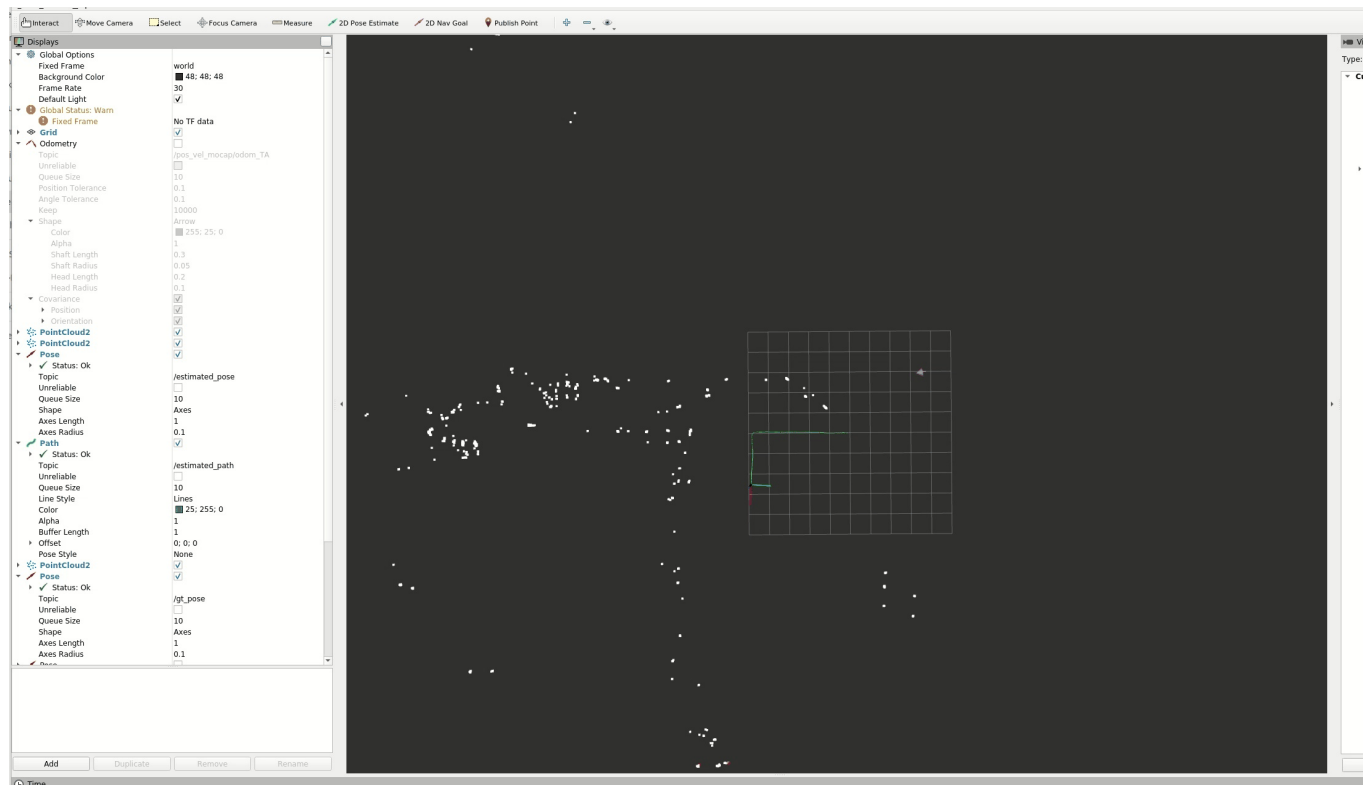
```
export ROS_MASTER_URI=http://localhost:11312
```

8.2. z error is very large



## 8.3. Bug

### 8.3.1. the rio msg is strange, always being jumping



### 8.3.2. the header timestamp of radar\_frame is wrong

now it is zero, not very good. also the frame is world, which is not appropriate for visualization.

### 8.3.3. the rviz is dark

we can adjust window size in the rviz file, makeing it smaller.

sometimes retry many times can help alleviate the problem.

other times waiting and retrying are okay.

## 9. Xu Yang questions