

Radar传感器相关问题总结

目录

- [RT6相关问题](#)
- [角雷达测速不准](#)
 - 问题背景
 - 具体表现
 - 总结
- [窄路双萝卜互相干扰底噪抬高](#)
 - 问题背景
 - 具体表现
 - 总结
- [RT5相关问题](#)

1. RT6相关问题

1.1. 角雷达测速不准

1.1.1. 问题背景

主要问题：当主车速度在20km/h以上时，角雷达（conti523）对侧面静止障碍物测速不准。

模式：NR模式（12米内密集点云）

Radar点云预处理核心代码在 [文件页：baidu/adu-perception/sensor *master](#)，主要是将雷达原始的局部极坐标点云转换到世界坐标系，并计算每个点的世界系速度，期间通过tf信息计算车速到每个雷达的线速度与角速度，叠加得到绝对径向速度。

Mermaid

```
</>
```

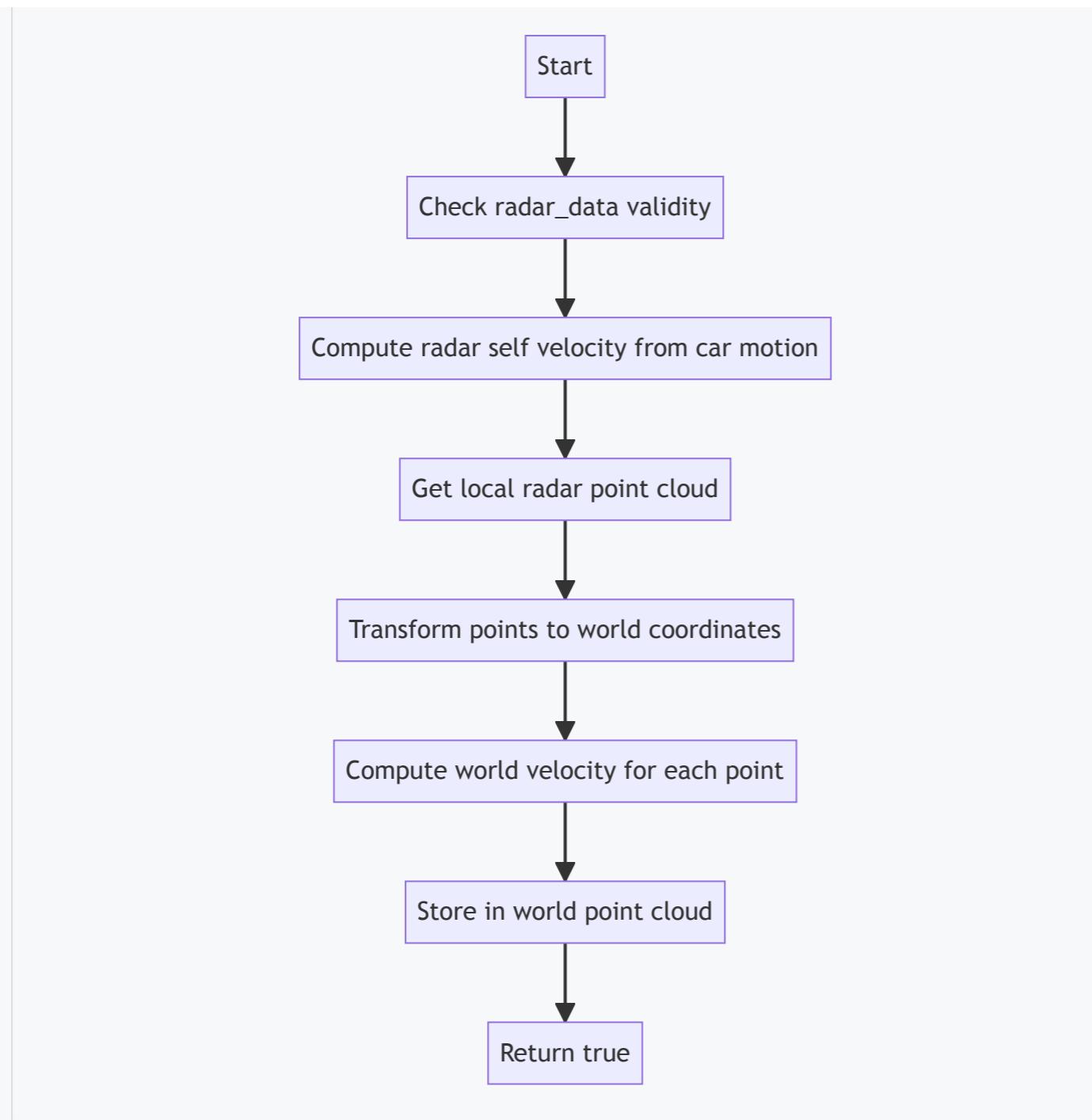
C++

```
</> Radar点云预处理核心代码
```

```

1 bool RadarPreprocessor::BuildWorldCloud(const RadarPreprocessOptions&
options,
2
3     (void)options;
4     if (radar_data == nullptr) {
5         LOG_ERROR << "radar data is null";
6         return false;
7     }
8     // coordinate transform

```



```

9  const Eigen::Vector3f& car_angular_speed = options.car_angular_speed;
10 const Eigen::Vector3f& car_linear_speed = options.car_linear_speed;
11 const Eigen::Matrix3f& radar2world_rotate =
12     radar_data->sensor2world_pose.rotation().cast<float>();
13 const Eigen::Matrix3f& radar2world_rotate_inverse =
14     radar2world_rotate.inverse();
15
16 const Eigen::Matrix3f& radar2novatel_rotate =
17     options.radar2novatel.cast<float>().rotation();
18 const Eigen::Matrix3f& radar2novatel_rotate_inverse =
19     radar2novatel_rotate.inverse();
20 const Eigen::Vector3f& radar2novatel_translation =
21     options.radar2novatel.cast<float>().translation();
22
23 // radar current velocity compute
24 Eigen::Matrix3f rotation_novatel;
25 rotation_novatel << 0, -car_angular_speed(2), car_angular_speed(1),
26     car_angular_speed(2), 0, -car_angular_speed(0), -car_angular_speed(1),
27     car_angular_speed(0), 0;
28
29 Eigen::Vector3f radar_rotate_motion_caused_vel =
30     radar2novatel_rotate_inverse *
31     (rotation_novatel * radar2novatel_translation); // vehicle-radar
32 relative // rotation motion
33 cased // radar motion (under
34 // radar coordinate)
35
36 Eigen::Vector3f radar_linear_motion_caused_vel =
37     radar2world_rotate_inverse * car_linear_speed; // vehicle translation
38 // motion caused radar
39 // motion (under radar
40 // coordinate)
41 Eigen::Vector3f radar_vel =
42     radar_linear_motion_caused_vel +
43     radar_rotate_motion_caused_vel; // radar motion under radar coordinate
44
45 // Build World Pointcloud
46 Eigen::Vector3d radar_local;
47 Eigen::Vector3d radar_world;
48
49 auto radar_cloud_ptr = radar_data->GetData();
  
```

```

50  if (radar_cloud_ptr == nullptr) {
51      LOG_ERROR << "radar cloud ptr is null";
52      return false;
53  }
54
55  const auto* local_polar_cloud = radar_cloud_ptr->Local();
56  if (local_polar_cloud == nullptr) {
57      LOG_ERROR << "local polar cloud ptr is null";
58      return false;
59  }
60  radar_cloud_ptr->World()->reserve(local_polar_cloud->size());
61  for (const auto& local_polar_pt : *local_polar_cloud) {
62      // Location Transform
63      radar_local << local_polar_pt.range * cos(local_polar_pt.azimuth),
64          local_polar_pt.range * sin(local_polar_pt.azimuth),
65          local_polar_pt.range *
66          sin(local_polar_pt.elevation); // note:the cosine of elevation
67                                         // angle is not used in x/y
68                                         // computation, since the
69                                         // precision
70                                         // of radar elevation is not good
71                                         // enough.(may need adjust after
72                                         // checking data status)
73      radar_world = radar_data->sensor2world_pose * radar_local;
74      // Velocity Transform
75      auto radar_relative_velocity =
76          radar_vel[0] * cos(local_polar_pt.azimuth) +
77          radar_vel[1] *
78          sin(local_polar_pt.azimuth); // radar velocity to point_i
79
80      // emplace back
81      radar_cloud_ptr->World()->emplace_back(
82          radar_world.x(), radar_world.y(), radar_world.z(),
83          radar_relative_velocity + local_polar_pt.rad_vel);
84  }
85  return true;

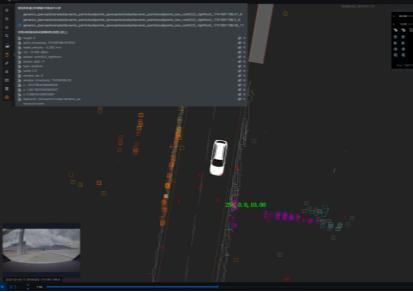
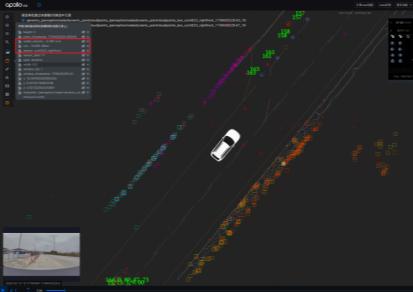
```

经过预处理的点送入Radar感知模块进行常规动、静目标检测。目前动目标检测依赖动态点，观察到RT6车辆行驶过程中，当侧向遇到栅栏等物体时，在栅栏位置生成的Radar点的绝对径向速度往往不准确，容易产生误检。

对于栅栏速度不准问题，感知内部进行了深入的评估，验证了前置预处理、解模糊等各个环节的算法精度，最终确定是角雷达（conti523）在该场景测速不准。举例如下：

1.1.2. 具体表现

下表截图中，方框为Radar动态点。(具体阅读方法可参考 [Radar基础链路可视化插件设计](#))

caseid	ov链接	截图	
dataset_flow-107-17435076506553123_JME2096_20250304160000_1741081189_1741081206	link		栅栏速度不准较多，容易形成动态误检
dataset_flow-107-17435076506553123_JME2100_20250213120146_1739420229_1739420246	link		栅栏速度不准较多，容易形成动态误检
dataset_flow-107-1746513357871657_JME1551_20250115200000_1736944485_1736944499	link		绿植也会有速度不准。
dataset_flow-73-1739762401262614_JME1535_20250113080000_1736732193_1736732203	link		栅栏速度不准较多，容易形成动态误检

1.1.3. 总结

总体来说，感知链路可以通过较大动静阈值来进一步抑制侧向栅栏速度不准问题，但这种方法同时也会大幅降低Radar对静转动车辆的速度报出灵敏性。而这在城市场景十分重要。

1.2. 窄路双萝卜互相干扰底噪抬高

1.2.1. 问题背景

在窄路工况下，当存在两台或以上搭载Conti 512/523雷达的萝卜车时，车辆之间存在较高的雷达相互干扰风险。

雷达通过发射电磁波并接收其遇障碍物反射的回波来进行探测。当多部同频段雷达在近距离同时工作时，一部雷达可能会接收到由另一部雷达发射的电磁波。对于接收方雷达而言，这个外部信号并非自身发射波的反射，因此会被视为无效的噪声或虚假点云，从而导致雷达感知的背景噪声水平（即“底噪”）被显著抬高。

1.2.2. 具体表现

caseid	ov链接	截图	
JME1579_20250516084832	link		道路上莫名误检变多
2			

1.2.3. 总结

此问题在巷道、单车道路宽有限等场景下尤为突出，车辆上来时需要重点解决的干扰问题。

此问题在巷道、单车道路宽有限等场景下尤为突出，是部署多车协同或车队行驶时需要重点解决的传感器干扰问题。

2. RT5相关问题

RT5目前各中问题暂时符合预期。正在持续收集中。