Radar Sensing Architecture Design #2531

Closed **scepter914** started this conversation in **Design**

scepter914 on Apr 19, 2022 Collaborator edited

Summary

We would like to initiate the discussion for sensing architecture with radars for Autoware.
The target of this discussion is the messaging systems used with radar sensors and sensing pipeline for radars.

We would like to make a separate discussion for the perception architecture including radar.

Messages

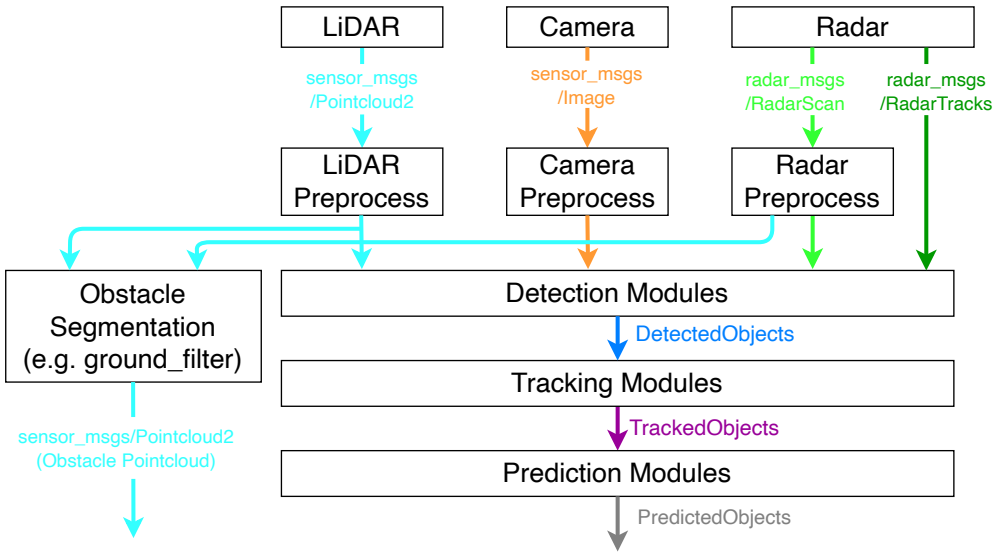
To summarize, We suggest

- ros-perception/radar_msgs/msg/RadarScan.msg for radar pointcloud.
- ros-perception/radar_msgs/msg/RadarTracks.msg for radar object.

For more detail, please see radar pointcloud message design and radar object message design.

Whole pipeline

The proposed pipeline follows the Sensing and perception architecture proposal.



This figure can be found at https://github.com/scepter914/autoware-radar-architecture-proposal/blob/main/sensing/figure/sensing_pipeline.drawio.svg.

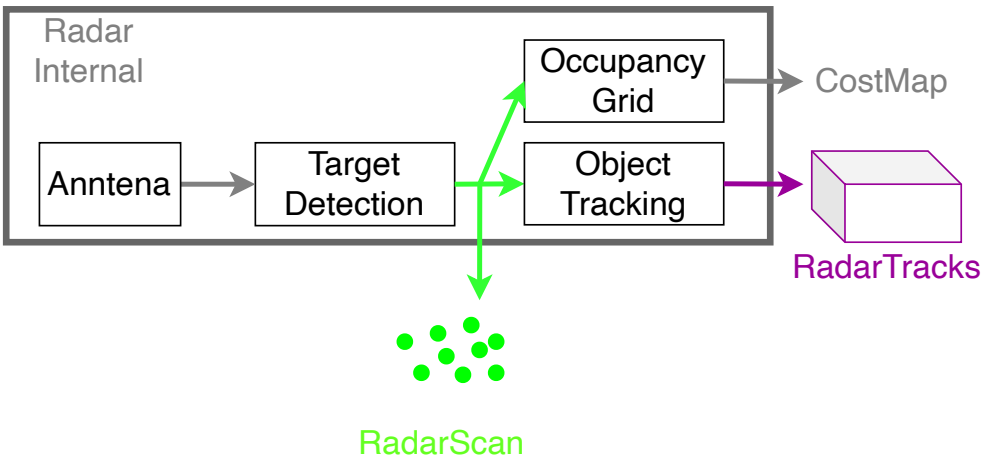
Radar driver

Category
Design

Labels
component:percept...
component:sensing

9 participants

For now, We suggest that Autoware radar drivers should support `ros-perception/radar_msgs/msg/RadarScan.msg` and `autoware_auto_perception_msgs/msg/TrackedObjects.msg`, because these two outputs are more useful for sensor fusion in the sensing and perception module than others.



This can be found at https://github.com/scepter914/autoware-radar-architecture-proposal/blob/main/sensing/figure/radar_driver.drawio.svg.

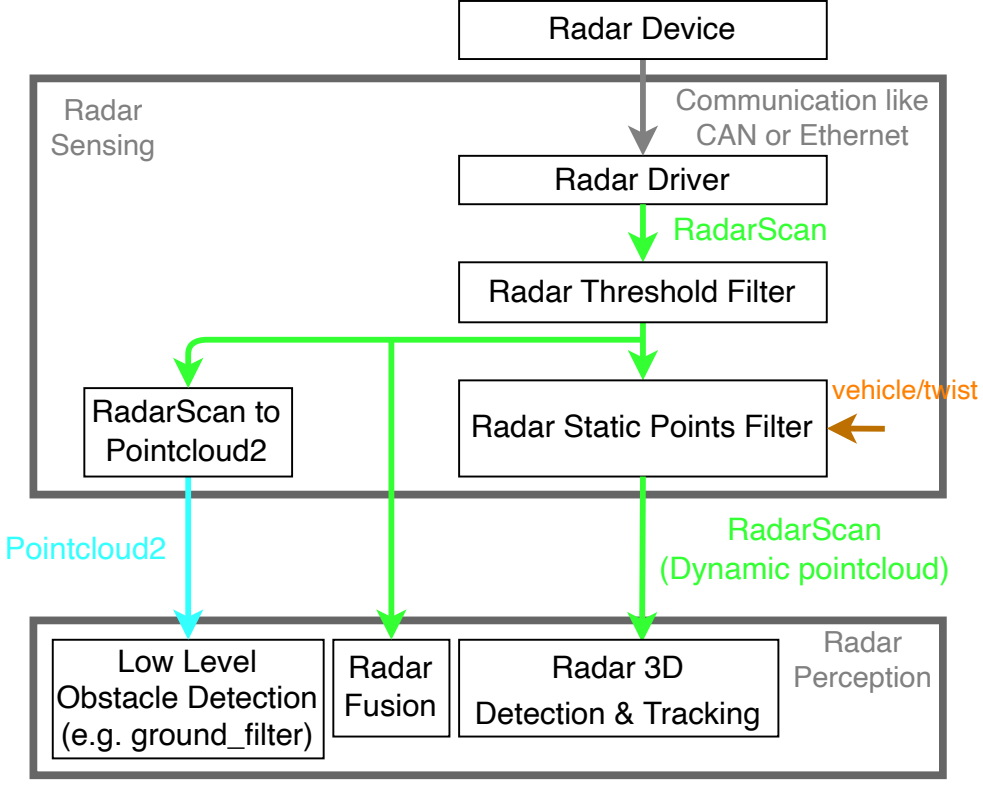
For more detail, please see radar driver design.

Radar sensing pipeline

To sum up, We suggest:

- In the sensing layer, the radar preprocess package filters noise through the `ros-perception/radar_msgs/msg/RadarScan.msg` message type.
- For use of radar pointcloud data by LiDAR packages, we would like to propose a converter for creating `sensor_msgs/msg/Pointcloud2.msg` from `ros-perception/radar_msgs/msg/RadarScan.msg`.

For more detail, please see radar sensing design.



This figure can be found at https://github.com/scepter914/autoware-radar-architecture-proposal/blob/main/sensing/figure/radar_sensing.drawio.svg.

Discussion list

- [ ] Message definition
- [ ] Radar driver design
- [ ] Radar sensing pipeline design
- [ ] Document location
  - We see autoware-documentation, but it is not clear where to put the architecture design document.
  - After discussions and if maintainers can instruct where to put the document, we will send a PR for it.

↑ 1 👍 7 🚀 1

8 comments · 15 replies

Oldest | Newest | Top

xmfcx on Apr 19, 2022 Maintainer

ros-perception/radar_msgs/msg/RadarScan.msg doesn't seem to have:

- https://en.wikipedia.org/wiki/Signal-to-noise_ratio
- https://en.wikipedia.org/wiki/Radar_cross-section
  values for points, SmartMicro radars provide these with raw radar point clouds, do you have any opinion on this?

↑ 1 12 replies

Show 7 previous replies

scepter914 on Apr 21, 2022 Collaborator Author edited

@xmfcx

> I see, also if we want to use point cloud algorithms, we will need to manually convert it to pcl point cloud then. An example use case is to use conditional euclidean clustering to cluster based on similar fields in addition to distance.

I suggest for converter for creating from `ros-perception/radar_msgs/msg/RadarScan.msg` to `sensor_msgs/msg/Pointcloud2.msg` for use of radar pointcloud data by LiDAR packages.
In the cases that convert to (x, y, z, RCS) or (x, y, z, doppler_velocity) is used, you can apply to euclidean_cluster package in Autoware.
(I mean it in convert RadarScan to Pointcloud2 section in radar sensing design)

I think deep discussion is necessary, but I can suggest one of the candidates that to use something like conditional euclidean clustering, converter select data field of `Pointcloud2` as follows:

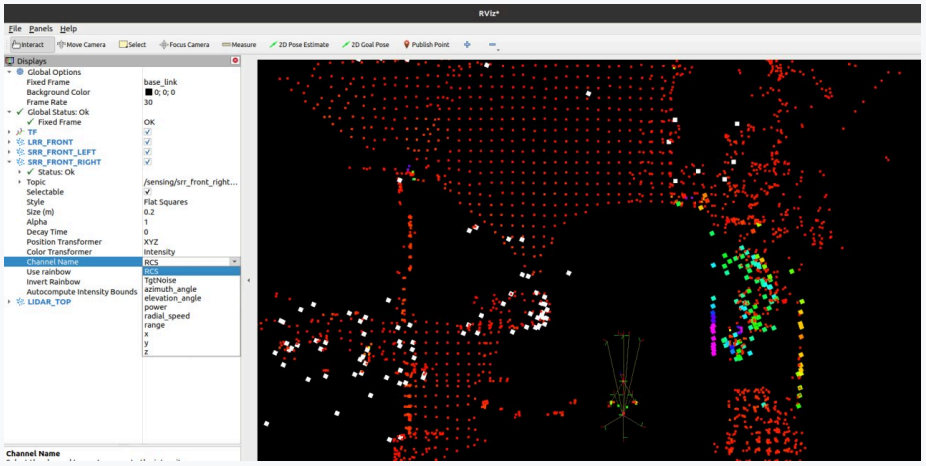
- (x, y, z, RCS)
- (x, y, z, doppler_velocity)
- (x, y, z, RCS, doppler_velocity)

(x, y, z, RCS) and (x, y, z, doppler_velocity) can be used for LiDAR package, and (x, y, z, RCS, doppler_velocity) can be used for radar package using PCL.

mschratter on Apr 22, 2022 edited

@scepter914 @xmfcx
Thanks for starting this discussion! It is nice to see that Autoware is going to use Radar as well.
We are using in our vehicle radars from Smartmicro, their ROS2 driver supports directly Pointcloud2. I do not see a benefit in using the radar_msgs/RadarScan, especially in using the sensor in a cluster mode. Since the Pointcloud2 has the possibility to store multiple layers and is therefore more future-proof. For the object mode the radar_msgs/RadarTrack might make sense.
As you mentioned above parts of the PCL can be directly used and also different layers can be directly visualized in RVIZ.

Data from our vehicle:



It would be great, if all radar drivers would use the same field names, to be flexible in replacing sensors.

What do think about this?

scepter914 on Apr 22, 2022 Collaborator Author edited

@mschratter

Thank you for suggestion from actual use.
I understand benefits of Pointcloud2 in PCL usage and direct usage of visualization.

I think data field of Pointcloud2 is suitable for small group development, however, not suitable for large open source development.

- When message type is clearly defined like RadarScan, it is easy to update packages when data schema is changed.

If Pointcloud2 is used, it is hard to understand change of data schema and update radar packages.

- When Pointcloud2 is used, everyone come to define different data field.

The data field of Pointcloud2 is useful because anything can be defined.
In your case, (x, y, z, dopper_velocity, power, RCS, Noise) is defined as data field for smartmicro radar (Maybe data field is here), but how about ARS408 is used?
As found in ros-perception PR autowarefoundation/autoware.ai#2455, each radar output more different data type than LiDAR do, and it is easy to lead to define different data field just as one likes.
After all of that making radar preprocess packages, it would be useless because of lack of message definition.

Of course, I know benefits of Pointcloud2 as you said, so I suggest to prepare

- The converter from RadarScan to Pointcloud2

- The rviz plugin for RadarScan like tier4_perception_rviz_plugin.
  I think these are not much cost to make.

xmfcx on May 23, 2023 Maintainer

Sorry for more than a year of delay. **@scepter914** I think we can go with the https://github.com/ros-perception/radar_msgs/tree/ros2/msg and update the Autoware documentation accordingly. Is this OK with you?

cc. **@yukkysaito** **@miursh** **@kaancolak**

scepter914 on May 30, 2023 Collaborator Author

> Sorry for more than a year of delay. **@scepter914** I think we can go with the https://github.com/ros-perception/radar_msgs/tree/ros2/msg and update the Autoware documentation accordingly. Is this OK with you?

Thank you for reply.
I would appreciate if you could update the documentation, but It is different from what I said on this proposal.
So I commented in autowarefoundation/autoware-documentation#328.

👍 1

scepter914 on Apr 19, 2022 Collaborator Author

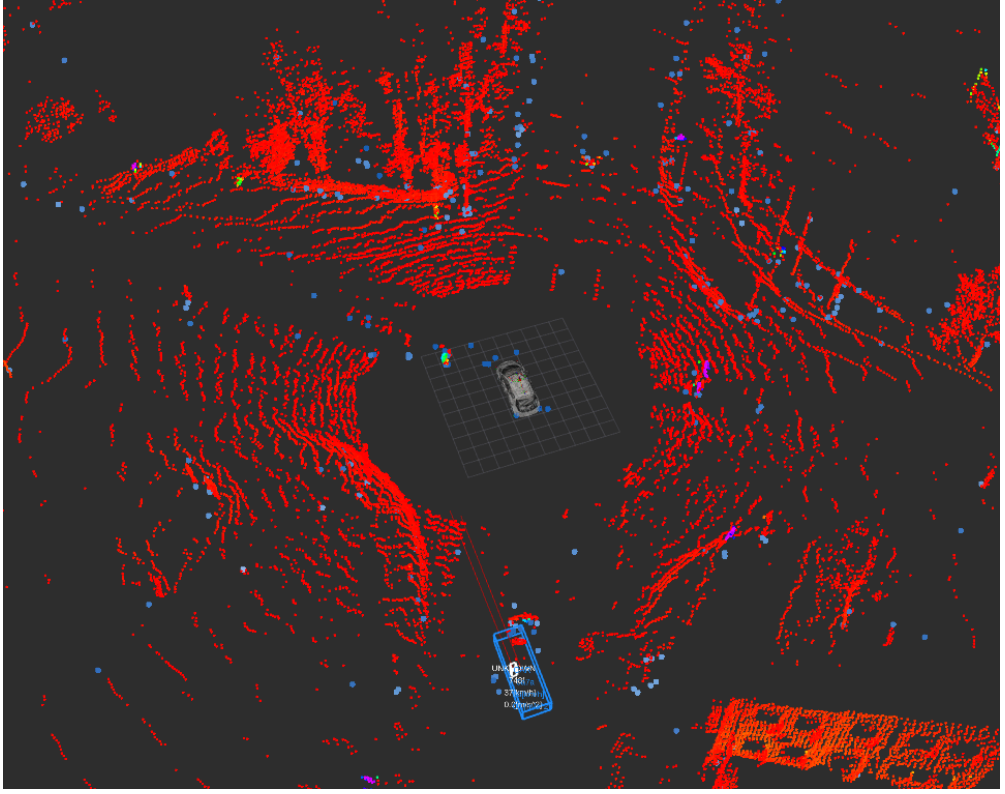
I found a mistake in this proposal, so I fixed it.

↑ 1 0 replies

FranzAlbers on Dec 5, 2022

I am currently working on a driver for our radar sensors as well. For that, I adapted the ars408 repository (https://github.com/tier4/ars408_driver, thanks for implementing this **@scepter914**!) to our needs and followed the proposed radar architecture: We publish RadarTracks and convert them with the radar_tracks_msgs_converter.

Just like the ars408 does, our radar sensors estimate object orientations. As far as I can see, these are currently lost when RadarTracks are published, because a RadarTrack does not contain the orientation of the object. Thus, our objects are always oriented in the same direction as our vehicle, though they are actually slightly rotated:



Is there already a way for dealing with object orientations? How are they considered in the current architecture?

↑ 1 👍 1 2 replies

yukkysaito on Dec 7, 2022 Maintainer edited

The only three interfaces defined by Autoware.

- DetectedObjects
- TrackedObjects
- PredictedObjects

Intermediate representation is not defined to generate DetectedObjects or TrackedObjects.
So you can use radar_msgs or define your own types. It is only good if the final result is DetectedObjects or TrackedObjects.
@scepter914 any comments?

👍 1

scepter914 on Dec 7, 2022 Collaborator Author edited

@FranzAlbers
Thank you for sharing practical use case.

> Is there already a way for dealing with object orientations?

No, there is not now.

> How are they considered in the current architecture?

As said from **@yukkysaito** , the perception interface of Autoware is defined to `DetectedObjects` , `TrackedObjects` , and `PredictedObjects` , however, other message is defined by own cases.
For example, TIER IV use DetectedObjectWithFeature in perception module.

Same as that, TIER IV use `RadarTrack` to follow ROS-perception definition for radar data.
As an interface of the current `RadarTrack` , `RadarTrack` doesn't have orientation information from past discussions, especially this discussion.
As OSS development, I think radar's interface is better to change ros-perception/radar_msgs as a ROS standard.

Anyway, as said from **@yukkysaito** , I also recomend for your development that your ROS radar driver publishes directly `TrackedObject` to add orientation information.
I think you can do
by using and modifing this function
https://github.com/autowarefoundation/autoware.universe/blob/4a75d7c0ddbd88f54afaf2bb05eb65138a53ea60/perception/radar_tracks_msgs_converter/src/radar_tracks_msgs_converter_node/radar_tracks_msgs_converter_node.cpp#L160

👍 1

FranzAlbers on Dec 8, 2022

Thank you very much for your help, **@yukkysaito** and **@scepter914**!

Publishing the `TrackedObjects` directly from the radar driver was exactly the hint I needed.
I've implemented this and we now have the orientation as well as some additional information provided by the sensors like the existence and classification probabilities available.

↑ 1 👍 2 0 replies

cycyc1949 on Oct 19, 2023 Collaborator

Radar has a great advantage in velocity detection, and architecture design can take advantage of this for tracking modules.

↑ 1 ❤️ 1 0 replies

hzkkk1997 on Oct 11

@scepter914 Hello, sorry to bother you, I have a question about the predicted speed of static obstacles(https://github.com/orgs/autowarefoundation/discussions/5330), could you give me some suggestions? Thanks:

↑ 1 1 reply

scepter914 last month Collaborator Author

I commented.

smileghp on Oct 17 Collaborator

This is a great design, which abstracts the pre-processing of sensor data into a standard process. We have borrowed this solution in practice, but encountered some problems in the implementation of the ORIN-based solution, such as multiple data copies and the output topic needs to be copied from the CPU memory to the GPU memory in the subsequent detection module. In the implementation, we changed the topic data to use object pointers to pass data. If you are interested, please feel free to communicate.

↑ 1 👀 2 0 replies

scepter914 last month Collaborator Author

As first prototype, we already integrate to autoware.universe and start to maintain the packages.
And new design start to be discussed at https://github.com/orgs/autowarefoundation/discussions/5264.
So I close this discussion and if you want to ask questions, please make a new issue.

↑ 1 0 replies