ILR - 11

Perception System Using Stereo Vision and Radar

Team A - Amit Agarwal
Harry Golash, Yihao Qian, Menghan Zhang, Zihao (Theo) Zhang
Sponsored by: Delphi Automotive

April 18, 2017

Table of Contents

1. Individual Progress

Since the previous progress review, I was tasked with clustering, filtering and visualizing the tracking points received by the Radar through Ethernet. For this I worked with Harry and obtained data about tracking points from the Radar and visualized them in RViz. In addition, I am working on implementing a clustering and filtering algorithm for the radar based on an advanced paper on tracking vehicles using radar. In addition, based on the feedback I received from Professor Dolan, I have been investing more time in managing the team, which has resulted in most of the team being more productive.

1.1 Radar Setup in ROS

Over the past couple of weeks, I have been working on getting sensible data from the Delphi ESR 2.5 Radar. The usual procedure, as described in the previous ILRs, is used to connect the Radar to the test car. Currently, no form of filtering is applied before visualizing the data received but that will change as soon as the implementation is complete.

To move from basic data visualization to complex filtered and clustered data, a few things need to be modified in ROS to implement the algorithms. The first will be to get certain information that will be used in the algorithms, i.e., the range, the range rate, the angle and the amplitude of all 64 detection points and the scan type (mid-range or long-range) for the detections. This will, in turn, lead to the modification of the message type used to transfer data from one node to the other. Then, certain classes will need to be created or imported to hold algorithm specific data from one frame to the next to do temporal association between data points. Currently, no information about the previous frames of data are stored or used.

The paper based on which I will be implementing the filtering is "Tracking vehicles using radar detections" from the 2007 IEEE Intelligent Vehicles Symposium. It presents a novel method to cluster, filter and track vehicles based on cluster constellations and a generalized version of joint probabilistic data association to confirm tracking hypotheses while minimizing false positives. It starts by defining the problem. The state vector for a vehicle, l, at a discrete time instance, k, is defined as,

$$\mathbf{z}_k^l = [\zeta_{x_k}^l \ \zeta_{y_k}^l \ \Psi_k^l \ v_k^l \ \dot{\Psi}_k^l \ \dot{v}_k^l]^T$$

where ζ represents the Cartesian coordinates of the vehicle with respect to the host vehicle (with radar), ψ represents the lateral angle deviation or the heading angle of the vehicle with respect to the host vehicle, and ν represents the velocity of the vehicle in the heading angle. The state vector is explained in more detail in figure 1 where the host vehicle is depicted in black and the tracked vehicle is depicted in gray.

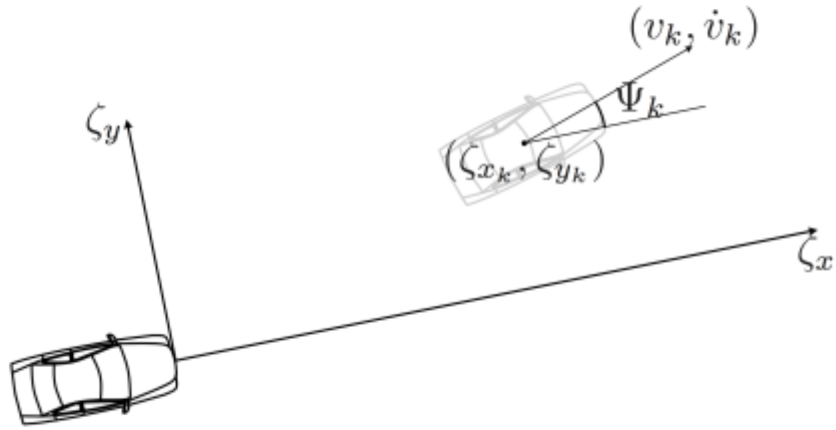

Figure 1: Vehicle state vector

The algorithm defines a vehicle motion model and a sensor model for the vehicles and the radar, respectively. These models incorporate all variables found in the vehicle and radar systems. A generalized version of JPDA is used for data association and clustering using these models. Approximations of these models are then used with a Kalman Filter to accurately predict the motion of the vehicles being tracked.

If completely and accurately implemented, the radar performance will be significantly better than its current state. The priority is to complete the spatial clustering. Once that is completed, temporal filtering will be considered to improve tracking ability and sensor fusion. The Kalman Filter will likely be implemented using the Generic Sensor Fusion Package (GSFP) in ROS.

1.2 Radar Visualization

The Radar was visualized in RViz as recommended by Sean. Previously, other avenues, one of them being OpenCV, were explored for visualizing the data. After careful consideration and comparison, RViz was the better choice since the system integration was performed in ROS.

Some changes need to be made to the visualization pipeline in ROS to accommodate clustering and filtering. The major change is that the data coming from the radar will go through a few nodes for filtering and clustering before being published to RViz. Earlier, the data points received from the radar were converted to Cartesian coordinates and directly published to RViz. For the purpose of testing and implementation of the filtering algorithms, the unfiltered data will also be visualized in RViz to compare it to the filtered data. Filtered and unfiltered data will be represented using different shapes for clarity.

2. Challenges

The major challenges faced were due to minimal knowledge of algorithms for data filtering from the Radar and their implementation. In addition, the test car for the project had broken down about a month ago.

2.1 Data Filtering Implementation

The knowledge possessed about filtering the Radar had been minimal up until this point. A lot of research was done in the past week but the main issue seems to be the lack of implementation details about the papers I read and found useful. The academic papers explain the math well, but implementing the methods presented in the paper requires extensive know-how of the intricacies attached with the methods. The plan is to attempt to eliminate as much of this as possible, using pre-written libraries like boost, bayes++ and GSFP. If no unexpected problems are encountered, the data filtering should be implemented and tested in time for SVE. On the other hand, it is hard to anticipate the kinds of problems that may arise.

2.2 Test Car

The test car used by the team, a Volvo S60 2.5T AWD, was not in working condition anymore because of hitting an animal on the interstate over spring break. The car should be ready to begin testing as soon as Harry brings it to the lab. This has been a major hurdle in getting actual road data for testing and validation.

3. Teamwork

The work this week was performed by all 5 team members. Harry and I worked on the Radar and getting data from the Radar and visualizing the data in real-time. I also worked on reading, analyzing and implementing Radar filtering methods. Yihao and Zihao worked on the stereo vision node to get better depth measurements at a quicker rate. They also created a visualization which displays the results of the object detection/classification algorithm. Menghan worked on the ROS node for object tracking. Most of the teammates worked coherently. The team is capable of still completing the project by the SVE but that would require continuous efforts of all the teammates which is something I have been working on in the past few weeks, as the project manager. I hope that all teammates come together and finish this project regardless of how much or how less they have worked on this until now. I will continue to try my best to get the team to work together and efficiently for whatever time is left in this semester.

4. Future Plans

In the future, we plan to complete system integration and move forward with validating the experiments in our test plan. Once this has been achieved, the team will proceed to prepare for

SVE. The team is far behind schedule in terms of progress, but most of the teammates have been spending considerable amount of time working on the project. I, personally, have changed my style of working to accommodate better team leadership in the last 3-4 weeks. By sharing progress updates every 2-3 days, we have significantly improved our productivity. I hope that we can present a good demo for SVE, although I am not confident that we will achieve that. I will certainly motivate the team to get as much done as possible.

5. References

[1] "Delphi ESR 2.5". Delphi Support Center. N.p., 2017. Web. 12 Feb. 2017.
[2] "TCP/IP Network Programming". Vichargrave.github.io. N.p., 2017. Web. 24 Mar. 2017.