Carnegie Mellon University

Team B : AutoPirates

ILR2

Author:
Shiyu Dong

Team Member:
Tushar Chugh
Tae-Hyung Kim
Bikramjot Hanzra
William Seto

October 23, 2015

1. Individual Progress

This week I mainly worked on visualizing radar data using the OpenCPN library with BR24 radar plugin and compiling the source code of the OpenCPN and BR24 radar plugin.

1.1 visualize radar data using OpenCPN

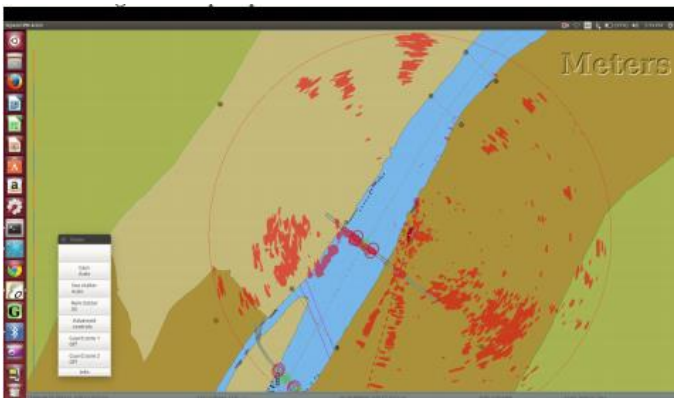
The OpenCPN tool is an open source library to plot chart and navigation for marine applications. The BR24 radar plugin is a plugin for OpenCPN that can visualize data for Simrad, Lowrance or B&G broadband radar. And the Simrad Radar is what we used for this project now.

I did research on how to use OpenCPN with the BR24 radar plugin. My work included installing OpenCPN, finding the offline map for 3 river area that OpenCPN will be able to use and configuring the BR24 radar plugin.

The map we used in OpenCPN is the Official Inand Electronic Navigation Charts(IENC) of U.S. inland waterways. The IENC map provides a map for the three river area including bridges.

Then we tried to receive the data through Ethernet on boat using OpenCPN. After failure of many times, we finally got connected to the radar and successfully visualized the radar data as figure 1.

Figure 1: Visualization of radar data



1.2 Compile OpenCPN with radar plugin

Since we would like to publish radar data to ROS nodes to get the raw data from radar, it's necessary for us to compile the source code for OpenCPN and the radar plugin. And I have successfully installed all the required environment to compile OpenCPN and make a package.

All the required environment can be installed using this command:

```
$ sudo apt-get install build-essential cmake gettext git-core gpsd gpsd-clients libgps-dev wx-common libwxgtk3.0-dev libglu1-mesa-dev libgtk2.0-dev wx3.0-headers libbz2-dev libtinyxml-dev libportaudio2 portaudio19-dev libcurl4-openssl-dev
```

1.3 Revise CoDR

This week I also revised the validation experiments and risk management of CoDR. According to the feedback from professors and TAs, I made changes to the risk management and picked the top 3 risk as required by system engineering presentation. For the validation experiments, I mostly changed the structure for spring validation experiment. Instead of using separate parts, I combined it into sequence of steps.

1. Get the boat on the river
2. Keep 5 static obstacles in the path
3. Switch on the engines
4. Enter the destination through OCU
5. The boat navigates autonomously and avoids obstacles from start to destination.
6. Display the real-time location of the boat in OCU and the estimated path
7. The boat arrives the destination
8. Show logged file.
9. Turn off the engine

We are still working on revising the CoDR, which will be presented on progress review 2.

2. Challenges

The challenge was to compile OpenCPN and the radar plugin. William and I mainly worked on this. The OpenCPN is not a well-developed library so that there were still some problems when we tried to compile. To compile OpenCPN, there are some required programs, including one called wxWidgets. The wxWidgets now have 2 versions, 3.0 and 2.8. To make sure that the OpenCPN works well, we just installed both versions.

And then we've tried different ways to compile, including putting the plugin in or out the OpenCPN folder and compiling them together or separately. William first succeeded in compiling the OpenCPN without the plugin first, then putting the source code for plugin in the OpenCPN folder and compiling the plugin. But I failed when trying that way. I then tried reconfigure the compiling environment but that was still not working. William and I worked together to fix it and finally we figured out what the problem was: I had installed OpenCPN directly from source code and also tried to generate a package and install it. So actually there were two locations that OpenCPN was installed in my system, which caused confusing when I tried to install the plugin. Once I deleted the one I installed directly from source code, the problem was fixed.

3. Teamwork

Tushar Chugh: He studied the use of SBPL library for path planning. He revised the functional and cyberphysical architecture of CoDR and acts as a good project manager.

Tae-Hyung Kim: He invested the possibilities of using open source library to do the path planning and suggested us to use SBPL library. He also revised the trade study on path planning part of CoDR.
Bikramjot Hanzra: He worked on implementing frame transformation in ROS, which is what we need to integrate perception output to path planning algorithms. He also updated the website and corrected the language errors in CoDR.
William Seto: He worked with me on using OpenCPN to visualize radar data and compiled the OpenCPN with BR24 radar plugin. He also revised the project description, user story and requirements part of CoDR.

4. Future Plan

The plan for perception team is to publish raw data from radar to ROS nodes.

The plan for path planning team is to learn state-of-art algorithms for path planning and choose which one to implement first.

The plan for project management is to prepare for the presentation about validation experiments and risk management for system engineering class.

My plan is to work with perception team to publish raw data to ROS nodes. Besides, I'm starting to read one paper that corresponding to the radar perception: *Collision Avoidance for Vessels using a Low-Cost Radar Sensor*.