William Seto

Team B: Auto Pirates

Teammates: Bikramjot Hanzra, Shiyu Dong, Tae-Hyung Kim, Tushar Chugh

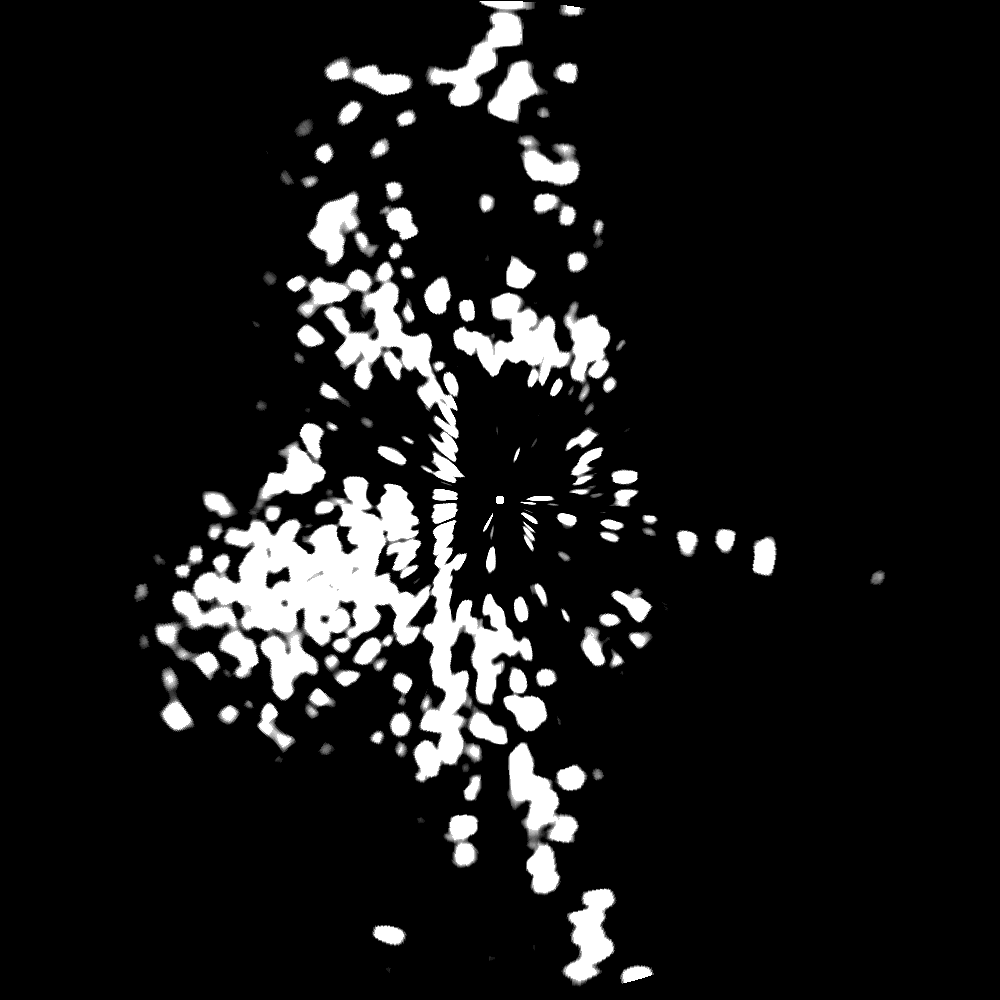
ILR04

November 13, 2015

Individual Progress

For this lab, I focused on improving the visualization of the radar data. This followed after we collected some radar data at NREC. This was important since we will need to be able to visualize the data real time, after receiving it as a ROS node, to ensure that when we use ROS to log the data, it is the correct data that we would be seeing on the boat’s radar display.

As mentioned last time, I had initially visualized radar data with Python’s matplotlib library. While it was OK for viewing a few frames, it wasn’t enough to view at almost real-time. After playing around with some other plotting libraries and attempting to learn OpenGL, I realized that the right thing to do would be to convert the radar image, which is in polar coordinates, into a regular image in cartesian coordinates. This is important since later in the pipeline, we will be able to leverage tools and algorithms from libraries such as OpenCV that can be used to process our image. Seen in Figure 1 below, the conversion of the image to cartesian coordinates visually looks good. We will have to see if our method is good enough for our later image processing.

Finally, I worked on preparing the test plan for our first field test. We identified key locations on the river which we thought would provide us good radar data. Additionally, I put together a rough time schedule of the trip, so that we can complete as many test cases as we plan to do. I also helped the rest of the team to set up the code I had been working on for passing radar data into ROS as well as visualizing it.

*Fig 1: Radar image converted to cartesian coordinates*

Challenges

I ended up struggling a lot more than I thought with plotting the data. Initially, I tried to make a few quick hacks that I thought might speed up the original way I was plotting. When that didn’t work, I tried looking for other Python plotting libraries. However, there either wasn’t any support for polar plots, or the documentation wasn’t clear. Then I decided to try OpenGL in C++, since the program we’ve been using, OpenCPN, renders the data using OpenGL. I spent some time trying to learn how it was done, but realized the effort would not be worth it.

Even when I decided to convert the image to cartesian coordinates, I spent some time thinking and researching techniques, since the conversion will never be perfect. Some information will be lost, or has to be interpolated in the new image. This may be something we have to deal with in the future.

Teamwork

Shiyu Dong

Shiyu focused on finishing up the PCB design and layout with Tushar. I also worked with him to set up the webcam which we will be using to record video in our field test.

Bikramjot Hanzra

Bikram worked on filtering of the radar data, applying some of the morphological operations. He also set up the structure of our code base and the repository, along with a wiki containing our documentation.

Tae-Hyung Kim

Tae-hyung continued to understand and experiment with the capabilities of the Stage simulator. He worked with Tushar to load a custom occupancy grid.

Tushar Chugh

Tushar worked with Shiyu to finish the PCB layout and also completed the bill of materials. Additionally, he mapped out a region of the river which we may use for our SVE and created an occupancy grid for it.

Plans

Although a team effort, the plan next would be to collect as much radar data as possible from the river. After this, we can start the serious work of analyzing and filtering the data. While Bikram experiments with blob discovery and contour extraction, I will attempt some preprocessing techniques before the radar image is converted to cartesian coordinates. Primarily, this involves calculating an occupancy probability for each cell of the radar image. Each azimuth scan of the received data corresponds to about 0.2 degrees, but the specifications indicate a beamwidth of about 2.6+ degrees, so we can do some simple filtering to clear out some noise.

As alluded to earlier, I will also continue to investigate the process of converting the image to cartesian coordinates, to see if there are some more high fidelity techniques.