ILR #10: Progress Review 11

Zihao (Theo) Zhang- Team A

April 6th, 2017

Teammates: Amit Agarwal, Harry Golash, Yihao Qian, Menghan Zhang

Individual Progress

Since the last progress review, I have continued to facilitate progress on the system integration in ROS. The comparison between the following two figures serves as a straightforward way to show our progress on system integration during the past two weeks.

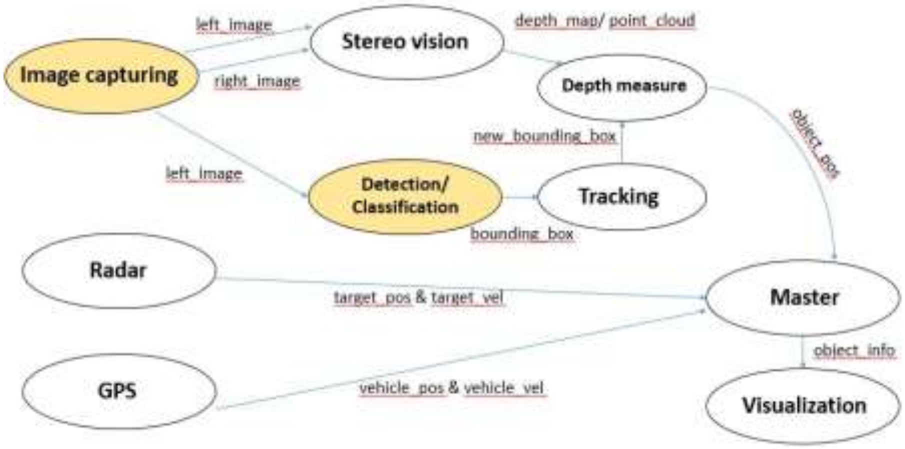

Figure 1. System integration status by Progress Review 10

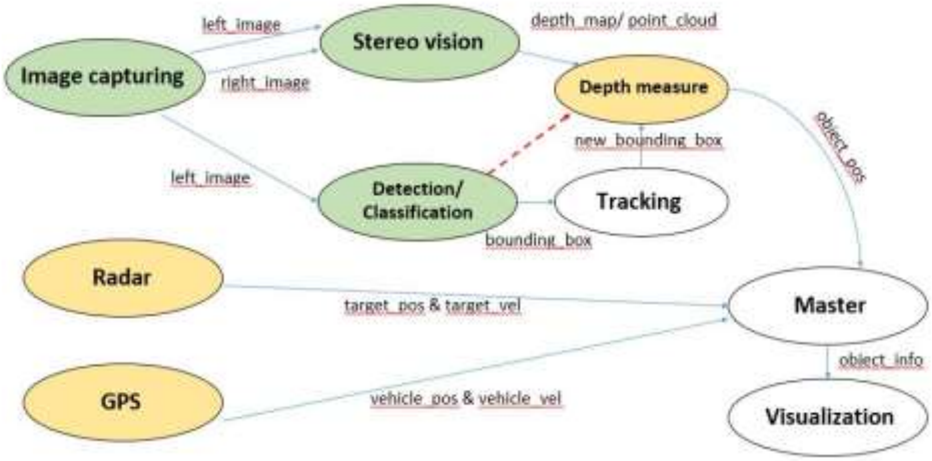

Figure 2. System integration status by Progress Review 11

In the outlines shown above, each ellipse represents a module built as a node in ROS. The arrow between two nodes represents the flow of messages under a certain topic (topic name shown as the text next to each arrow) that are published by one node and subscribed by the other node. The ellipses marked in yellow stands for the modules that is in progress (under construction), while the ellipses marked in green stands for the completed modules.

I finished building the image capturing module, which is adapted from the *flea3* package by Kumar Robotics on the GitHub, as well as the stereo vision module for our system.

By the last progress review, we could only acquire raw images from the two cameras continuously using the image capturing module. However, the stereo vision module requires the input (subscribed) images to be processed already. Therefore, I adapted and included an additional image processing node in the image capturing module so that it can provide the rectified color images to the stereo vision module.

The post-processing of images also requires the complete set of camera information such as the intrinsic matrix and rectification matrix. Thanks to the data from the previous camera calibration, I generated the correct *yaml* files for providing the camera information to the image capturing module. This way, the properly rectified images can be created and sent to the rest of the system.

For the stereo vision node, I adapted the ELAS package on ROS, which wrapped the Library for Efficient Large-Scale Stereo Matching (LIBELAS), and connected it with the rest of the system. The depth information stored in the disparity map (published by the stereo vision module) can be now visualized in Rviz.

For the GPS module, I have experimented two different ways of migrating the existed work on our Adafruit GPS to ROS. One way is to take advantage of the *rosserial_arduino* package, which allows the user to publish or subscribe ROS messages directly using Arduino. Another way is to use the *nmea_navsat_driver* package, which takes unprocessed NMEA sentences from the GPS at serial port and parses the data into useful position and velocity information. Currently, the (useful) GPS information can be published in ROS using NMEA driver package, while I am trying to solve some communication issues encountered by the other method. Since we have had at least one working solution tested, we can expect to have the GPS module integrated with the rest of our system soon.

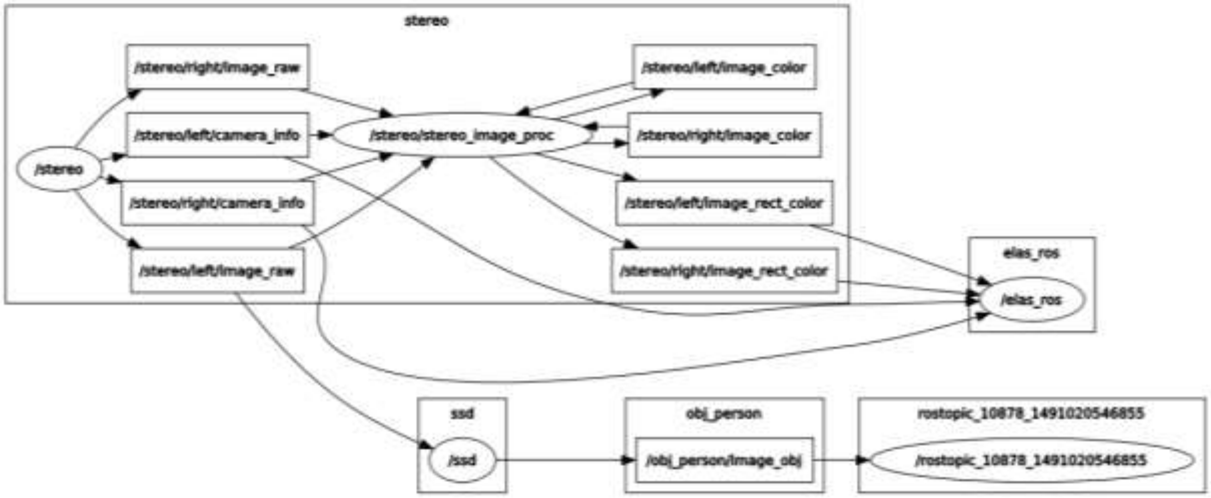
stereo
/stereo/right/image_raw
/stereo/left/camera_info
/stereo
/stereo/right/camera_info
/stereo/left/image_raw
/stereo/stereo_image_proc
/stereo/left/image_color
/stereo/right/image_color
/stereo/left/image_rect_color
/stereo/right/image_rect_color
elas_ros
/elas_ros
ssd
/ssd
obj_person
/obj_person/image_obj
rostopic_10878_1491020546855
/rostopic_10878_1491020546855

Figure 3. Current system architecture in ROS (without the depth measure node added)

Challenges

The relatively limited knowledge in ROS would still be a challenge for me to make significant progress independently within a very short period of time. To address this issue, Yihao, Menghan, and I have been setting up daily meetings to work together on the system integration. This collaboration should hopefully provide some efficiency and acceleration on our overall progress.

On the technical side, the biggest challenge we have encountered during system integration is that the processing speed of the finished part of the system is still far from ideal. Even with the use of the strong computing power offered by our new desktop, displaying the disparity map (in greyscale) in real time is still very challenging. To address this issue, we have been working on adapting the existing code in the stereo vision module to offer better efficiency. For example, Yihao has been working on getting the same program run on GPU instead of CPU, and the result from some initial experiments has shown the possibility for further improvements.

Teamwork

Amit Agarwal:

Amit has been working with Harry on the radar. They were able to display the (unfiltered) detection result from the radar in Rviz by this progress review. Amit also lead the team to discuss and revise our test plan and SVE.

Harry Golash:

Harry has been working with Amit on the radar. They were able to display the (unfiltered) detection result from the radar in Rviz by this progress review.

Yihao Qian:

Yihao has been working with me and Menghan on the system integration. He helped with generating the camera information file of the image capturing module and have been working on migrating the tracking module to ROS and improving the current stereo vision module.

Menghan Zhang:

Menghan has finished building the detection and classification module for our system in ROS after setting up the environment (CUDA, Caffe, etc.) on our new desktop. She has also been working on building our new depth measure module, which is still in progress.

Plans

Before the next ILR/ progress review, our team would like to achieve the following goals on our project of developing the perception system using Stereo Vision and radar:

1. Overlay of radar and stereo vision data
2. Completed system architecture in ROS
3. An initial (working) integrated system as a rehearsal for the SVE

As a minimum goal, the three modules that are currently in progress (depth measure, radar, and GPS) and the two new modules (tracking and fusion) need to be all completed by the next progress review. The proper communications among them need to be also set up successfully by then. Depending on the overall progress, we will decide which visualization method to use eventually as the next step.

In terms of individual work, I will first finish my current work on the GPS module. I also plan to help with the integration of the tracking module in ROS. I plan to work together with the rest of the team to get all other scheduled work for system integration done by the next progress review.