Setup Foundation for the Husky

Marisa Warner, Dr. Gabriel Elkaim, Dr. Jeremy Gottlieb  
Worcester Polytechnic Institute; University of California, Santa Cruz

Introduction
The purpose of this project was to be able to build a foundation and documentation for the recently acquired Husky A200 robotic vehicle from Clearpath Robotics. What I accomplished was figuring out how to set up the Point Cloud Library on the Kinect and being able to stream different filters and perfect each setting, set up the settings for the recently acquired Husky A200 robotic vehicle from Clearpath Robotics. What I accomplished developing the code for autonomous drive both in the simulator Gazebo and on the Husky.

Kalibr

ROS

Kinect

LiDAR

Stereo Cameras

Gazebo

Introduction
The Robot Operating System is used to control and manage sensors and actuators; it uses the idea of publishers and subscribers. For example, Node 3 can publish a topic to Node 2 that was subscribed to. Other than controlling and managing sensor feeds, ROS also can record the topics and store the to be played back at a later time. It also provides a visualizer, Rviz, to be able to view cameras, point clouds, laser data, mapping, and simulate model robots.

Kinect

The Kinect is a RGB camera, depth sensor, microphone device developed by Microsoft for gaming consoles and computer stations. On the Husky, I used the RGB camera and the depth perception camera.

Stereo Cameras

Stereo Camera is two or more images from cameras put together to create a 3 dimensional view.

OpenCV

OpenCV is a program that will be used to calibrate the cameras taking both pictures and meshing the together to create wider and in depth picture.

Future Work
There are several more steps to completely the setup for the Husky. After the full calibration of the stereo cameras, the MIDS and the GPS will be installed with further documentation. Also, Gmapping would provide an excellent future for future projects along with a full 3D image of the LiDAR scans in Rviz. Lastly, full documentation including pictures and a sample code bank for future base projects for the Autonomous Systems Lab.

Acknowledgements
I want to thank the Autonomous Systems Lab for the space and providing the Husky. I also want to thank Dr. Elkaim and Dr. Gottlieb for the opportunity to work on the project and SURF-IT for selecting me as part of their research undergraduates.

Calibration of Kinect, LiDAR and Cameras
Kinect

Rviz is an add on to ROS, give the ability to view and manipulate different video feeds.

To view through the Kinect, I had to set up the Point Cloud on Rviz and be able to select through the different filters such as optical depth and RGB to able to come up with a clear and functional image. With this set up I was able to select the topics that will successfully be record video streams to be able to sample and play back later.

To view the Point Cloud, it had to calculate the different registered points. The points ranged from billboards (surface area covered with squares to real life points).

LiDAR

The LiDAR measures distance with a laser it is a mapping system that helps with navigation and creating a perimeter visual. I helped set up how to access this information within Rviz. This LiDAR can also be viewed through Rviz.

Stereo Camera

The stereo cameras provided true color images from in front of the husky.

Another set of eyes allows the use of motion sensing for the Husky and a 3D vision.

OpenCV is a program that will be used to calibrate the cameras taking both pictures and meshing the together to create wider and in depth picture.

Future Work
There are several more steps to completely the setup for the Husky. After the full calibration of the stereo cameras, the MIDS and the GPS will be installed with further documentation. Also, Gmapping would provide an excellent future for future projects along with a full 3D image of the LiDAR scans in Rviz. Lastly, full documentation including pictures and a sample code bank for future base projects for the Autonomous Systems Lab.

Acknowledgements
I want to thank the Autonomous Systems Lab for the space and providing the Husky. I also want to thank Dr. Elkaim and Dr. Gottlieb for the opportunity to work on the project and SURF-IT for selecting me as part of their research undergraduates.

---

Gazebo is a simulator that allows the user to test and control objects in a simulated world. One of the primary uses of Gazebo is to be able to test code on the simulated Husky before using it in real-time. It helps debug code without even having the Husky.

Another great thing about Gazebo is how it is integrated with Ros. You can have the Husky move in the environment while interacting with the other sensors such as the Kinect and LiDAR.

There are two ways to be able to test the code. The first way is to create a ROS Bag. A ROS Bag is a file that records the selected topics. Then replay the Bag while in Gazebo. To make sure the Bag plays, sometimes you will need to kill the node you are trying to emulate. It can be seen with recorded video streams.

Another way to test the code is to write a node. In this project I learned how to code in python to write executable nodes. A node is a file that can either publish topics or subscribe to topics. To make a Node move the Husky, I wrote a node that publishes the topic cmd_vel to Gazebo in either a loop or in a set of commands.

From this node that simulates the Husky in Gazebo, it is easy to transfer it onto the Husky replacing the topics with /husky/cmd_vel.