

MRSD Project

Tiffany May

Team G / Robographer

Teammates: Gauri Gandhi, Jimit Gandhi, Sida

Wang, Rohit Dashrathi

ILR #6

January 27, 2016

Individual progress

Improve the algorithm of pan tilt unit to remove vibration of servo motors

The original algorithm for human face following is very unstable. If the target point, nose, which is used to represent the whole face, is around the center, the pan servo motor will keep moving forward and backward. Several attempts have been tried, and the reason was finally spotted: The original algorithm is trying to maintain the nose at the central point, but the servo motor moves one degree each time. One degree could be too much and lose the target again, then the motor would move backward but still lose it again. That's why the servos keep vibrating all the time. The way to solve the problem is to put a rectangle inside the proportional tracker.

Here we will go through the simplified idea of proportional tracker to make this statement intelligible. First we catch the whole image from Intraface, and we transfer the pixels to according servo motor angles to keep tracking the face. According to experiment and monitoring, the result is pan angle from 55 to 125 degree with X pixel from 0 to 700 and tilt angle from 0 degree to 25 with Y pixel from 0 to 450. The outcome is distorted, though, during the transformation from laptop to Arduino board. We even get minus 50 Y pixel value from the Arduino monitor. An algorithm similar to Celsius-Fahrenheit transformation solved this problem, and a low-pass filter was also used to smooth the noises generated in the transmission from Intraface to testing board.

Back to the idea of rectangle, which might be easier to understand now that the rectangle was put to define an acceptable area. The acceptable area is an 50X50 pixel rectangle at the center. While the nose is in this center, the proportional tracker for following the nose wouldn't work. But once the nose leaves this area, the proportional sub function will kick in and start tracking, thereby significantly reducing the vibration from servo motors.

Connect Arduino to ROS

This week's team main task is to integrate all the sub system on ROS. My part is connecting Arduino to ROS and create our own package. After it's done, the Arduino IDE is not required anymore.

1. Learning how to publish and subscribe using Arduino IDE

Here I use `rosserial` package, which provides a ROS communication protocol that works over Arduino's UART. Installing this package would help you learn how to setting Arduino environment with `ros_lib`. If the setting up is successful, the Arduino IDE will

contain the new library as shown in Fig1.

2. Create our own arduino package

Rosserial package provides a convenient method of connecting laptop and USB serial link. But creating our own package would save us from using Arduino IDE all the time and make Arduino code work as a node like all the other sub system in our team. Use `catkin_create_pkg` to create new package and setting two different layer of cmake lists. The second one makes sure we can still use all the library from Arduino official site. The `generate_arduino_firmware` function we used to source the Arduino libraries is provided by the `arduino-cmake` toolchain. Though many issues regarding to environment setting happened, the Arduino code could be still applied. Since Arduino code could be considered as a `.cpp` file, which saved us a lot of time. Because the Arduino pan tilt tracking could be not demonstrated by a picture. Here we provide a picture showing the laptop is subscribing the information published by Arduino board, the classic hello world testing code in Fig2.

Challenges

Error messages received and bugs during connect Arduino to ROS

1. `avrdude: stk500_paged_write(): (a) protocol error, expect=0x14, resp=0x14`
Arduino 1.0.1 has a new version of ArduinoISP. Upgrading everything to Arduino 1.0.1 would work.
2. The specified base path `"/home/tiffany/catkin_ws/src/ROS-semi-official/src/helloworld"` contains a package but `"catkin_make"` must be invoked in the root of workspace

Deleted workspace and tried again, then it worked.

`Catkin_make` uses a flawed heuristics relying on the default folder layout to decide whether the user managed to invoke `catkin_make` in the correct folder.

3. `rosserial_arduino` fails for `.ino`, works for `.cpp`

the message: `make: *** [mechbot_v3_firmware_code-upload] Error 2 Invoking "make" failed`

Changing the entry to `generate_arduino_firmware` (program SKETCH file.ino SRCS `${ROS_LIB_DIR}/time.cpp BOARD uno PORT /dev/ttyACM0`) fixed the issue and allows us to use a `.ino` symbolic link

4. Both pan and tilt servo motors almost broke because of overusing and collision.

Teamwork

Jimt:

Implement photo clicking function. He wrote a piece of code to capture a image from intraface using `opencv` bridge function. When the person of interested represented by april tag reach within a certain distance, which is when 1 meter, the april tag

searching function would be hold. And the intraface will start to work. If the person smiles for 2 seconds, the photo clicking function would kick in and get the photo. He said the biggest issue encountered is that we can't store image to a random address. Need to use a pointer to prevent relevant errors.

Gauri:

Integration of nodes on single robot.

She organized all the nodes and make them work together, including april tag detection and intraface. She merge the mechanical and detection subsystem with the planning and navigation subsystem for a single turtlebot.

Sida:

Upgrade the turtlebot workstation computers to ROS Indigo and Ubuntu 14.04. She also contacted with the developer of intraface and get the updated ROS package version of intraface. She made a node to connect intraface and other subsystem.

Rohit:

Doing trade study for the laptops and auto calibration camera we are going to use. Taking into accounts of prices, weight and many other factors, he recommend chrome book, mainly because of its compatibility with Ubuntu.

Figures

Figure 1

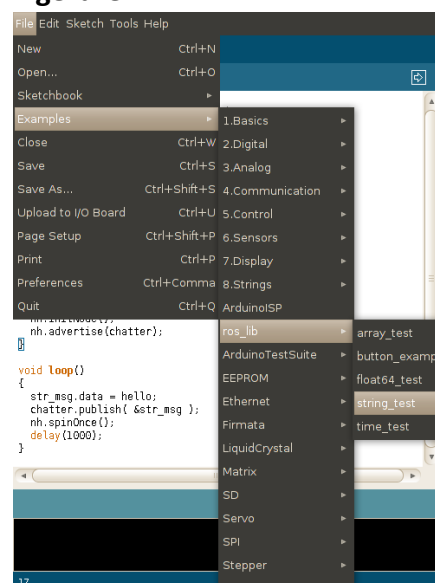


Figure 2

[illegible]

Future Plan

team

1. Each team member is going to learn gazebo and then we can stimulate turtlebot movement on computer. Starting and shutting down turtlebots each time we need to test a code took a significant amount of time.
2. Operating swarm and communication function on the ROSCON.

personal

1. Replace servo motors with new ones
2. Learning Gazebo as mentioned
3. Learning how to use ROSCON