

Progress review #7

Gauri Gandhi

Team G Robographers

Teammates:

Rohit Dashrathi

Jimit Gandhi

Tiffany May

Sida Wang

ILR #6

January 28, 2016

a. Individual Progress

For the seventh progress review, I first worked on upgrading the turtlebot workstation computer, i.e., my laptop to Ubuntu 14.04 and ROS Indigo to ensure its compatibility with the Intraface software. Then I worked on integrating the navigation sub-system and the face detection sub-system.

I. Upgrading everything to Ubuntu 14.04 and ROS Indigo

Until the fall semester, half of our sub-systems were working on Ubuntu 14.04 equipped with the ROS Indigo and the rest half on Ubuntu 12.04 and the ROS Hydro. The reason behind this was that the turtlebot netbooks were old and required the older versions whereas the latest version of the Intraface software provided to our team worked on the latest versions. So, our major goal this time was to upgrade the operating system and the software on all the workstations. As right now we are working on a single turtlebot, I upgraded my laptop as everything was required to be integrated on that. This process took me a lot of time as there were a lot of broken files and directories in my previously existing Ubuntu version that were hindering the upgradation process. So, I decided to delete the complete Ubuntu partition from my dual boot system and install the newer version by creating a new partition. That worked for me. After that I installed the ROS Indigo and all the required packages. Then, I installed the Intraface software with the renewed license on the laptop.

Once I was done with setting up my computer, I decided to set up the turtlebot settings and the network configuration. For the turtlebot installation, I had to export the following environment variables for the in my .bashrc file.

```
export TURTLEBOT_BASE=create
export TURTLEBOT_STACKS=circles
export TURTLEBOT_3D_SENSOR=kinect
export TURTLEBOT_SERIAL_PORT=/dev/ttyUSB0
```

After this I did the bringup of the turtlebot using the following roslaunch command.

```
roslaunch turtlebot_bringup minimal.launch
```

Hence a proper communication was setup between the turtlebot and the workstation.

Although, right now the laptop itself is the host computer for the turtlebot but once we start working on the swarm, a proper network setup would be required to achieve multi-master communication. So I did an initial ssh setup on the laptop to facilitate the process.

II. Integration

The major programming task for me this time was to integrate the navigation and the face detection sub-system. The goal was to launch the Intraface node only after the robot has reached at the 100 centimeters distance from the person of interest. ROS does not allow the sequential launching of nodes. So first I tried to put some conditions in the launch file based on the arguments passed using the if and until clauses. But that didn't work. So I decided to do some changes in the participating nodes themselves.

Firstly, I created both run and build dependencies of the intraface package on the robographers package that contained the april_tag_detection, navigation and rotation nodes in the xml file of the intraface package. Then, in the main Intraface node, I created a global Boolean variable `g` and used it as a flag in the code. I made the intraface node subscribe to both the sensor messages from camera video feed and the april tag list messages from the running april tag detection node. In the callback function for the april tag subscriber, I set the `g` flag to true if the april tag detected distance from the camera is less than 100 cms and false otherwise. Then I used this `g` flag in my main function by publishing on the advertised topics only when the value of `g` was set to true. All these steps resulted in giving me the desired results of launching the intraface only when the distance between the human and the camera was less than 100 cms. This is shown in fig 1.

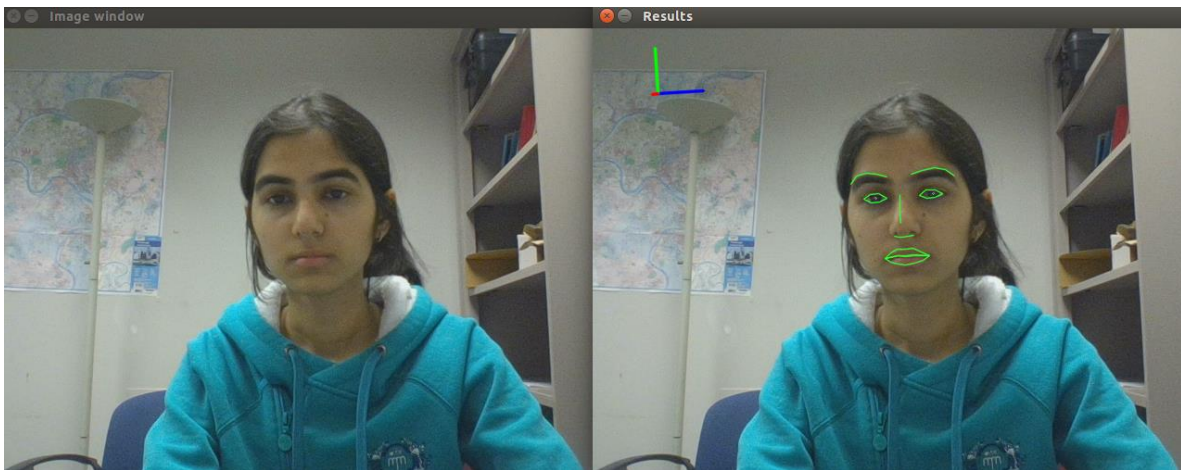


Figure 1 Intraface launched after apriltag distance is less than 100cms

After achieving all this, I launched all the subsequent nodes of photo clicking, smile percentage detection and nose coordinate detection for running the Arduino node. The rqt_graph for the whole process is shown in figure 2. The

rostopic list in figure 3 also shows that all the nodes are running and publishing to different topics.

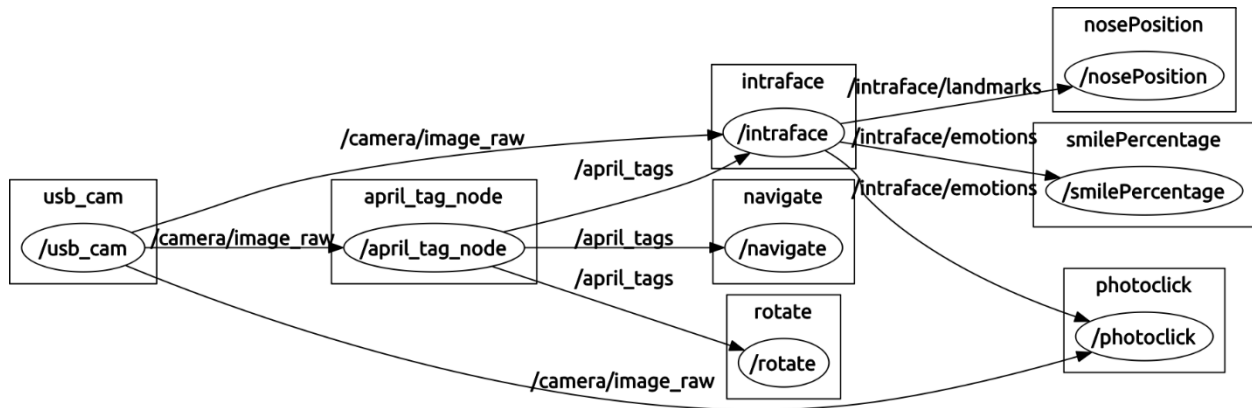


Figure 2 rqt_graph

```

gauri@gauri:~$ rostopic list
/april_tag_debug/output_video
/april_tag_debug/output_video/compressed
/april_tag_debug/output_video/compressed/parameter_descriptions
/april_tag_debug/output_video/compressed/parameter_updates
/april_tag_debug/output_video/compressedDepth
/april_tag_debug/output_video/compressedDepth/parameter_descriptions
/april_tag_debug/output_video/compressedDepth/parameter_updates
/april_tag_debug/output_video/theora
/april_tag_debug/output_video/theora/parameter_descriptions
/april_tag_debug/output_video/theora/parameter_updates
/april_tags
/camera/camera_info
/camera/image_raw
/camera/image_raw/compressed
/camera/image_raw/compressed/parameter_descriptions
/camera/image_raw/compressed/parameter_updates
/camera/image_raw/compressedDepth
/camera/image_raw/compressedDepth/parameter_descriptions
/camera/image_raw/compressedDepth/parameter_updates
/camera/image_raw/theora
/camera/image_raw/theora/parameter_descriptions
/camera/image_raw/theora/parameter_updates
/intraface/emotions
/intraface/irirgaze
/intraface/landmarks
/intraface/parameter_descriptions
/intraface/parameter_updates
/intraface/rotation
/mobile_base/commands/velocity
/rosout
/rosout_agg

```

Figure 3 rostopic list

b. Challenges

- I. The first challenge I faced this time was to upgrade the Ubuntu version 12.04 to 14.04. I had to encounter a lot of issue in this process. Even the grub loader for

the dual boot was showing some problems. So I took the backup from both the Ubuntu and the Windows and deleted the Ubuntu partition to install it again.

- II. For integration, first I decided not to alter the codes in the nodes and rather change the launch file. But that didn't work. So I decided to create some new node that would help me achieve integration but that also didn't work. Ultimately I had to alter the Intraface node. But all these steps took many hours of trial and error.

c. Team Work

For this progress review, Rohit worked on doing the trade study for the laptops and the cameras. He did an exhaustive study on both and finally ordered the components. Sida was successful getting the nose and smile estimate from the Intraface. She was able to send the position coordinates of the nose from the detected face to the serial port. Tiffany set up the connection between the obtained parameters from Sida's part and the Arduino to run the pan tilt servo motors. This time she did this complete task on ROS. Jimit created a ROS node to autonomously click a colored photo from the incoming video stream as soon as the person smiles. All of us then worked together to integrate the whole system.

d. Future Plans

For the next Progress Review, I will be working on the following tasks:

- I. Running all the nodes in a loop so that not just one person is detected or clicked.
- II. Studying and setting up ROCON for swarm communication.
- III. Updating the team website.

e. References

1. Turtlebot ROS tutorials

wiki.ros.org

2. Stack overflow for clearing the ROS related doubts

<http://stackoverflow.com/>

3. Cpp tutorials

<http://www.tutorialspoint.com/cplusplus/>