



Team G- Robographers

Progress Review 7

Individual Lab Report 6

Jimit Gandhi

Team Members

Rohit Dashrathi

Sida Wang

Gauri Gandhi

Tiffany May

1.0 Introduction

This week the team focused on integrating and fulfilling the requirements of photo-clicking single robot system.

2.0 Individual Progress and Contribution.

The goals for our Progress Review 2 were as follows

1. Upgrade the turtlebot workstation computers to ROS Indigo and Ubuntu 14.04.
2. Merge the mechanical and detection subsystem with the planning and navigation subsystem for a single turtlebot.
3. Implement the photo clicking algorithm.
4. Finalize and order the required parts for the project.

My major work was in implementing the photo-clicking algorithm and help Gauri in merging the subsystems. In order to implement the photo-clicking algorithm, I had to learn the Open CV libraries as there was no direct C++ based function or library available which would convert the sensor_msgs image files to jpg compressed images. Sensor_msgs is a format that is used by ROS package called image_transport for transmitting image or live feed from video camera/ web camera to different nodes subscribing to the ROS topic named sensor_msgs/raw_image_camera/. This topic basically transmits image in the form of an array of the size of number of pixels in the image i.e. each image array equal to height* width of the image in uint8 format. To convert this in to a matrix of image with rows and columns equal to the width and height of the image respectively, Open CV provides a function toCVCopy () especially for this purpose.

The sensor_msgs are basically encoded in a compressed format by image_transport node which transmits the data along with channel meaning, ordering and size of the image as separate strings. OpenCV's toCVCopy () and toCVShare () both do similar functions of using the above information and array data and convert them to a mat format which is a matrix of the size of the image in BGR8 format. I basically created a node called "smile-detector&photo-clicker.cpp" which would subscribe to both Intraface's /emotion topic as well as the camera/raw_image topic.

The goal was to receive the video feed frame by frame and the emotion reading at the same time. This was another challenge that I faced. Luckily, on some research of ROS, I came upon a

function called `boost::bind` in which ROS master would synchronize and publish the two or more topics at the very same time using their Time-Stamp and Sequence number.

Once the node started receiving the topics in a synchronous manner, the photo-clicking implementation became straight forward. Intraface's emotion topic is basically a six float number array in which each location of the array contains the six probability values of basic human emotion based on the expression and they are – Sad, Disgust, Happy, Anger, Neutral and Surprise as shown in figure 1.

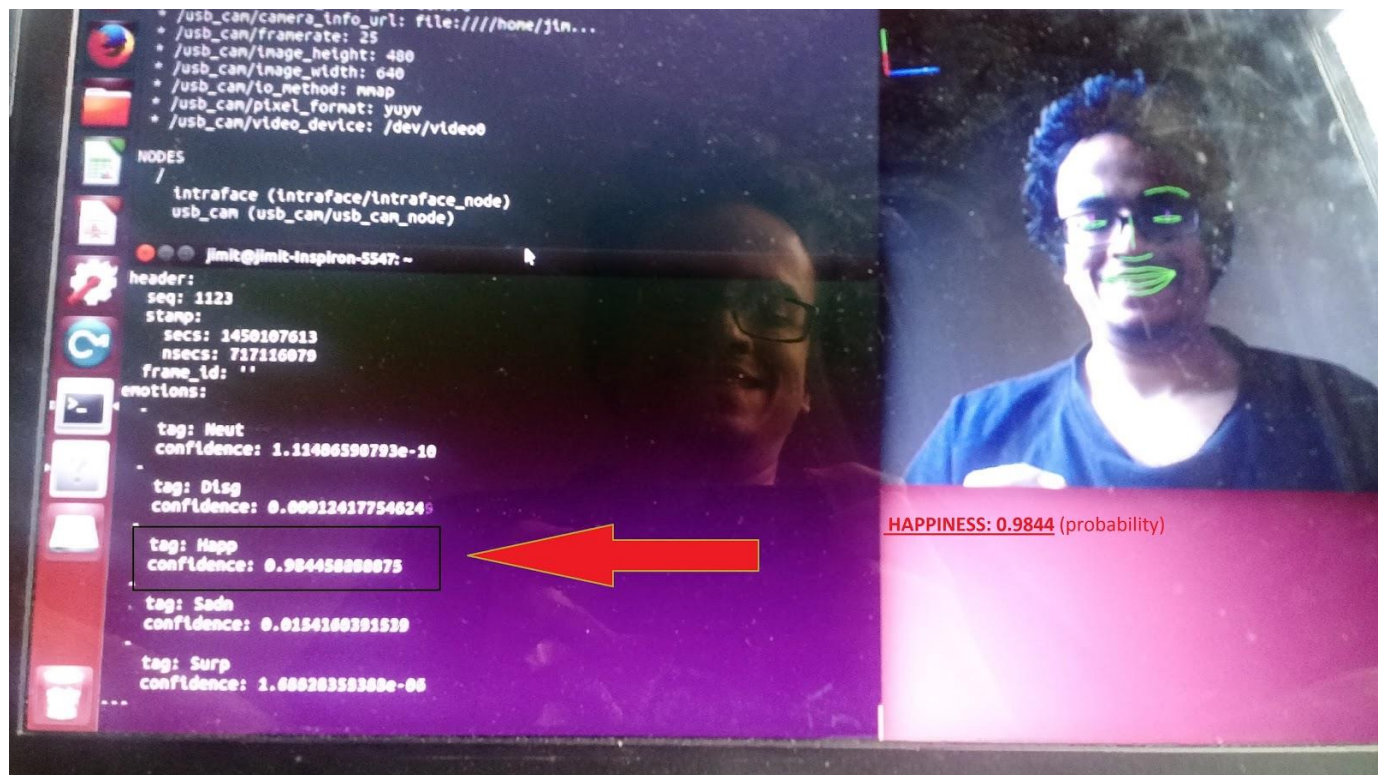


Figure 1: Intraface displaying real time analysis of face expressions. It identifies 6 basic expressions in the order Neutral, Disgust, Happy, Sadness and Surprise giving their probability values.

I basically extracted the third element of the array which corresponds to the probability value of Happy expression. As Intraface is a very robust and one of the best expression recognition software, it gives out high probability values for a very explicit emotion which is what we wanted to capture. The goal of project is to capture obvious smiling faces and Intraface does a great job in recognizing them giving probability values as high as 0.99996 for a perfectly smiling face. Now to improve the stability and assure that the person is indeed smiling, I kept a check if the person smiles for a duration of 3 seconds after which only photo will be clicked. This ensures good quality of photo. Other features that made the photo aesthetic is to keep the face in the center of the frame using the face tracking using pan tilt algorithm we created during fall validation. The final output of photo-clicking node is shown in figure 2.

Apart from the above I also helped Gauri in integrating the subsystem and also helped Tiffany and Gauri install Intraface on their laptops so they can work on their subsystems and carry out various tests.

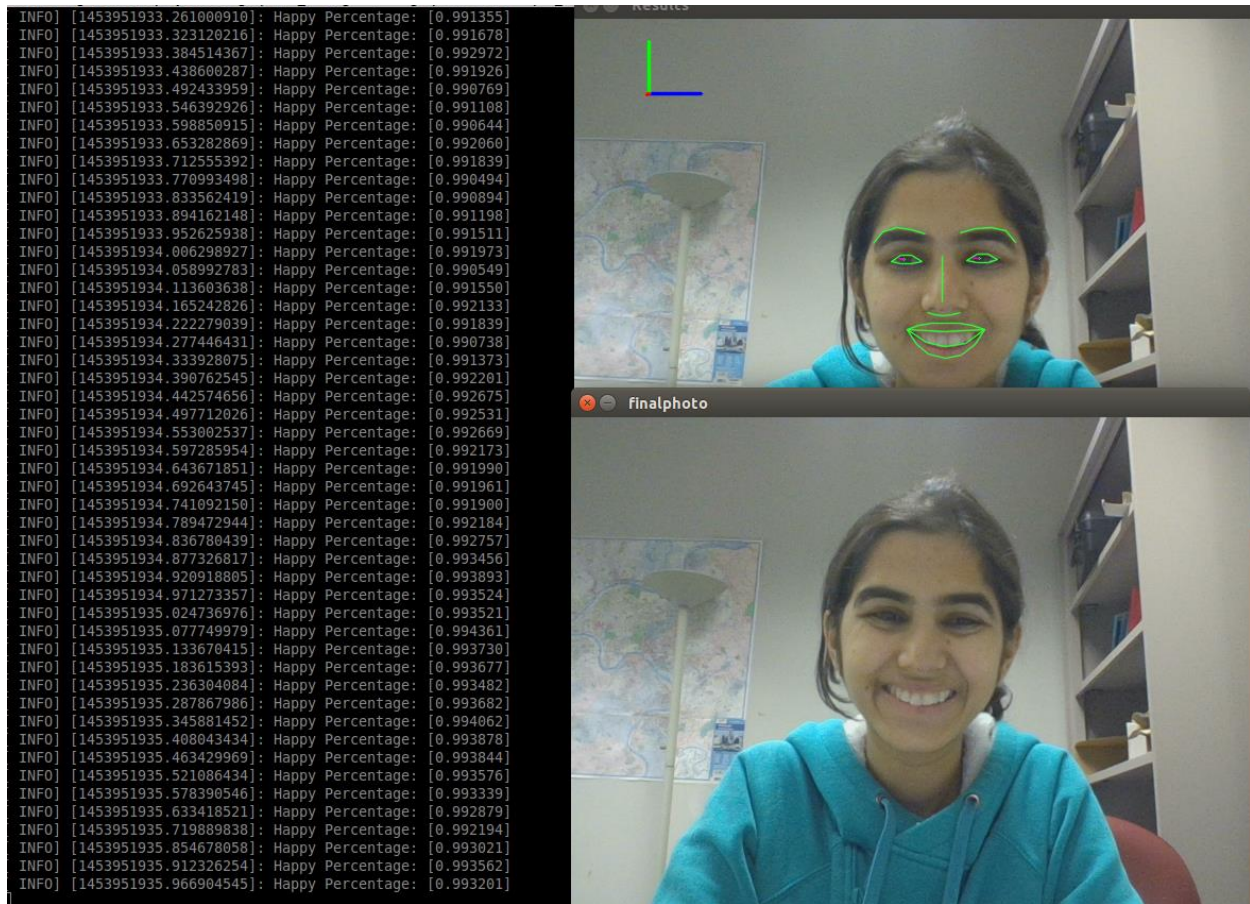


Figure 2: On the left the probability of person being happy is being printed out. Right top is Intraface working real time on person's face. Right bottom is the final photo which was stored on computer desktop folder as finalphoto.jpg.

2.1 Challenges Faced

The challenges that I faced was difficult to point out. When I compiled the photo-clicking node successfully and ran it, the node launched and crashed instantly giving me an error called "Segmentation fault. Core dumped". Now in my experience I had an idea that this error is given when you have an uninitialized pointer or when a pointer tries to access the memory of the operating system which it is not allowed to or when the pointer points to the non-existing member of the array. But unfortunately I also was aware that these errors are one of the most difficult and take excruciating debugging skills. However I went on to debug it as that was the only viable option left. After few hours of debugging, I finally realized that it was due to the /camera_raw_image subscriber. The pointer would point to the incoming sensor_msgs image data but somehow it is the ROS internal mechanism that the first few messages are not

received instantly after the launch of the node. Hence initially the program crashed because there was a Null Pointer being formed. I fixed this bug by initializing the pointer only if the data it receives is not null but an actual image data.

The other challenge I faced was synchronizing the messages of emotions and video frames. For this I had to go in depth and study how ROS communication works and how `boost::bind` helps to solve that. Basically ROS master receives a data from one topic and notes its time stamp. Then it keeps this data on hold until another topic publishes the message with the same time stamp. Only after this does the ROS master publish both of them for subscribers to receive. This was a less of a challenge and required more of study of ROS and its working.

The other challenge I faced is when Gauri was working on integrating the navigation and smile detection subsystem. We discussed various ways in which we could integrate the system. Finally after much careful analysis and trial by errors I and Gauri introduced a flag variable such that when the navigation subsystem gets the robot to 1 meter from human the flag is switched to true and this is when Intraface starts publishing the emotion topics and gets photo clicked by the photo-clicking node. This worked best for us as we did not want to interfere or tightly couple the navigation and Intraface nodes as it could lead to complications in a more complex system.

3.0 Team Work

Rohit worked on trade study of camera and new laptops we have to buy. He made in depth analysis and kept important things in mind such as compatibility with Intraface and Ubuntu of the new laptops, price, size which was also important as they will be mounted on the Turtlebot and space required by laptop should be less than 12" screen laptop. He is also learning Gazebo on ROS as we will require it for simulation of swarm robots. Gauri worked mostly on upgrading the Turtlebot laptop to Ubuntu 14.04 and ROS Hydro after which she integrated the subsystems of navigation and photo clicking. Sida worked on creating a node that displays the Intraface reading of Happy emotion while Tiffany worked on integrating pan tilt subsystem with the entire robot system. They also spent quite a while learning ROS as they are new to it and were focusing on different things last semester.

3.1 Future Work

We plan to move as per our work breakdown structure. Next week's agenda includes the following.

1. Implement swarm by making robots move from one location to another using Flocking algorithm

2. Get the new laptop and camera and test them out for their performance before ordering the same for other Turtlebots.
3. Set up communication among the Turtlebots and also set up ROCON gateway such that these Turtlebots could communicate.

4.0 References

1. Intraface Software - Fernando De la Torret[†], Wen-Sheng Chu[†], Xuehan Xiong[†], Francisco Vicente[†], Xiaoyu Ding[†], Jeffrey Cohn^{†‡} [†]Robotics Institute, Carnegie Mellon University, (<http://www.humansensing.cs.cmu.edu/intraface/>)
2. A gentle Introduction to ROS <https://cse.sc.edu/~jokane/agitr/>
3. www.stack-over-flow.org
4. www.ros.org
5. www.opencv-tutorials.org