MRSD Individual Lab Report 10

Name: Sida Wang

Group Name: Group G, RoboGrapher

Teammates: Tiffany May, Rohit Dashrathy,
Jimit Gandhi, Gauri Gandhi

March 30, 2016

1. Individual Progress

For this week's progress review, my individual progress can be divided into 3 parts:
(a). Realized voice command functionality in our project
(b). Integrated of detection subsystem: Intraface + extracting region of interest + voice command
(c). Connected Intraface and April tag cordinate value with Arduino

1.1 Implemented voice command functionality

(a). Pipeline design

The general idea of voice command functionality is that after Intraface has been launched, if the person does not smile for some time(for example, 5 seconds), then the master computer will send a voice signal “say cheese” to the person. And this process will loop if the person sill does not smile. In this way, the system will send “say cheese” to the person every 5 seconds until he or she smiles. After the person smiles for 2 seconds, the system will send another voice command “nice smile” to the person, and also smiling photo will be taken.

(b). ROS package: sound_play

ROS has its internal package for voice signal called <sound_play> package. The `sound_play` node considers each sound (built-in, wave file or synthesized text) as an entity that can be playing, playing repeatedly or stopped. Nodes change the state of a sound by publishing to the `robotsound` topic. Multiple sounds can be played at once.

Thanks to the built-in package, I don't have to setup everything by myself. Instead, I can just learn to install the package and study how to use the user interface.

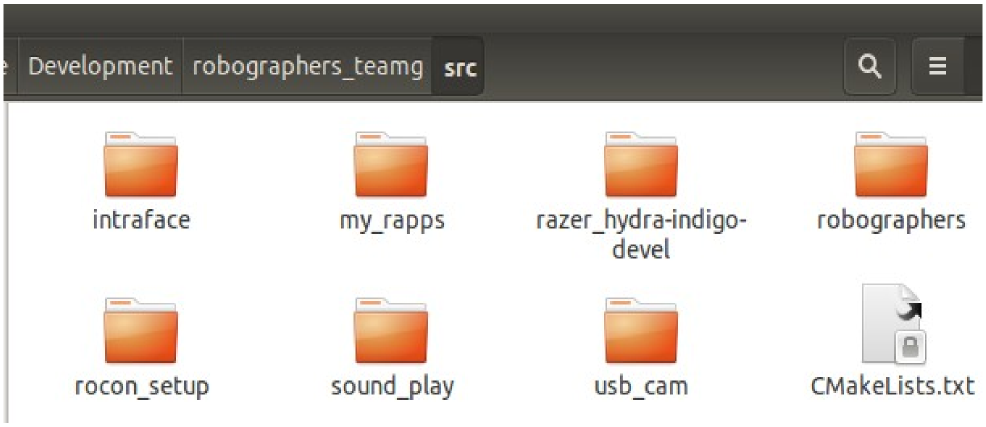


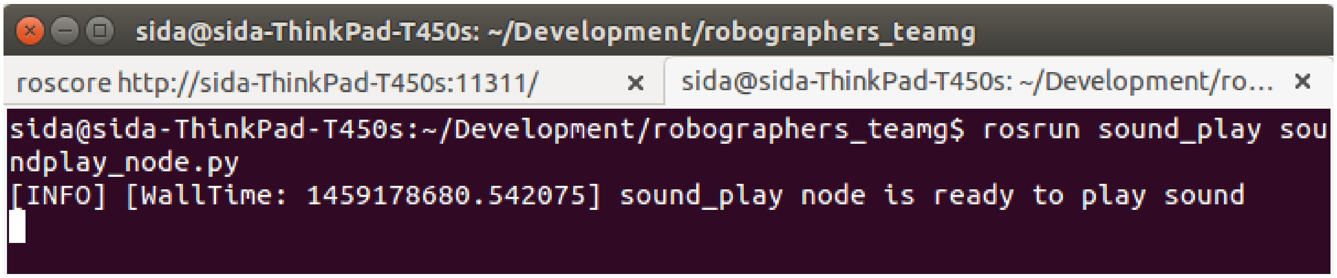
Figure 1: sound_play installation

(c). <sound_play> installation and setup the environment

First, I downloaded the package from github provided by ROS wiki: Source: git https://github.com/ros-drivers/audio_common.git (branch: hydro-devel). As shown in Figure. 1, I put the <sound_play> package under my mrsd project src package.

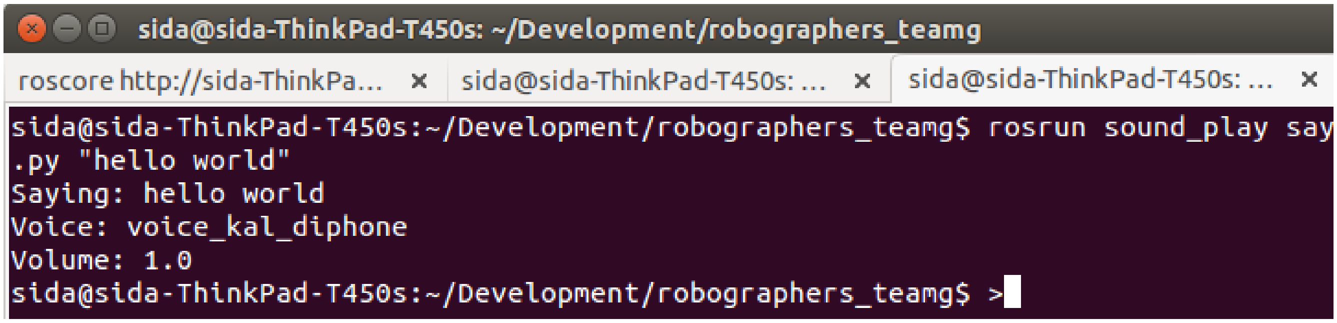
Second, in order to build the whole project ROS package, I catkin_make the package.

Third, I referred to the sound_play tutorial, and followed the instructions. I ran “roscore” in the first terminal. In the second terminal, “rosrun sound_play soundplay_node.py”. And “sound_play node is ready to play sound” is shown in Figure 2.a. In the third terminal, “rosrun sound_play say.py "hello world"” is shown in Figure 2.b. And then a voice command “hello world” can be sent to the user.


sida@sida-ThinkPad-T450s: ~/Development/robographers_teamg
roscore http://sida-ThinkPad-T450s:11311/
sida@sida-ThinkPad-T450s: ~/Development/ro...

```
sida@sida-ThinkPad-T450s:~/Development/robographers_teamg$ rosrun sound_play sou
ndplay_node.py
[INFO] [WallTime: 1459178680.542075] sound_play node is ready to play sound
```

Figure 2.a: rosrun sound_play soundplay_node.py


sida@sida-ThinkPad-T450s: ~/Development/robographers_teamg
roscore http://sida-ThinkPa...
sida@sida-ThinkPad-T450s: ...
sida@sida-ThinkPad-T450s: ...

```
sida@sida-ThinkPad-T450s:~/Development/robographers_teamg$ rosrun sound_play say
.py "hello world"
Saying: hello world
Voice: voice_kal_diphone
Volume: 1.0
sida@sida-ThinkPad-T450s:~/Development/robographers_teamg$ >
```

Figure 2.b: rosrun sound_play say.py "hello world”

(c). Use <sound_play> package in our project

I modified the 'photoclick.cpp' code to add the functionality of voice command. First I include sound_play.h in the code. And then, I can use “sc.say("Say Cheese");” to send the voice command. And after the node has received the smiling percentage, which indicates that Intraface has launched.

And when Intraface has been launched, and count the time. If the person still does not smile for 5 seconds, then will run “sc.say("Say Cheese");” until the person smiles, then will run “sc.say("Nice smile");” to the person.

1.2 Integrated of detection subsystem: Intraface + extracting region of interest + voice command

In this week, we were trying to integrate detection part. I mainly worked on integrating Intraface with extracting region of interest with voice command, and Gauri mainly worked on integration on Rocon.

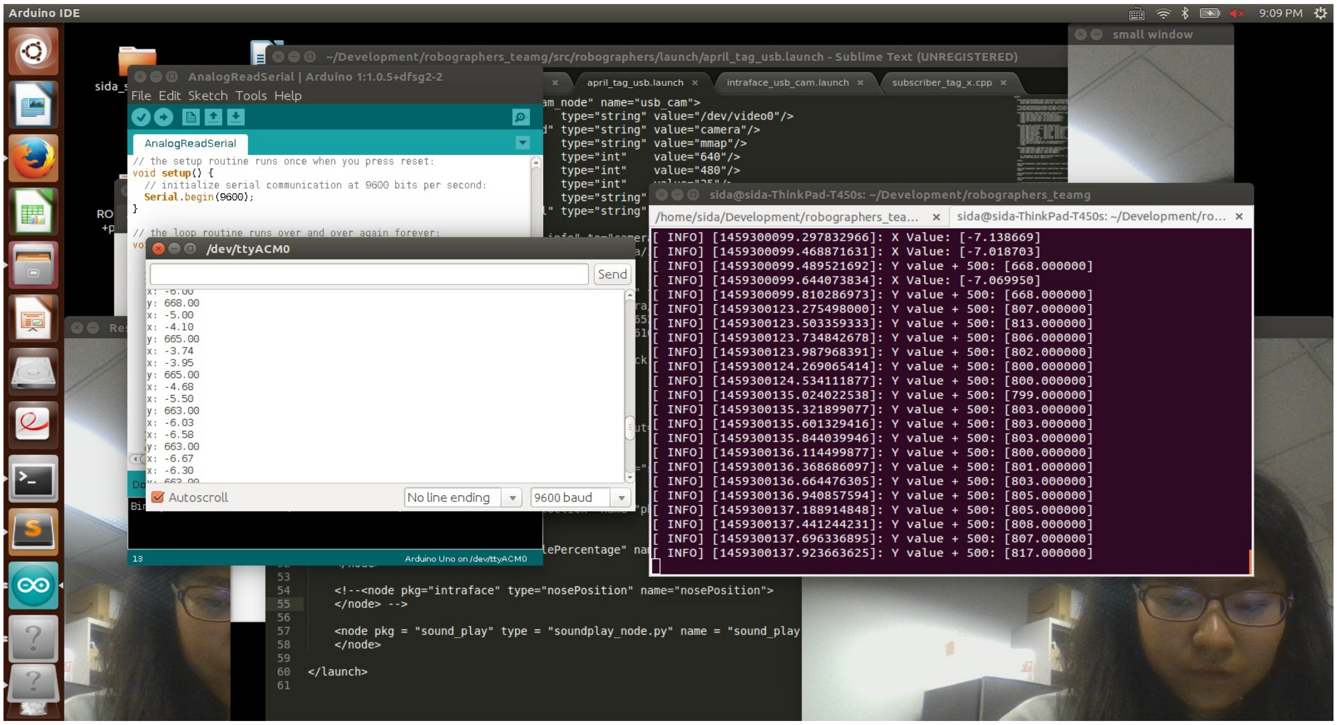
(a). Integrated Intraface and extracting region of interest

I changed the code
/home/sida/robographers_teamg/src/robographers/src/april_tag_node.cpp
for extracting region of interest: extracting only the person's face above April-tag, and publish the image massage on the topic called “small_frame”.
Then, I changed the code
/home/sida/robographers_teamg/src/intraface/src/ros/intraface_node.cpp
Originally, Intraface will subscribe from the webcam. And now, it should subscribe from the image message called “small_frame”.

(b). Integrated voice comman in our system

I helped Gauri in this part. What I did is to install <sound_play> package in her laptop and told her how to use my code. And she integrated into the whole system.

1.3 Connected Intraface and April tag cordinate value with Arduino


Arduino IDE
9:09 PM
small window
~/Development/robographers_teamg/src/robographers/launch/april_tag_usb.launch - Sublime Text (UNREGISTERED)
april_tag_usb.launch
intraface_usb_cam.launch
subscriber_tag_x.cpp
AnalogReadSerial | Arduino 1:1.0.5+dfsg2-2
File Edit Sketch Tools Help
AnalogReadSerial
/dev/ttyACM0
Send
Autoscroll
No line ending
9600 baud
13
Arduino Uno on /dev/ttyACM0
sida@sida-ThinkPad-T450s: ~/Development/robographers_teamg
/home/sida/Development/robographers_tea...
sida@sida-ThinkPad-T450s: ~/Development/ro...

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
```

```
x: -6.00
y: 668.00
x: -5.00
x: -4.10
y: 665.00
x: -3.74
x: -3.95
y: 665.00
x: -4.68
x: -5.50
y: 663.00
x: -6.03
x: -6.58
y: 663.00
x: -6.67
x: -6.30
y: 663.00
```

```
am_node" name="usb_cam">
    type="string" value="/dev/video0"/>
d" type="string" value="camera"/>
    type="string" value="mmap"/>
    type="int"    value="640"/>
    type="int"    value="480"/>
    type="int"
    type="string"
l" type="string"
```

```
    <!--<node pkg="intraface" type="nosePosition" name="nosePosition">
    </node> -->

    <node pkg = "sound_play" type = "soundplay_node.py" name = "sound_play
    </node>

</launch>
```

```
[ INFO] [1459300099.297832966]: X Value: [-7.138669]
[ INFO] [1459300099.468871631]: X Value: [-7.018703]
[ INFO] [1459300099.489521692]: Y value + 500: [668.000000]
[ INFO] [1459300099.644073834]: X Value: [-7.069950]
[ INFO] [1459300099.810286973]: Y value + 500: [668.000000]
[ INFO] [1459300123.275498000]: Y value + 500: [807.000000]
[ INFO] [1459300123.503359333]: Y value + 500: [813.000000]
[ INFO] [1459300123.734842678]: Y value + 500: [806.000000]
[ INFO] [1459300123.987968391]: Y value + 500: [802.000000]
[ INFO] [1459300124.269065414]: Y value + 500: [800.000000]
[ INFO] [1459300124.534111877]: Y value + 500: [800.000000]
[ INFO] [1459300135.024022538]: Y value + 500: [799.000000]
[ INFO] [1459300135.321899077]: Y value + 500: [803.000000]
[ INFO] [1459300135.601329416]: Y value + 500: [803.000000]
[ INFO] [1459300135.844039946]: Y value + 500: [803.000000]
[ INFO] [1459300136.114499877]: Y value + 500: [800.000000]
[ INFO] [1459300136.368686097]: Y value + 500: [801.000000]
[ INFO] [1459300136.664476305]: Y value + 500: [803.000000]
[ INFO] [1459300136.940857594]: Y value + 500: [805.000000]
[ INFO] [1459300137.188914848]: Y value + 500: [805.000000]
[ INFO] [1459300137.441244231]: Y value + 500: [808.000000]
[ INFO] [1459300137.696336895]: Y value + 500: [807.000000]
[ INFO] [1459300137.923663625]: Y value + 500: [817.000000]
```

Figure 3 Connection between Arduino, April tag, and Intraface

Figure 3 shows the connection between Arduino, April-tag and Intraface. Compared to our system last symester, we have made some changes in our pipeline. Before, we just used the x and y cordinate value given by Intraface. But as we only publish small frame of person's face to Intraface, the x value of the nose position in the small frame will not change if the person moves with his or her aptil-tag. But in y direction it remains Intraface's y value of the person's nose.

So I have to publish aptil-tag's x value and intraface's y value to Arduino. And as these 2 values are publishing together, and as ROS is multi-thread, the data transfer speed is different when they are publishing together. So it took me a relative long time to diffentiate x and y values. What I did is to add 500 to y value. And when we get the value, if the value is higher than 500, then we can get to know that it is y value. Otherwise it is x value.

In this way, we can use the x and y values that Arduino gets to control the motors for face tracking.

1.5 Setup new Chromebook with Gauri

I worked with Gauri to set up our new Chromebook. It's a little bit tricky to install Ubuntu on Chromebook. We asked our leading PHD student and also searched a lot of information on the Internet.

The order is shown as follow:

(a). Boot Chromebook into Developer Mode
(b). Enable booting from USB device
(c). Clear all partitions on /dev/sda
(d). Make a new swap as well as ext4 partition
(e). Setup the working environment for turtlebots
(d). Install everything we need for our project and setup robographers' ros package in the Chromebook

2. Challenges

2.1 Voice command feature

As we all do not have any experience on voice command, it was hard for me to explore this field. And I searched on the Internet for a long time, and there is limited resources about it. And then I found there is built-in “sound_play” package for this function. Then, I installed, studied, and integrated in our project.

2.2 Setup Ubuntu working environment for Chromebook

It took Gauri and I a relative long time to setup Ubuntu operating system on Chromebook. And we worked according to the leading PHD student's instruction in our lab and also searched on the Internet. And finally it works!

2.3 Nose position sent to Arduino

Since last PR we started to extract only the face of the person with April-tag and we published the image message to Intraface. And before that, Arduino would receive the relative cordinate value of the person's nose position to the image frame. However, as we use April-tag to detect the person's face, and publish the small frame only with the person's face to Arduino, so the cordinate value of the person's nose in the Intraface will not change when the person's face changes.

In order to solve this problem, we consider to send April-tag's x value to Arduino and y value of nose's position to Arduino. And face tracking sybsystem can be determined according these 2 values. This is the change of pipeline of our project. We truly discussed for several times and finally determined this scheme.

3. Team Work

This week, the main progress our team has made is shown as follows:

(a). (I myself) Voice command feature
(b). (Gauri, I) Integrated detection sybsystem with rocon and pan-tilt on turtlebots
(c). (Gauri, I) Setup new chromebook with all the detection subsystem
(d). (Tiffany, I) Connected Intraface with Arduino
(e). (Jimit, Gauri) Localization and flocking as a seperate subsystem
(f). (Rohit) Make the system more beautiful on mechanism part

4. Future Plan

For next PR, we are trying to show the 2 seperate subsystems: navigation subsystem and detection subsystem. Navigation subsystem is mainly flocking part of project, including self-collision avoidance and global map setup. As the detection system has almost been integrated, the whole team will focus more on flocking subsystem and the whole project integration.

Integration of detection substem has been almost done. April-tag detection, person with April-tag extraction, Intraface facial expression recognition, voice command, smiling percentage comparison between different master computers, as well as photo clicking, have already been done. The only thing left if pan-tilt unit.

What I will do:

(a). Detection subsystem integration.

I will use ROS command to run Arduino code for the purpose of face tracking subsystem. And put the node into launch file. So that after we launch the overall launch file, the Arduino will automatically start to run. And I will finish the whole process of integration of detection subsystem.

(b). Get involved in navigation subsystem.

In our project, detection subsystem has been nearly done. And after I have finished the whole detection subsystem integration, I will work on flocking part with my teammates.

5. References

(1). ROS package: sound_play for voice command
http://wiki.ros.org/sound_play

(2). Install Ubuntu on Chromebook
http://lifehacker.com/how-to-install-linux-on-a-chromebook-and-unlock-its-ful-509039343