

MRSD Project

Tiffany May

Team G / Robographer

Teammates: Sida Wang, Rohit Dashrathi, Gauri

Gandhi, Jimit Gandhi

ILR #10

March 31, 2016

Individual progress

Connect Intraface with Sida using a publisher

Sida published a topic from Intraface and April Tag, and I subscribed it. The Intraface provides the y coordinate of the nose position as usual. But the x coordinate is now provided by April Tag on the target person, the reason that we having xy coordinate from different sources is because the frame of single person. Intraface can only get a smaller rectangle frame that includes April Tag and the target person, other people will be excluded from the scene, x position of the nose became meaningless since the new shape of the rectangle frame is pretty narrow.

The issue is, we can only execute Intraface in one frame, hence we cannot have the original frame which includes the appropriate x coordinate and the new narrow rectangle frame which only includes the target person both being processed by Intraface at the same time. The first resort is using another tracking function to send back the x coordinate of the nose in the original frame. There's already a Lucas-Kanade tracker from the last semester, but the extra usage of the kernel made the processing speed which is already slow due to the multi command windows even slower. The lagging problem became a main issue. To deal with it, a backup idea was applied, which is using the x coordinate of April Tag to replace the x coordinate of the nose position. The x coordinate of the April Tag is coming from the original frame, and it's not processed by Intraface. Only the narrow frame with the target person needs to go through the Intraface. Hereby the problem that Intraface could only be executed in only one terminal at the same time is solved and the arduino board can get the appropriate x y coordinate of the human face.

Set a direction controller in Arduino code

The new pan-tilt unit's pan motor is upside down, but we still need the original upright pan-tilt unit in case of any malfunction of the servo motors, which is a problem happens more frequently than expected. Instead of adjusting all the motors one by one, a direction controller was used. Also, frequently changed parameters in the transferring function and other sub function will be take out, therefore the adjusting procedure will be faster and the function will be more adapted to possible future changes. The transferring function is applied because of the different metrics of cameras and the random distortion happened during the transferring of data from laptop to arduino board.

Try to set up flocking function

There's three area in flocking, attracting, aligning and repelling area. If the distance

between the turtlebot and the leader turtlebot exceed 3 feet, the turtlebot will try to get closer to the leader turtlebot. While the two turtlebots are in the aligning area, which is 1.25 to 3 feet, the linear and angular speed will remain the same. Once they are closer than 0.75 feet and reaching the repelling area, the turtlebot will move away from the leader turtlebots. There are some problems with the node between global camera and master computer so we cannot read the position from the turtlebots now. Hence the flocking algorithm hasn't been tested yet.

Challenges

2 blind spots in global cameras

As the original global camera is settled, the height of the April Tag wasn't taken into concern. So we have two blind spot of the global camera.

Because of the height of the turtlebots, there are two blind spots in three global cameras, in the following figures, you can see Jimit first shown in the Figure 1, but as the time and he moving toward, the April Tag he holds disappear in the blind spot in Figure 2 and then the tag shown again in the Figure 3.

We are planning to use visual odometry that will make turtlebots moving directly to the target person while the turtlebots is in the blind area.

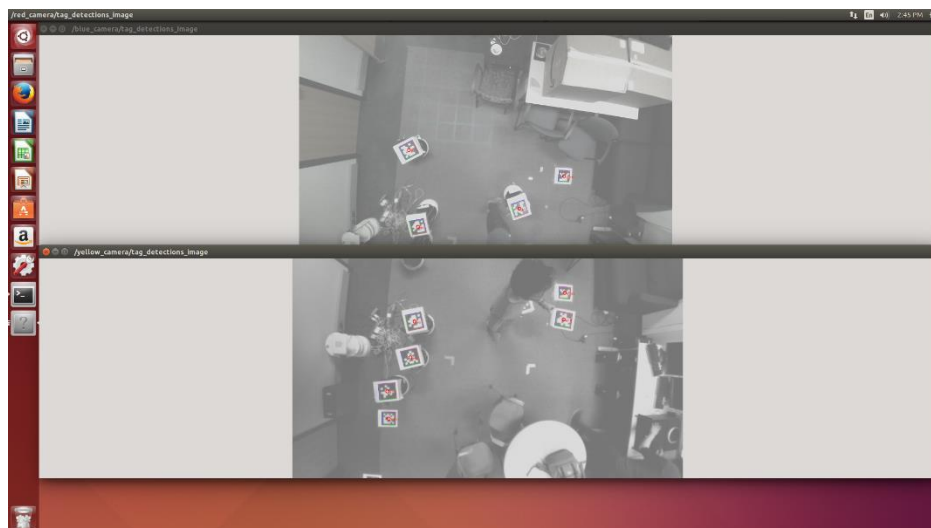


Figure 1

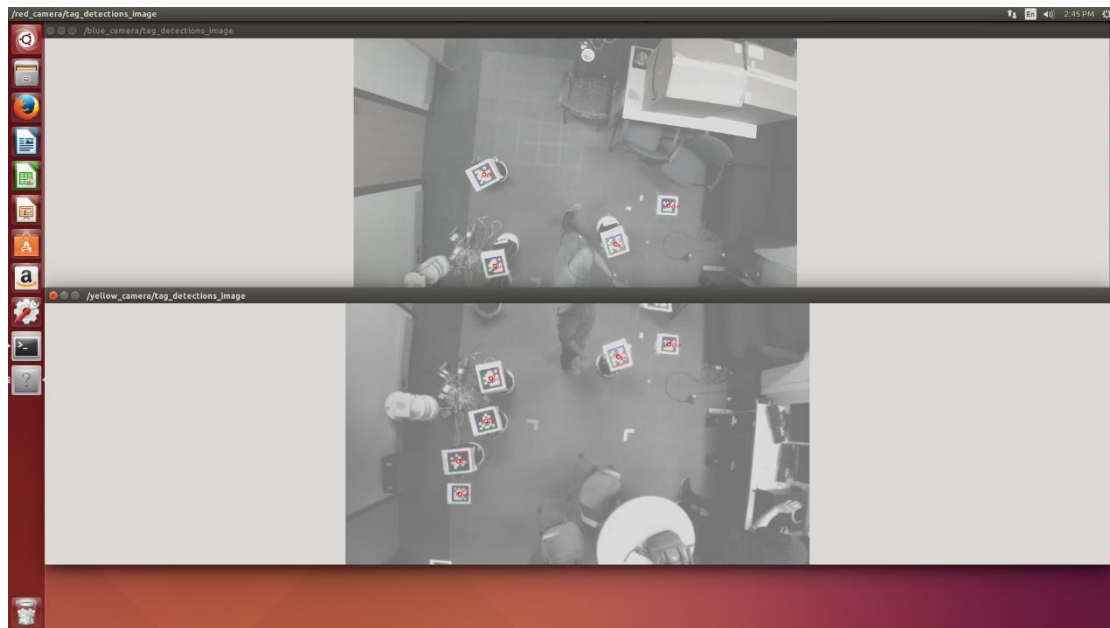


Figure 2

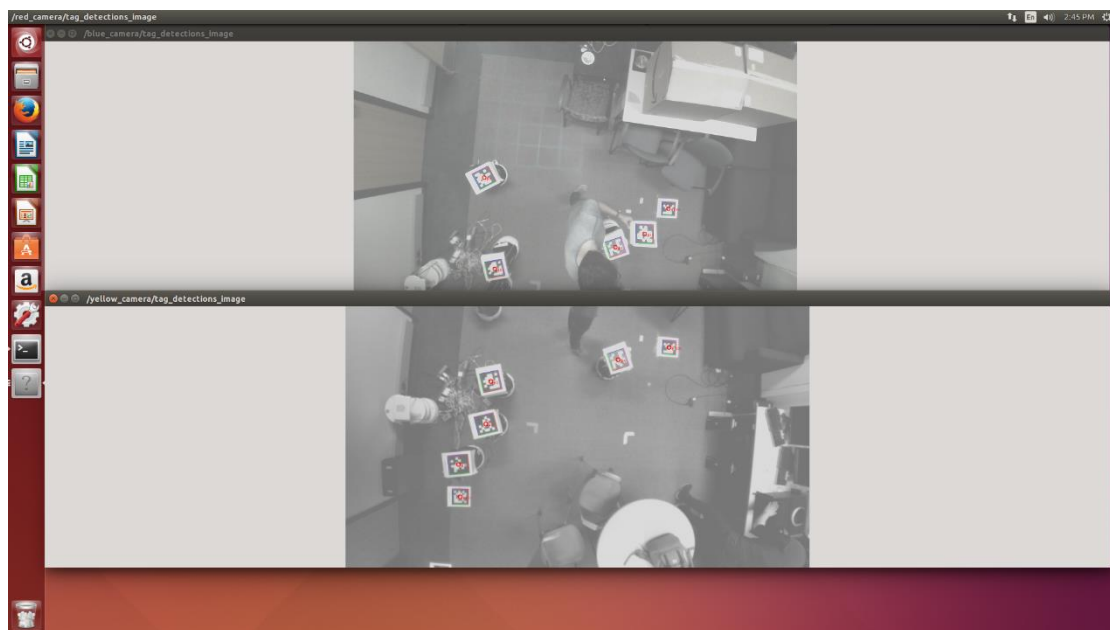


Figure 3

The node in is not publishing position coordinate of turtlebots

The initial condition has to be reset to make the whole flocking code simpler. Now the expected initial condition is when all the turtlebots are in the frame, the coordinate will be got. Given the blind spots issue mentioned earlier, this condition has to be modified to fit the scenario that some of the turtlebots may disappear for a while.

3D printer malfunction

Getting the new pan-tilt units in the last moment made it a little hard to adjust all of

the parameters. But my speed of adjusting could be improve if I focus on the most critical issue and don't get in the detail too much first. We might need to check all the hardware condition (such as 3d printers) before setting our goals.

Teamwork

Gauri integrated the current subsystems and set up all the software on the new chromebooks, but the different version of kernel between chrome book and her laptop made the crushing of system happen more frequently. She's planning to update the version of kernel in the chromebooks.

Sida implement the say cheese function while the target person is not smiling. Once the smile is detect, the voice of "nice smile" will be sent out. She also cut the original frame to make the target person stand out and a publisher to send out the coordinate of the April Tag.

Jimit tried to implement flocking function and localization. For now the whole skeleton is done.

Rohit set up the hardware system for pan-tilt units and April Tag.

Future Plan

1. Gauri will manage to make the turtlebots arrange around the target person.
2. Gauri, Jimit and Sida will work together to fix the system crushing issue while using multiple windows and ssh issues of the connection between two chromebooks.
3. Rohit will set up the initial position of servo motors and make them face the center instead of the initial 0 degree using a state machine.
4. I will adjust all the three pan-tilt units and make them working on chrome book, then go back to implement flocking function with Gauri, Jimit and Sida after finishing adjusting part.