Jimit Gandhi
Individual Lab Report 10
Progress Review 11

Team G -Robographers

Other team Members

Rohit Dashrathi
Tiffany May
Gauri Gandhi
Sida Wang

Introduction

In this Progress Review, I have worked on implementing the overhead localization as well as the flocking algorithm.

Individual progress

Last progress review, I faced a challenge which required me to delay the completion of this long time pending task. The challenge was basically due to the change in focus settings in one of the overhead camera. So in this progress review I first started off by calibrating the intrinsic matrix of that camera.This is a 30-60 minute process where you download a camera_calibration package and rosrun the Monocular camera node. After that you just follow the instructions and move the checkerboard according to the instructions until it calibrates all the parameters. This task was done last time by Gauri.

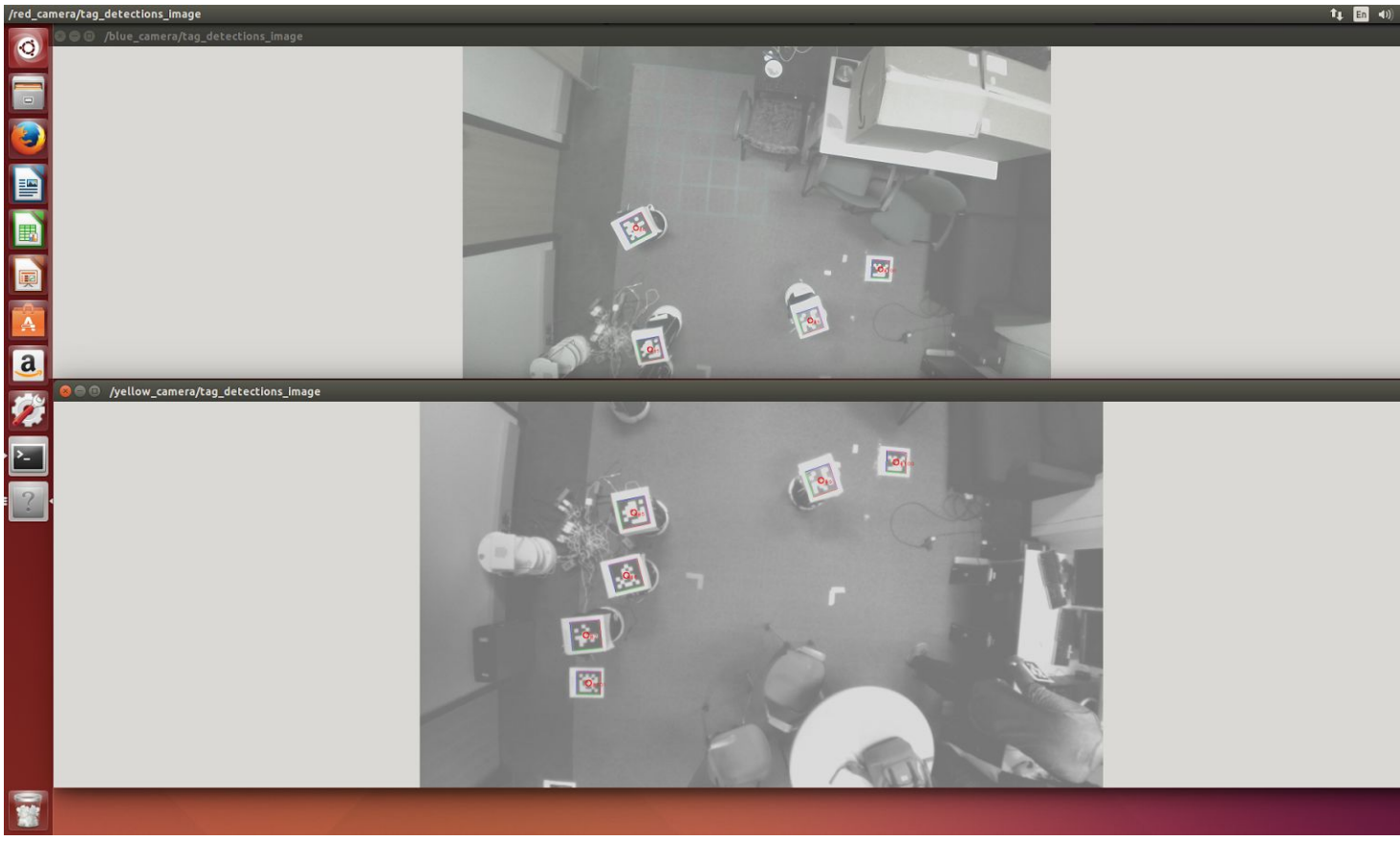

Figure 1: Overhead Localization. The tag to the extreme left (tag_id:100) is used by both the cameras to get their extrinsic matrix in global frame. Similarly it was done for others.

Once that was done, I entered the values of that matrix into the overhead localization file which I and the Phd advisor Sasanka Nagavelli had implemented in unison. The idea was to glue April Tags to the floor in the areas that are common between the two pairs of nearby cameras. We have installed 4 cameras in total and thus we have 3 such common areas **[Figure 1]**. The April tags are mounted in such a way because the camera may move due to the various external reasons. If such an incident ever occurred than we may have to calibrate the extrinsic matrix of

the camera which defines the camera pose in the global frame and do that for each camera. So to automate this process, I wrote a ros node that would launch with every instance of the project launch. On the launch the first thing that would happen is that the cameras would detect the tag ids on the floor in the common areas and discard the rest. The node would then calculate the extrinsic matrix transform for each pair of cameras and thus we could use frame transformation to find all the cameras and the Apriltags they detect in one common global frame. In our Turtlebots (robographers), the April tag size was oriented in a different way than Sasanka had. So I had to change the April Tag parameters on the file as well and test the code since the code only accepts April tags of a specific size. If other size April tags are shown to the camera then it gives out incorrect April Tag poses thinking that they are far away or near instead of small size and big size.

Figure 2: Bio-Inspired flocking in birds. Only neighbours within certain distance account for flocking behaviour of an agent.

Since this code was working before, I was under the impression it would work the same way on my system as well. So i moved on to the next most important thing, implementing the flocking algorithm.

I referred to the Team Roborn website to take a look at their code but could not access their repository as Allard had made it private some time back. So I read his ILR where he had mentioned a reference of the original paper on Flocking algorithm by C.Reynolds [1]. I went through the algorithm. The algorithm basically is based on bio-inspired school of fishes or a flock of birds. The intuition is that the birds when they fly are unaware of the entire world i.e. they do not know the position and velocities of all the birds in the flock but only of those around them **[Figure 2]**. The birds while they flock do not collide with one another, thus they may change their velocity direction when they come too close to the other bird. The birds also move around with almost same velocity. Based on these observations, Reynolds devised a model with three functions that these birds would follow. The functions are as follows. **[Figure 3]**

Cohesion: The velocity and the trajectory of the agent in question is only affected by those near it. Thus we define a certain R_attract as a circular approximation to the area within which an agent could affect other agents velocity. In other words we only consider the agents within this circle with the agent in question as the center. For each such agent found we add its velocity vector such that (Pseudocode)

\

```
Function Cohesion
   {
   agent_in_Question.Vel=0;
    For each agent != agent_In_Question
    (if dist(agent-agent_In_Question)< R_attract)
    {
      agent_In_Question.Vel+=(Agent.Vel- agent_In_Question.Vel)
      //same for y
    }}
```

Repulsion: To avoid colliding with each other the agents should move away from each other if they happen to come too close to one another. So we define a R_Repel as the safeguard distance within which if the agent is found then it will experience a virtual repulsion force. So for each agent_In_Question

```
Function Repulsion
{
  For each agent!=agent_In_Question
  (if dist( agent-agent_In_Question<R_Repel))
   {
     agent_In_Question.vel= - agent.vel   //opposite velocity vecotr
  }}
```

Alignment: In this function, if the agents are within R_attract and beyond R_Repel they tend to move in a similar fashion. Reynolds defined this function as each of the agent moving towards the center of mass of the entire flock.
Thus

```
Function Alignment
{ number_of_agent=0;
  For each agent != agent_In_Question
if (dist(agent-agent_in_Question)>R_Repel & dist(agent-agent_In_Question)<R_Attract))
{
   agent_In_Question.Vel += agent.vel
   number_of_agent++
}
 agent_in_question/=number_of_agent;
```

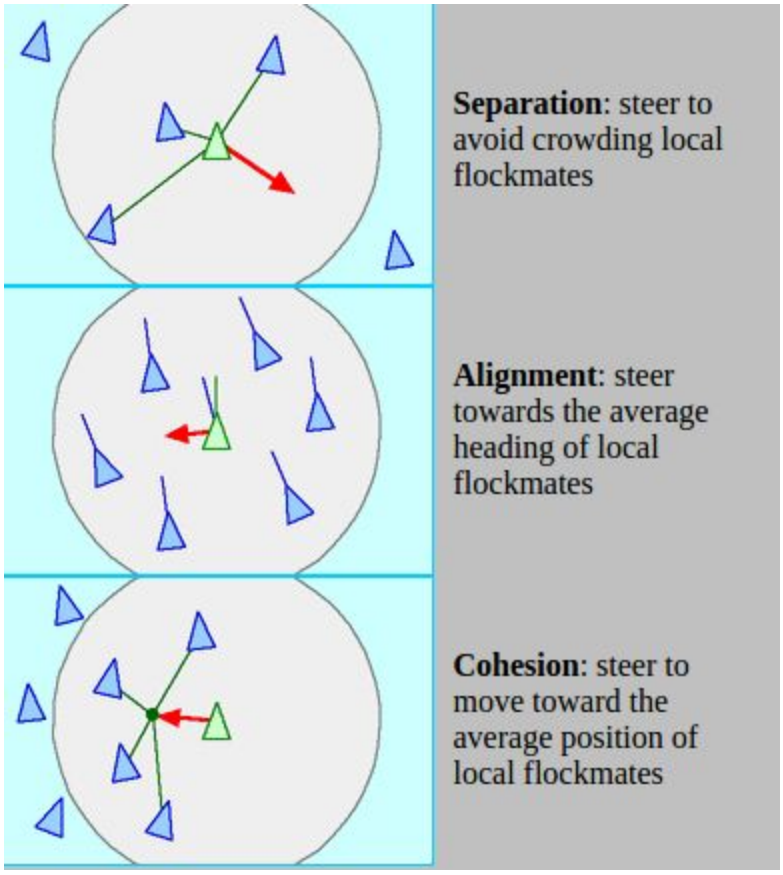

Figure 3: The 3 functions Separation(repulsion), Alignment, Cohesion

I studied the algorithm in great detail of implementation and did it from scratch. So I implemented my own flocking algorithm as a ROS node controller that would take in the global location of each of the turtlebot given by the overhead localization node I described in the above paragraph and velocity values by subscribing to the Turtlebot odometry topic as well as the IMU readings. Using them I implemented the flocking controller and tried to run them with the overhead localization node. But to my surprise nothing worked since I was not getting the overhead_localization readings to the flocking controller at all. I got upset for a while trying my best to fix what went wrong but I could not determine the reason. I re-checked the overhead_localization code line to line and found it to be correct. The tags were accurately being detected, the values were correct, the node was correctly created but the topics were not getting published and I had no idea why this would happen. Unable to determine the reason and determined to test the flocking implementation, I decided to modify it and make the system flock using relative localization. I had the velocity of the turtlebots from odometry and all I needed was a way to get a relative pose estimate between the Turtlebots. So after few hours of tinkering, I came up with the idea of using the person's April tag detection code we implemented in the previous semester for a single turtlebot system and use the April tag psoe readings relative to turtlebots and determine the relative distance between the turtlebots. So I changed the flocking code that would do all the necessary transforms and find the relative distance between any two turtlebots. Gauri created a Rapp package for this flocking behaviour so we could establish parameters for successful

multimaster rocon set up. I tried to run this node and again nothing worked. All the nodes were launched, but no topics were getting published. This led to dismay, but I still did not lose hope. I spent the most of the night before the Progress review trying to find the cause and debug this issue but all efforts were in vain. I will try my best in the next progress review to focus most of my attention to this issue and complete the project.

Challenges

The major challenge was what I described above. Although after Progress Review, I set up a meeting with Sasanka to fix this issue. Apparently the program was perfectly right and should work correctly. The bug was that since the lab has just one server connected to overhead cameras, all of us had different accounts. So when me and Sasanka developed the code, we uploaded the entire package to his Git Account. After that he created a Username account for me on the server so I could use that server for Robographers. When I downloaded the package for overhead_localization from his Git as it is without any modifications except for the April tag size value. After some tuning, Sasanka pointed out that when he created my Username account he forgot to mention that I needed to set some variables in my .bashrc files to get the ROS files configured so that they work for Rocon communication since the server was not connected over wireless connection of the lab but instead was connected via lan wire while Turtlebots were connected wirelessly. So one has to instantiate fixed IP and port numbers so the Rocon works correctly. And thus we finally figured out the reason why topics were not being published to the Turtlebot. Now I have corrected them and in the next Progress Review I should be able to demonstrate flocking and other navigation subsystems.

Another challenge we faced is the random and frequent crashing of the chromebooks. Again we are not sure of the reason but I read the error message which said Kernel Panic mode which is basically an inherent kernel problem and hence we have to look for a stable version of Linux installed because none of my teammates including me have the capability to hack into the root and fix this error. This error is caused and according to my knowledge is not fixable since it is the hardware kernel architecture and its compatibility with the Linux. Our program is actually abusing the kernel as we are launching many video windows in parallel. We may try to reduce the video windows in our program and if that does not work then we may try to install another version of Linux.

All of my work for this Progress review did not account to anything substantial apart from learning lessons. I learnt the lesson to keep my head calm and cool. It helps one to think clearly, take a step back and approach problem in a different perspective. I should have checked the localization node before implementing flocking. If I had detected the problem earlier then it could have been possible for me to detect it beforehand since one tends to get stressed during final hours before Progress review or any such demonstration. So this PR was all about systems engineering and not much of technical. My team members were very supportive especially Gauri and Sida. I should have tried my second implementation of flocking using my laptop as server as it was all relative localization and we did overhead camera for global localization thus eliminating the need for central server. I could have done that but again, my brain was in no condition to think prior to progress review.

Teamwork

1. Rohit completed the entire pan tilt hardware with Turtlebot integration. He also did a fantastic and smart job figuring out where to mount the April tags since April tags should be seen by the overhead camera for localization.
2. Sida implemented the voice command feature which detects if the person smiles or not. If the person does not smile then it plays a voice command that says “Say cheese
3. Gauri and Sida performed integration of all the detection subsystem which incorporates expression detection by multiple cameras, voice command feature, photo clicking into the multi-master setup.
4. Tiffany fine tuned the pan tilt unit for the new design to make it follow the face . She made some modifications such as the pan tilt now uses the April tag to pan and persons nose coordinates to tilt for reasons she may mention in her ILR.

Future Work

This PR was not very inspiring and we did not achieve the goals we thought we would. So first we would finish the goals of this PR by mitigating two major challenges. I would mainly work on navigation part and finish flocking and other parts of navigation where robots arrange themselves around the person. Once this is done I would focus my attention towards mitigating the issue of the crashing chromebooks. Until then, as suggested by Professor John Dolan, my team would work towards completion of the subsystems and integration using their own machines. I will also focus my best efforts towards finishing the entire navigation subsystem.

References

1. http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/
2. https://sites.google.com/site/mrsdproject201415teamc/