An Experimental Environment for Mobile Sensor Deployment for an Urban IoT Project

Team Member: Weiyu Huang, Wenzhuo Wang
Advisor: Solmaz Sajjadi Kia, Pai H Chou
Quadcopter Swarming

***Abstract*- Our main goal is to build a large-scale mobile monitoring system. We expect to see multiple quadcopters swarming together finally. After we enter various destinations to multiple quadcopters, each quadcopter is supposed to realize real-time positioning through deployed mobile sensors and optitrack, including locating its own position and the desired destination, strictly follow the flight path received through commands, avoiding all obstacles on the way, and successfully arrive at the target location. During this quarter, our group have already setup quadcopters, mobile sensors, and Optitrack, which contributes to locating and sending commands back to quadcopter. We also finished implementing some basic autonomous flight commands of quadcopters and hope to achieve quadcopters swarming in the next quarter.**

I. INTRODUCTION

In the extension of cities and technology, there is always a need of surveillance to monitor for incidences of interest. Traditionally, the surveillance systems are stationary, and they are only able to cover limited areas. To achieve reliable monitoring via stationary sensors in a large area, it is necessary to deploy a huge number of them. Even in cases where the cost is not a major prohibitive factor, with the current technology, the communication bandwidth utilization certainly is a limiting factor. Therefore, to solve the coverage within the limits of the system, the use of mobile sensors, which the infrastructure can move within the urban area is of interest (especially aerial sensors that have wide measurement zones). The aim is to compensate for the lack of full spatial coverage at all times by context--aware temporal dynamic distribution of a set of mobile sensors.

The purpose of this project is to design a program that can control a group of quadcopters to jointly accomplish a global deployment task. For this system to work, we need to design a reliable localization system that can give the quadcopters the sense of where they are located. Along with the theoretical development, we look to setup a state-of-the-art experimental environment consisted of several quadcopters that demonstrates our algorithms with respect to an information background that is projected on the ground via a high--resolution projector.

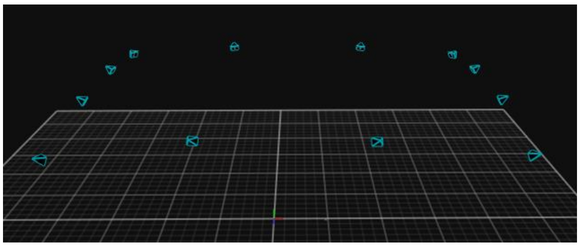
II. HARDWARE

Crazyflie

We choose Crazyflie 2.1 which is small and could fly in our simulated environment. It is the most suitable one since all the current tests are in lab and the area for quadcopter is limited. The Crazyflie 2.1 is 50 grams and could carry up to 90 grams object. We set retroreflective ball on Crazyflie 2.1 for locating system to capture its location (Giuseppe Silano, Emanuele Aucone, Luigi Iannelli, 2018).

Optitrack

For the flight of quadcopter, it needs an outer system which could detect its location and feed the location information to program. We select Optitrack as the locating system. Optitrack is the system including 12 cameras above the ground (Gergely Nagymáté, Rita M. Kiss, 2018). With the help of Optitrack, we are able to directly visualize movements via Motive tracker. The virtual location is shown below:



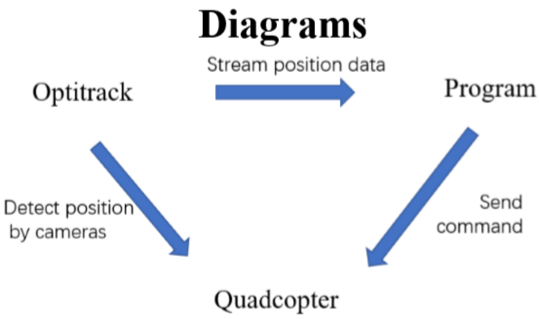
Each camera could capture the object with retroreflective material, and the location information would be calculated on Optitrack program. We streamed the location information to our program via internet (Aerospace Controls Lab).

III. SOFTWARE

Our main program is written in Python 3.7. It is based on an open source library cflib, which is intended to be used by client software to communicate with and control a Crazyflie quadcopter. More specifically, in the program, there are 2 main part: one controls the flight of Crazyflie 2.1 and the other one receives location information from Optitrack via a router (Arnaud, 2019).

We handle the quadcopter and locating system by thread function, so in the main program, it ought to keep sending location information to quadcopter in one thread. Simultaneously, it sends the command to the quadcopter.

We build a UDP connection with the program of Optitrack via router. The Optitrack program streams the data to the router and the data goes to personal computer. The workflow is shown below:



IV. FUTURE PLAN

We will set a projector to simulate an environment for Crazyflie 2.1. In this final stage, the quadcopter would receive the environment information, including locations and obstacles, and move from its current location to designed destination.

In addition, we would also need to implement a friendly user interface for the final version. The user interface is built for users to enter the destination to each quadcopter, and then the program will calculate the flight path and send the command to quadcopter by steps.

V. SUMMARY AND CONCLUSION

After the work of this quarter, we already setup the cameras of Optitrack for capturing and build a prototype of program. It could receive the location information from Optitrack and send the command to Crazyflie 2.1. Moreover, we successfully achieved autonomous and stable flight of quadcopter by adjusting the PID data in the open source library. From what has been discussed above, we can safely conclude that the quadcopter is mobile and better than a fixed point. With the mobile monitor, the limited bandwidth could control more monitor and cover a larger area.

APPENDIX

1) We choose UDP connection for receiving the location information. The UDP connection could handle more data packet than TCP connection. The command is sent every 10 milliseconds so the lost of single packet is not fatal. The command to quadcopter is based on radio and the streaming of location information is based on WIFI.
2) The quadcopter is controlled by radio and the range of radio is limited. Currently it is 10 meters. It is enough in the lab because our goal is simulated environment instead of real environment. However, for future implementation of our system, we would need to consider how to magnify the radio signal to extend its range.

   The power of one battery could sustain the flight for 10 minutes. During the test, we have to change battery after 10 minutes and make sure the battery is always in charging.
3) The propeller of quadcopter revolves in high speed during flight. Once it drifts to wrong direction, the propeller is very dangerous. We set a big net to restrict the flight area of quadcopter. Even if we lose control, the quadcopter could only bump to the net.

   The battery is not fixed on the quadcopter, so during flight, it might slip out. We tried to insert the battery tightly, but it is still possible for battery to slip out. We tried to increase its friction by a rough sticker and then the battery did not slip anymore.

ACKNOWLEDGEMENT

We would thank Professor Solmaz Sajjadi Kia for providing us with the lab and all the relrequired materials for early stage. Besides, professor Solmaz also gave us many suggestions on the structure of our project. When meeting any problems, professor Solmaz also helped us find other professors in technology-related to amswer our questions.

What's more, we would also thank Professor Pai H Chou to be our advisor of EECS 159A and allow us work in Kia's lab.

REFERENCES

[1] Giuseppe Silano; Emanuele Aucone; Luigi Iannelli, "CrazyS: A SoftwareIn-The-Loop Platform for the Crazyflie 2.0 NanoQuadcopter", ISBN 2473-3504, 2018 26th Mediterranean Conference on Control and Automation (MED)

[2] Aerospace Controls Lab. "Decentralized Planning Using Macro-Actions: Dec-POSMDP Hardware Experiments." YouTube, YouTube, www.youtube.com/watch?v=34xHxXrnPHw.

[3] Arnaud, "Making plans for the Crazyflies app-layer", Nov. 2019

[4] Gergely Nagymáté, Rita M. Kiss, "Application of OptiTrack motion capture systems in human movement analysis A systematic literature review", Vol 5, 2018/07/02, Recent Innovations in Mechatronics