Drone Pet

Aditya Bansal Dhruv Kaushal

University of California, San Diego

ABSTRACT

We all know how big photography is for travelers, but very few of us understand the discomfort these people have to go through to capture these moments. Carrying a heavy camera, a tripod, and numerous other accessories can be bothersome. Therefore, we decided to create a ready to fly drone, that is fully autonomous, and is capable of recording videos of travelers doing activities such as hiking, skiing etc.
We were able to use a cheap drone to fly autonomously in a non-windy environment, that was easy to set up, and had fail safe features to protect it from crashing. We also developed a custom PID controller, which would calculate the error between the drone's current position and the person to follow, and then suggest the speed the drone should use to reduce the error. This way, we were able to get it working for slow activities like hiking, as well as fast-paced activities like running/jogging. Overall, we were able to reduce the cost of such an autonomous drone from $1500 to $300.

INTRODUCTION

The drone pet project that we worked on for this class mainly consisted of a drone that would autonomously track, follow and record people while they would perform outdoor activities, allowing them to be free of the burdens of carrying a camera, etc. We encountered many different requirements and obstacles to fulfill our goals for the project. The cheap drone that we used for the project required more care due to its unresponsive nature during heavy winds due to its inbuilt flight controller and stabilizer.

In this article, we present how we implemented the functionality of autonomous navigation and tracking, how we were able to overcome the challenges we faced during the implementation of the project, how our time was structured during the quarter through the milestones that we had set and how we evolved our milestones as we went along the project and gained more experience building the navigation system. We also talk about our failed attempts to integrate certain technologies into our project and how we ended up learning from them and finally we talk about what kind of future development this project would require to be commercialized and fine-tuned.

On a high level, our drone follows a specific work-flow so accomplish its job. The ordering of the items in the workflow show the steps required by the drone to be fully functional in its objective and be able to complete the task with as little outside interference as possible-

- Record videos or take images depending on the chosen preference (Skipped for class)
- Fly to the preset flying altitude
- Identify the person to follow
- Navigate and keep the preset hardcoded distance from the person
- Keep tracking person for the allocated time
- Safely land at the end of the allocated time or when the battery is low

This workflow allowed for us to be able to accomplish all the tasks that we had set up for the project, as you will see in the milestones section. Due to time limitations and being forced to use the drone's inbuilt flight controller, we were unable to customize behavior during different weather conditions, which also made it so that we did not have complete control of the movements of the drone in windy environment.

TECHNICAL MATERIAL

We used a parrot ar drone 2.0, and the tags that it came with to implement our autonomous flight controller. This was the only piece of hardware that we had to use, other than the batteries and the chargers.
We also used ROS (Robotics Operating System) running on a virtual Ubuntu machine, that behaved as a messaging system between our code and the drone itself.
To connect to the drone, we used the parrot drone's inbuilt wifi module.

The calculations for navigation for the drone was done in python as ROS nodes, which would run on a computer and then sent to the drone through wifi using ROS topics.

MILESTONES

Our objective for the project was to fulfill these requirements that we deemed necessary for the success of the project-

1. Easy to set up

One of the biggest problems with off-the-shelf solutions that we had encountered was that they required training and calibration after before being able to use them. Thus one of our major goals was to make sure that our drone could be ready to use straight out of the box.

2. Affordability

We wanted to make the drone affordable, since off the shelf solutions costed upwards of $1,500 which we considered to be too expensive for general adaptation of this technology.

3. Recording recreation activities (Skipped for this class)

Finally, we wanted it to be able to record people’s recreational activities such as hiking, skiing, etc. This required it to have a tracking feature, autonomous navigation, etc.

During the planning phase for this project, we had set up the following milestones for the project-

1. **Interfacing the drone with ROS**

We wanted to be able to control the drone using the well-known robotics communication framework ROS. ROS would have allowed us to easily integrate our project unto numerous other drones allowing for greater portability. ROS stands for Robotics Operating System, and is an intermediary between your code/terminal, and your robot, which in our case is the drone.

2. **Semi-autonomous flights**

During the initial testing phase, we had planned to check our drone's responsiveness by coding semi-autonomous flights first, such that we could learn the limitations the drone had on its navigability, outside interferences (like winds) and also to test the controller which we wrote for the drone.

3. **Tag detection or OpenCV**

In order to keep track of the person the drone had to follow, we used the in built tag detection capability of the Parrot AR drone 2.0. This was probably the easiest part of the project as we didn't have to write any OpenCV code or re-invent the wheel. We used the given SDK to write some basic functions that would recognise the orange-blue markers shown in the videos.

4. **PID Controller**

PID stands for proportional, integral, derivative, and is a feedback loop mechanism that is commonly used to correct the error between the current situation, and the recommended situation. In our project, we used it to calculate the error between the position of the drone and the position of the tag. Then we sent this information to the next part.

5. Autonomous flight navigation

One of our major milestones was for the autonomous flight navigation system to be set up, where the drone would follow the person detected by the tag autonomously without any other interference. We had planned to use the inbuilt flight controller and implement the autonomous behavior in python and use it using ROS.

6. Bug fixes and fine tuning

Our last milestone initially had time allocated for bug fixes and fine tuning the autonomous navigation algorithms while doing real world flying tests.

Revisions made during the middle of the quarter

For the most part we had similar milestones in the middle of the quarter, expect a few changes to the tracking part of our project. The drone that we used, came with certain Tags that its camera could recognize which we used instead of OpenCV's object detection for tracking people. Additionally, we switched OpenCV to work more on obstacle detection to improve the safety and survivability of the drone.

At the end of the quarter, the milestones we were unable to implement were related to OpenCV. We had failed to calculate the computing power that OpenCV algorithms took, which made it such that the computational delay that the algorithms cause was long enough for the drone to be unable to track and navigate properly. One of the fixes we were working on included only running the object detection algorithm once a second such that there would be less computing to do and thus less delay, but it caused erratic behavior in our navigation. Also it caused the drone to stop randomly due to false positives and objects detected far from the drone for which it should not need to take any special action.

Another problem we encountered during the course of the quarter was the fact that our drone would follow the movement commands perfectly during still weather, but strong winds made it unstable and unable to move as instructed. This ended up causing a lot of problem due to the fact that San Diego got really windy during our testing phase which made it such that we were unable to collect as much flying data as we wanted to.

OBSTACLES

1. **Testing indoors.**
   As you could see in the final video, the indoor spaces are very small and not ideal to test projects like quadcopters. The way the parrot ar drone works, if it even touches anything on the side, it starts to behave in a very erratic manner.

2. **Small battery**

   The flying time on the quadcopter was rather small, around 7-8 minutes, and it would take a full 2 hours to charge. This slowed down the testing quite a bit, and we would spend a lot of time just trying to test the drone without actually flying it.

3. **Windy environment**

   The biggest challenge we faced was making the drone work in windy environment. As soon as the wind speed went >5 mph, the drone would not working, even manually. While reading about, we realized that this is more of a problem with quadcopters than our flight controller. Some of the higher end drones also struggle with manual flight control in windy environment, let alone autonomous functionality.

4. **Practicality issues.**

   Whilst tag following was a fun project to implement, its not the most practical way to develop an autonomous drone, as it needs ideal conditions like the right wind speed, the right lighting for the camera, the right environment etc. To make a more real world friendly drone, we're going to try incorporating a GPS based following mode to our drone.

CONCLUSION

Overall, we had a lot fun in the class, and we really enjoyed building our own autonomous quadcopter. With the exception of making it work in windy environment, we are very happy with the results of our milestones. We will continue to enhance our project, so in the future we can compete with higher end drones like the DJI Phantom or the 3DR Solo. Next, we want to make it GPS compatible, so it can follow people around without a tag. We hope to finish the next milestones sometime within the next 6 months, and then take our project to market.

Video Link: https://www.youtube.com/watch?v=4CUk7CVd9Zs
Github: https://github.com/maddymanu/CSE-145-