Final Report

A27-Autonomous Bomber

Auto Boom

ECE 592/492, Spring 2023

Team members:
Wesley Cowand(Team Lead) , Amy Deepee ,
Matt Parker, Victor Minin , Dominique Barrera

Table of contents

Abstract

This report describes the development of an autonomous drone system capable of detecting blue targets and dropping payloads onto them. We programmed the drone to scan an area using an onboard camera and identify blue targets using a computer vision algorithm. Once a blue target was detected, the drone automatically calculated the best trajectory for payload delivery and dropped the payload using a mechanical release mechanism. The system was tested in a simulated remote area and achieved a high level of accuracy in detecting blue targets and a medium level of centering and delivering payload.

Introduction

Finding a predetermined target in a known gps range and drop a package on the target if confirmed by ground control.

This system can have important applications in military and security operations. For instance, in a military operation, it may be necessary to quickly deliver supplies or equipment to troops on the ground, or to target enemy positions. An autonomous drone system that can detect a predetermined target and drop a payload on it can provide a valuable asset for such missions. It can also be used for supplying food, medicine, or other supplies to people in a general area.

Overall, the development of an autonomous drone system that can detect a target and drop payloads on them has significant potential for improving the efficiency and effectiveness of a range of operations, including emergency response, search-and-rescue, and military operations.

Related Work

There have been several related works in the area of autonomous drones and object detection. For example, in a study conducted by Khatib. (2017), a drone was programmed to detect and track objects in real-time using a combination of computer vision algorithms and machine learning techniques. Similarly, in a study by Ricken et al. (2018), a drone was used to locate and track a moving target using an onboard camera and an object detection algorithm. In the context of emergency response, Li Zhang. (2019) developed an autonomous drone system that can quickly locate and rescue people trapped in disaster-stricken areas. Current military drones can scan areas, identify moving targets, and drop payloads onto those targets, but as far as we are aware, there hasn't been an instance where they have done all of these at the same time with only a confirmation of targets given from a command center. There have been reports during the war that unmanned aerial vehicles (UAVs) have been deployed autonomously to attack vessels, but it is unclear whether this has been verified. Nevertheless, our project represents a novel contribution to the field of autonomous drone systems by specifically focusing on the detection of blue targets and the autonomous dropping of payloads onto them.

Approach

1. Scan: The first step is to scan the area of interest by flying a drone over the specified GPS coordinates and capturing images of the terrain below. The goal is to achieve even coverage of the area with minimal overlap and no gaps. The altitude of the drone is determined based on the desired resolution of the target and the size of the field being tested. During testing, a wifi connection is needed, but the ultimate goal is to not need a connection until the drone is ready to receive confirmation.
2. Detect: Once the images are captured, the next step is to use computer vision techniques, specifically the OpenCV library in Python, to detect the target object of interest. In this case, a blue tarp is used as the target. The images are first processed to filter out unwanted colors and isolate the blue color range associated with the tarp. The center of mass of the blue area is then calculated to pinpoint the location of the target. The mask image is also blurred using a Gaussian Blur and Median Blur to reduce the impact of small, stray pixels and improve the clarity of the target.

Figure 1: Image taken by camera on drone prior to processing

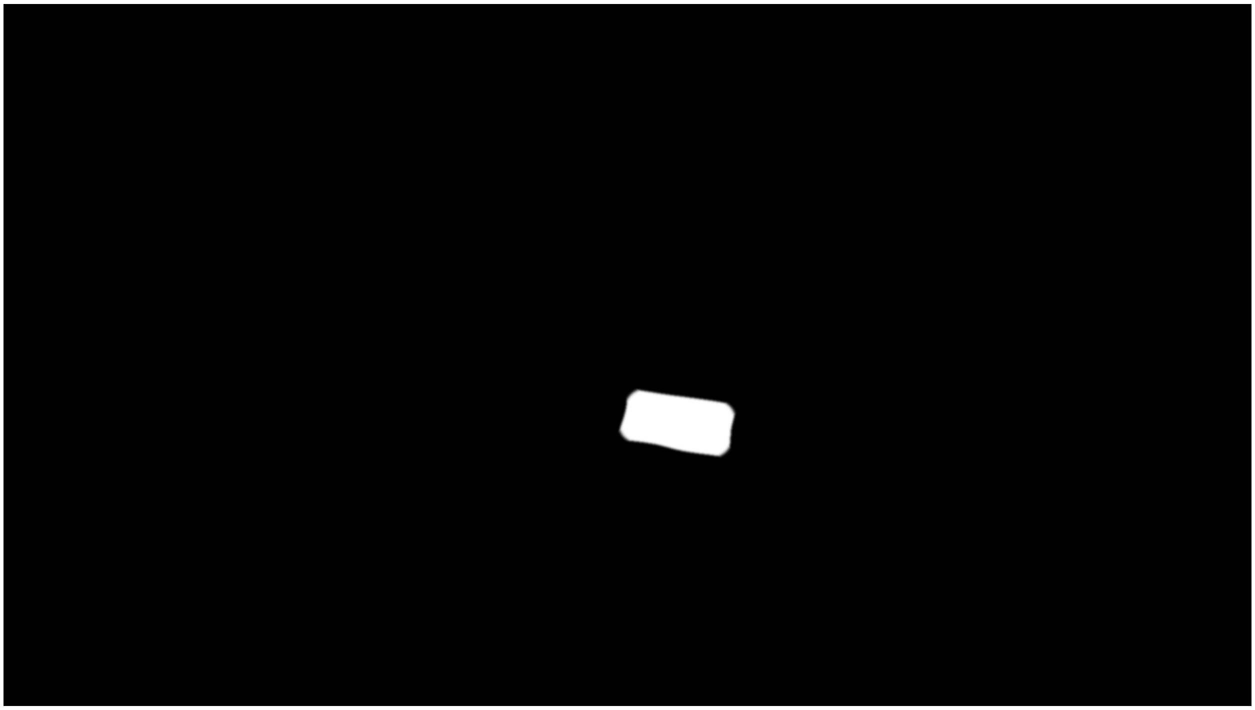

Figure 2: Processed image mask created using OpenCV in Python to detect the blue color range

Figure 3: Center of masked region found and dot placed on the location

Figure 4: Image Taken with no tarp present

Figure 5: Mask of image with no tarp present

3. Estimate: With the location of the target identified, the next step is to estimate the distance between the drone and the target. This is done by calculating the distances dx

and dy based on the image resolution and altitude at which the picture was taken. These distances are then converted into degrees of latitude and longitude using the circumference of the Earth. The estimated coordinates are added to the original GPS coordinates of the drone to determine the expected GPS location of the target.

4. Communicate: The estimated location of the target is communicated to the ground control, where it is plotted on a map with the same GPS range that was scanned using the drone. The user selects the target from the map and the drone receives the command to either return to launch or go to the estimated location.

Figure 6: GCS view of map with blue dot of tarp location sent from drone

5. Re-acquire: Once the drone reaches the estimated location, the detect phase is run again to refine the location of the target. The drone is instructed to move in the x and y directions based on the raw estimated distances in meters, rather than relying solely on GPS coordinates. The drone continues to run the detect phase until it has calculated that the center of the blue tarp is within 5% of the expected location.Then this process shall be repeated at 20 meters such that the accuracy is increased even more.
6. Drop: When the drone is centered over the target, it sends a signal to a servo to release the payload.
7. Return to landing (RTL): Once the drop is complete, the drone is instructed to return to the launch location.

Experimental Results and Analysis

We did approximately 25 flights in the course of a month. Many of the first flights were figuring out the communication between the drone and our systems since they weren't exactly the same as with the virtual drones.

Our first huge breakthrough was getting the drone to travel to the desired predefined waypoints.

We flew additional flights to gain actual pictures for color correction and target detection fine tuning

Our final flights were figuring out small errors in our code that would cause the drone to go to a location we didn't intend. And fixing issues with the accuracy phase. Many of the issues were from getting x , y, latitude, longitude, north , and east variables getting flipped.

Climb to 50 meters-> error -> RTL

Climb to 50 meters-> Go to waypoints -> error taking picture-> RTL

Climb to 50 meters-> Go to waypoints -> take pictures -> no target-> RTL

Climb to 50 meters-> Go to waypoints -> take pictures -> all are targets->RTL

Climb to 50 meters-> Go to waypoints -> take pictures -> ID target -> send estimates gps to gcs-> wait->confirm-> go to estimated location-> not acquire target -> RTL

This was a common flight pattern on the last day. This was being caused by an issue in the detect code where it was meant to subtract the target center from the image center. The code would calculate the offset from the drone to the target center , then add the offset to the target center causing the drone to be twice the distance away from the target.

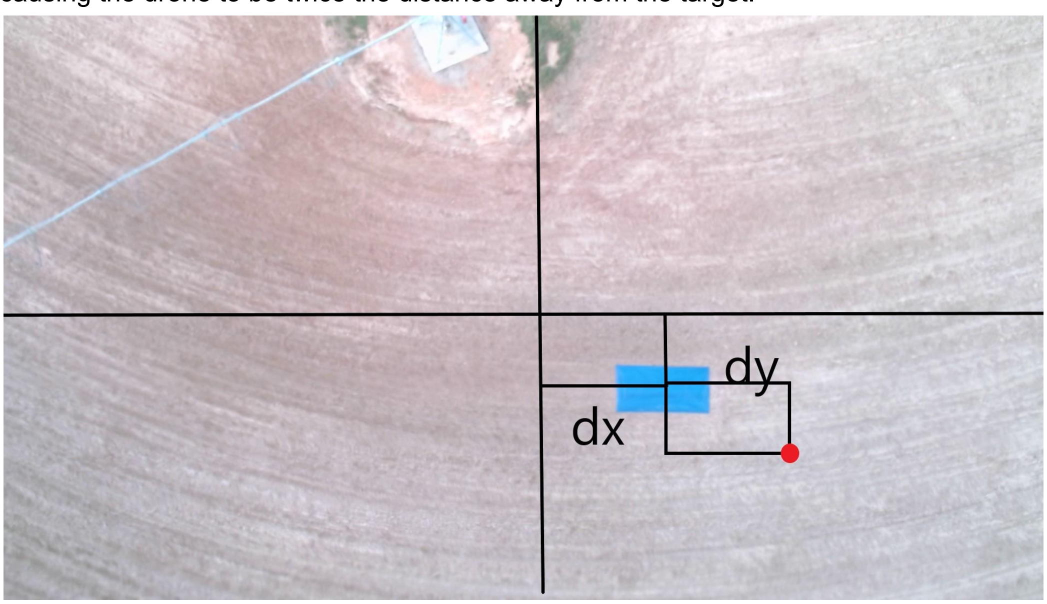

Figure 7: Representation of the error in using the center of the tarp rather than the image

This would have been an easy fix to find and solve if it had not coincided with an additional error in the code that caused dy and dx to be flipped resulting in the drone moving north east instead of south east.

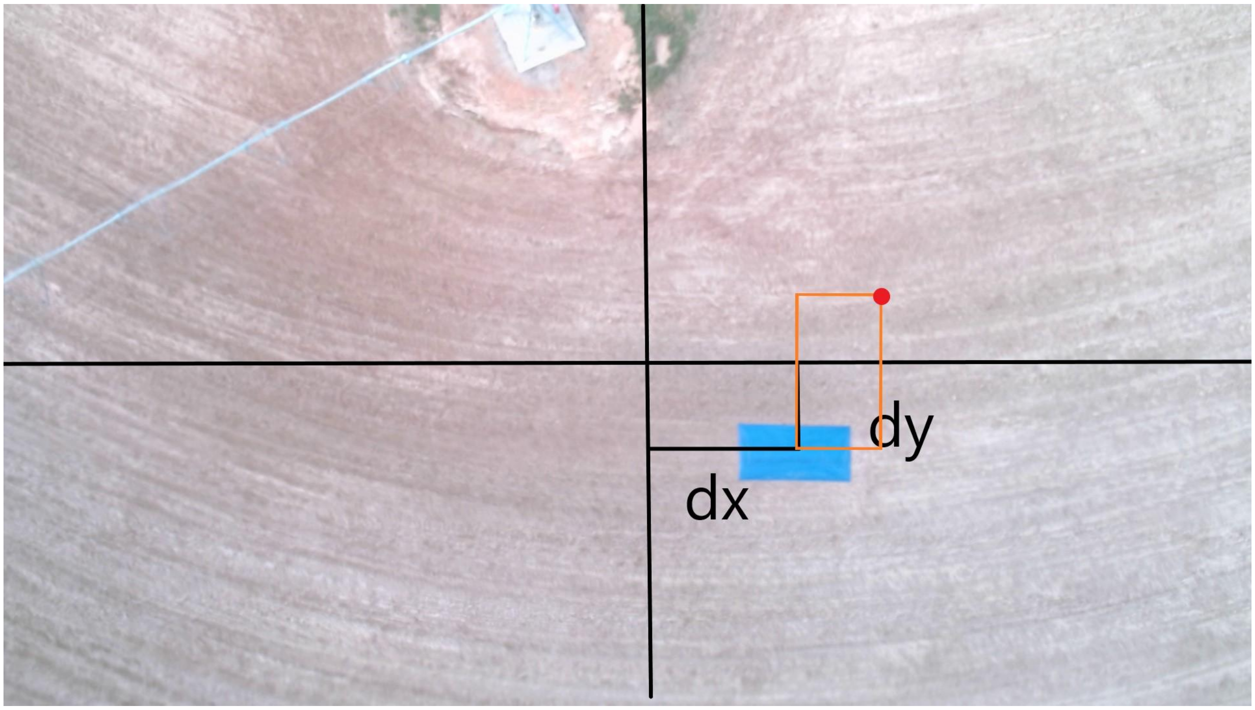

Figure 8: Representation of the confusion between N/E and S/W in the implementation

Conclusion

Overall, the objective of our project was to conduct a field search, go through waypoints, and capture images at each waypoint to perform computer vision analysis on them. The purpose was to identify the target, which was a blue tarp. Once the tarp was identified, we determined the center of the tarp by analyzing the pixel values in difference from the X and Y center of the image. We then used these pixel difference values to determine the GPS coordinates of the center of the tarp.

After the drone completed going through all of the waypoints, it returned to the GPS coordinates of the center of the tarp to perform a centering algorithm. This algorithm calculated the difference between the center of the tarp and the center of the image to determine the difference in X and Y meters that the drone had to travel. The centering process continued until the center of the tarp was within 5% of the center of the image, and the algorithm loop ended. We repeated this centering process at 40 meters and then again at 20 meters.

Once these two centering processes were complete, the drone released the payload onto the tarp. Finally, the drone returned to the launch site.

One of the biggest challenges of this project was the variability of the sunlight's effect on the tarp's color values. Our filter that we used to identify the tarp had a low filter value and a high filter value. For example for the filter values with the low filter range in BGR being [130,80,0] and the high filter range in BGR being [255, 160,30] the worst case scenario of the color identified is [130,160,30] which is a green as shown in the image below at the top.

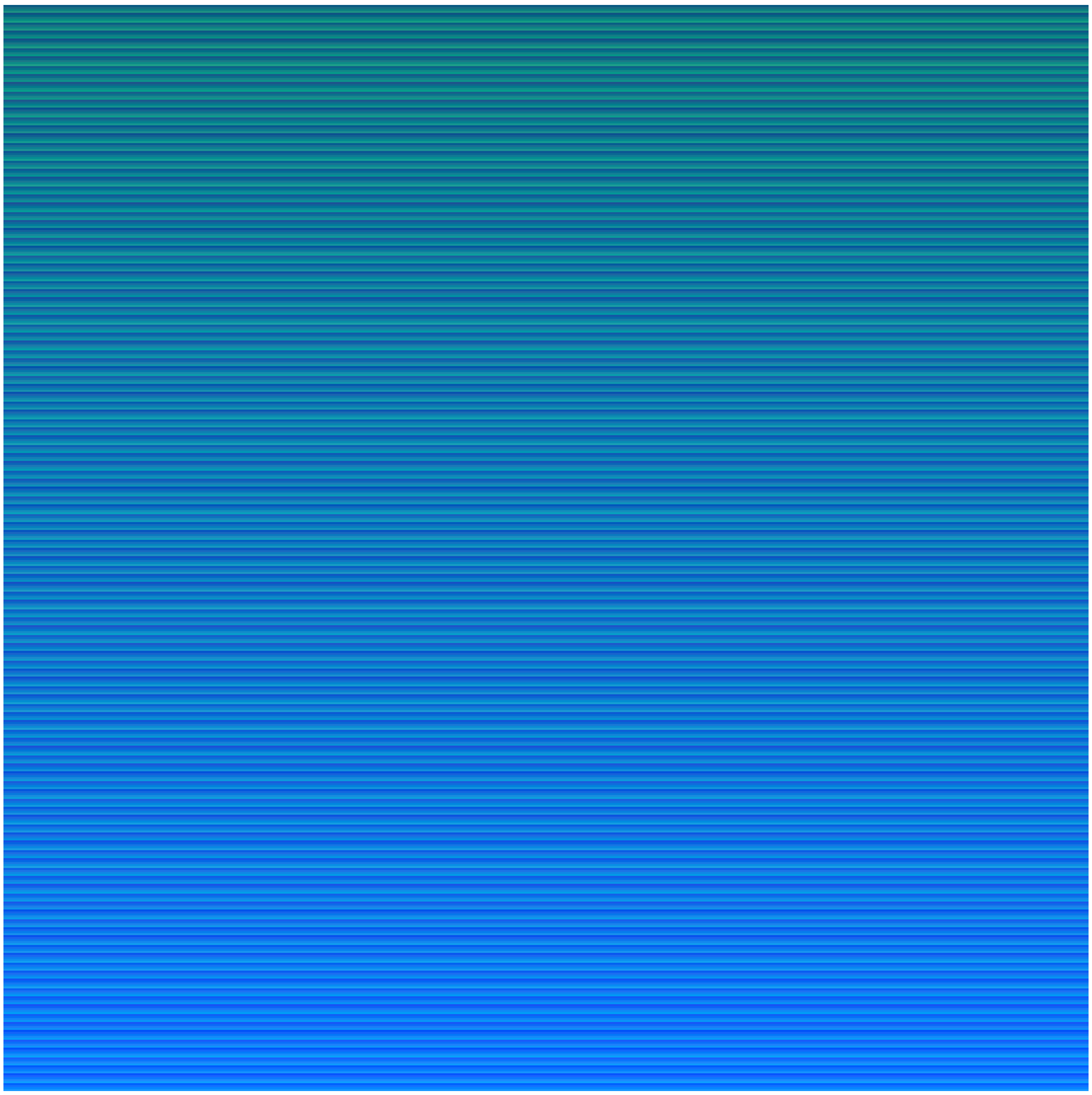

Figure 9: Blue Range used for Detection

The main issue is that the sun's light adds white to the image in all channels meaning that the green and red values will all be higher which makes the worst case scenario for the filter more green which can be confused with the field's grass.

We found that after taking images of the tarp in the field with a certain sunlight condition we could experimentally determine the correct filter values, but after the sunlight conditions changed the filter would not work. Our recommendation for the next iteration of this project is to improve the low and high filter ranges.

References

Khatib, O., Zimmermann, K., Basiri, M., & Eichenseer, S. (2017). Object detection and tracking for autonomous indoor and outdoor flying vehicles. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4037-4042.

Ricken, T., Konig, A., Gavrushenko, E., & Jain, A. (2018). Robust real-time visual tracking for autonomous UAVs. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 5314-5321.

Li, W., Zhang, X., Wang, H., Chen, L., & Chen, X. (2019). A multi-UAV system for disaster response: Design, implementation and field experiments. Journal of Field Robotics, 36(1), pp. 142-161.

The Drive. (2021, April 15). Ukraine unleashes mass kamikaze drone boat attack on Russia's Black Sea Fleet headquarters. Retrieved from https://www.thedrive.com/the-war-zone/ukraine-unleashes-mass-kamikaze-drone-boat-attack-on-russias-black-sea-fleet-headquarters

Stack Overflow. (2011). Calculating new longitude/latitude from old + n meters. Retrieved from https://stackoverflow.com/questions/7477003/calculating-new-longitude-latitude-from-old-n-meters