Note: Ferb = Robot

Progress

Introduction

This lab has the team implementing basic computer vision on Ferb so that they can identify and chase an item. The first step to accomplish this is finding the object based on color and creating a mask with a positive of the object. The second step is using this mask to create actions, like drawing a circle and moving Ferb.

1. We converted the image to the HSV color space because it is difficult to mask based on color while defining colors in RGB. HSV simplifies selecting certain colors unlike RGB that struggles because many colors are combinations of red, blue, and green. From there we found general values for yellow (we are tracking a yellow beach ball) and iterated from there. Using our found values we created a dictionary that could be referenced instead of the sliders in the original code.
2. To find the target in the image we used the connected components command in CV2. This returns information about the components in the image; our mask is black and white so the background and our target would be the two main components followed by any noise that makes it into the mask after our filtering. Our filter was chosen to be an opening followed by closing to remove noise from the mask while maintaining the same general shape of the target before the filtering. Like choosing our HSV values, this had a lot of iterations. After filtering and finding the connected components, we could identify the target using a circle imposed onto the image.

3. Control of Ferb to follow and track the target was done using a state machine. We chose to start in the search state where Ferb will look for the target; from there they move into an intermediate state which decides which state is most appropriate to use next. The intermediate states were helpful because they allowed us to cut down on repetitive statements that branched to each state. We followed the general outline given in the lab for states as it was easy to implement by checking where the center of the target was.

Data

Thresholds for the HSV values discussed above were determined live using cv2. In the pictures later in the report the sliders can be seen along with some of the older HSV values we tried. The final HSV values chosen were:

- Hue: 0-90
- Saturation: 155-185
- Value: 130-230

Algorithms

The state machine used by Ferb can be seen at the end of the report and described above in the progress section. It follows the basic states outlined in the lab instructions.

Bonus

Results and Media

Results

Ferb's performance was adequate in most scenarios. At the end of the lab, they were able to identify the yellow volleyball consistently from the background and the state machine worked smoothly to control movement. A challenge throughout the lab was our color choice: yellow was very hard to distinguish because of how bright it is and how close it is to some other objects in the room. We consistently dealt with high amounts of noise stemming from yellow buckets and wood shelves that were within the HSV color range of the volleyball. There was nothing that we could consistently do that would eliminate these without also harming the performance of our computer vision. One thing we hoped to do that our robot couldn't is identify a yellow tennis ball. The tennis balls were too bright in the camera, bringing the highlight to nearly white which matched too much of the room. What was left was too small to get consistent performance out of.

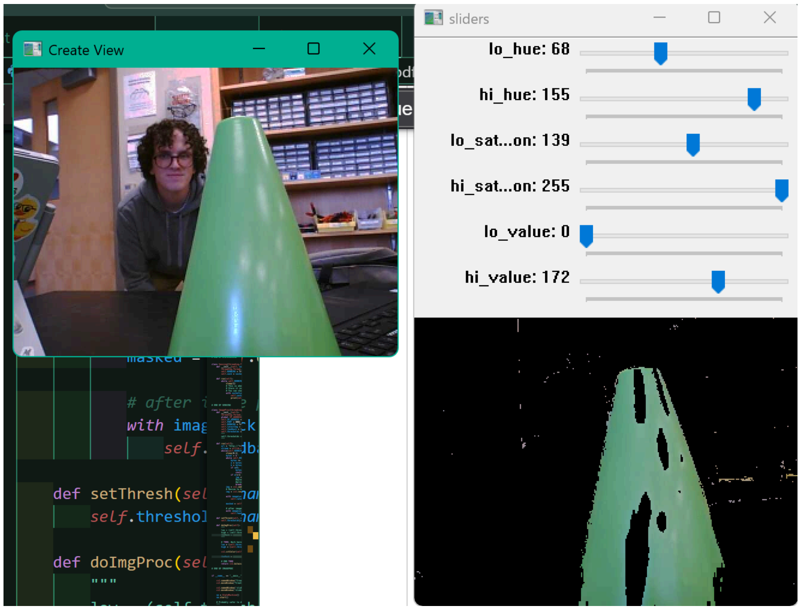
A couple larger issues arose throughout the lab, many of them easily solved. The first issue we had was defining our color space incorrectly; we did not use the full gamut of hues initially leading us to a slow start. After noticing our error, we quickly fixed this and were able to get a lot more performance out of our mask. Another issue was defining the image size wrong. We had a major inconsistency that we thought was from the state machine, it turned out we had mistaken the image size so our state machine never reached some of the states. After changing the screen size, the state machine we created worked perfectly.

Another big issue we dealt with was getting Ferb to move. We struggled with understanding the dimensions of left and right in terms of the width of the frame. Our initial approach involved using the cv2 command that would recall the width of the frame, which we would then divide for left and right. There were quite a few errors with this approach, with the big one being that Ferb would just not move. Our final solution was to manually use the size of the frame which worked with our code and permitted Ferb to move. Once this change was made, Ferb was able to turn left and right in order to detect the beach ball and stop when it does.

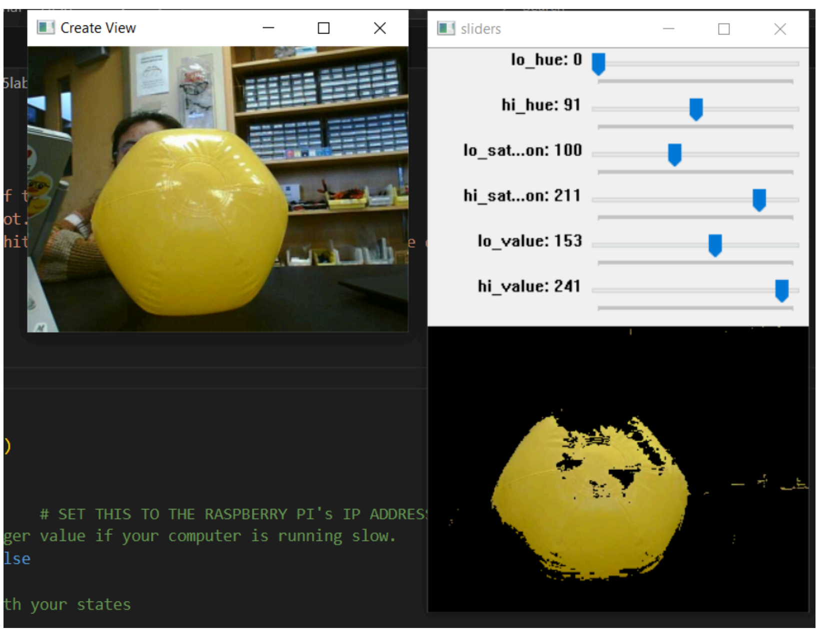
Pictures

Image of Robot

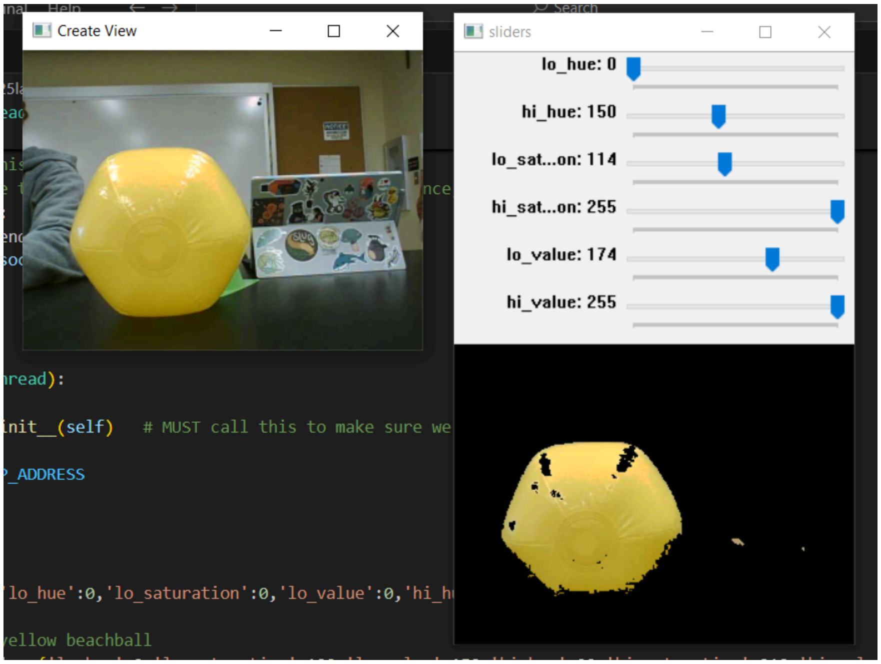
Selfie of group with robot

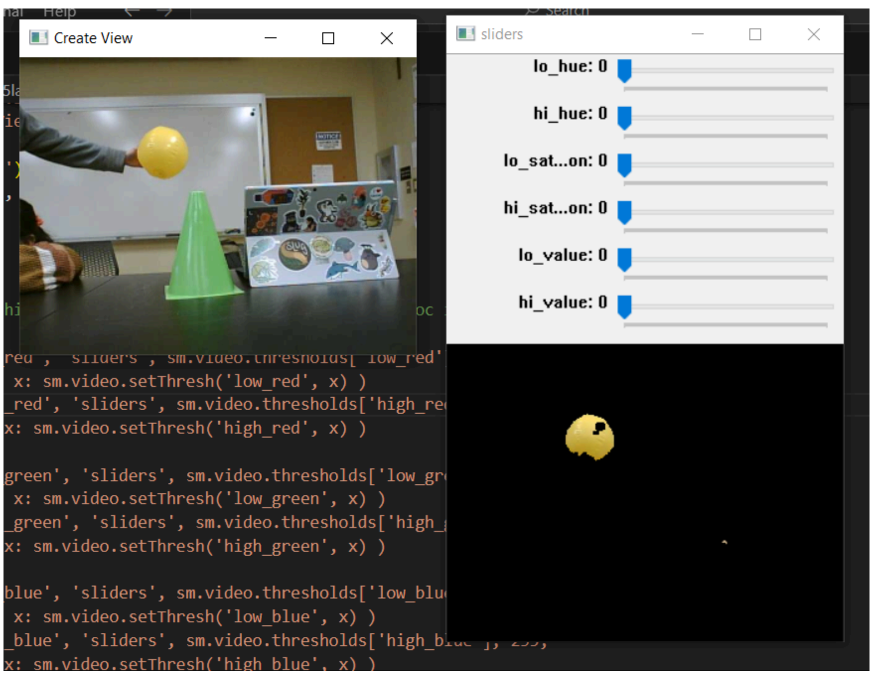


The above image corresponds to the first part of the lab where we were setting up the sensors and identifying a green cone.



This image corresponds to having the sensors set to identify yellow images.





The above two images are when we were testing distance on the image identifier.

Videos

Video of Ferb

This video shows various different iterations of our computer vision on Ferb. The first clips show Ferb in the search state where they rotate to find the target. As the video goes on, the state machine progresses and in the last clips the end goal of tracking the target is achieved. The last clip shows both Ferb's actions and what they see through the GUI on the laptop.

Reflection

This lab showed how easy and hard computer vision can be. While it was easy to get quite basic results out of Ferb, it became much harder to get them consistently with varying conditions. We initially set up the thresholds on the table under direct light but after moving to the ground so that we could drive Ferb, these thresholds would often fail. It was easy to remedy but showed how easy it was to throw our system off using something simple like time of day. Given more time, we would try to incorporate more targets of varying colors that could be toggled between. Another stretch goal would be testing more dilation and erosion combinations. We were often left with small amounts of noise in the mask even after aggressive erosion. It would be nice to get rid of this given time to focus on it.

Our team's bonus would be implementing the different speeds in our state machine and using a yellow volleyball. With the lighting in the room, a yellow reflective object was one of the harder targets to get consistent performance from.

Owen

I felt very accomplished in this lab, we were able to get decent results in the end despite having flakier masking. That being said, the masking of a yellow object was very frustrating. It was a real knife's edge between the HSV thresholds and our erosions/dilations to get an even half decent image. Even when we got a good image, it would often fail when the target was not on screen, getting caught out by wood or other tan objects.

Srean

I think this lab allows me to understand how we can use real-time color-based object tracking. It allows me to understand the concept of converting images from RGB to HSV and isolate the colors under different lighting conditions. This lab also gave me a perspective on how to apply dilating and eroding to clean up the images and how to identify the biggest object depending on its color and how we can use the centroid of the object as a point signaling the robot when to stop.

Rowena

Being able to use the thresholds in order to have Ferb detect objects was both challenging and fun. The biggest issue we ran into was with the initial set up of the thresholds and sliders and then, figuring out how to get Ferb to move. Overall, this was a fun lab and Ferb was able to detect the yellow beach ball.

Divya

This was one of the most challenging labs so far. It involved us understanding thresholds at a deeper level. It was very interesting to get the sliders to work with our image requirements and ensure that we could see the image in its entirety. One of the hardest parts of the lab was ensuring that the Ferb would turn and stop in the direction of the yellow beach ball. Understanding how to turn Ferb left and right was crucial to figuring this out.

Code

Included in the .zip file.

Image of Software and Hardware