

## Daily Log

### Tuesday October 15

Tried running my partners code on my computer, but it wouldn't work. Kept getting "ValueError: too many values to unpack (expected 2)", even though I checked the method description, and the number of outputs is the same. Researched the error, but couldn't find anything.

### Thursday October 17

Fixed the problem, since there was actually an extra array of 0's at the beginning when the method was returning values. Got rid of that, and the program ran. I took a picture of a cube on a table and uploaded it to the program to see what it outputted, and it was a good output of the edges

### Monday October 21

I looked at the program in more depth, analyzing what each of the variables meant in relation to the coordinate plane so that I can tweak it for my purposes.

### Tuesday October 22

I tested the program for more pictures of a cube, online and taken in real life. I thought about how I would identify the colors of the center squares based on the edge detected image and the original image. I first decided to determine the coordinates of a bunch of points in the visible center squares. Then I will average the colors of those points in the original image to determine which color its closest to. I started programming it.

### Thursday October 24

Determining points inside the center squares of the edge detected image turned out to be harder than I thought. I researched other methods of determining the colors of squares on a cube. As I read through a few papers, and watched a few videos, I came across OpenCV shape detection, which should be helpful.

## Timeline

Date	Goal	Met
Today minus 2 weeks	Finish the fully working cube GUI with all the moves, and begin coding edge detection to eventually determine which frames of the video the cube are in the shape of a cube (not in the middle of turning),	Yes, finished the fully working cube GUI and started researching and coding edge detection.
Today minus 1 week	Upload a sample image to program, and be able to get a picture of just the edges on the image. Possibly, zone out everything but the cube outline and the lines separating the squares.	Yes, can clearly see the edge detected image. Might edit to make it more clear if needed.
Today	Identify the colors of the centers of the squares in the visible edge detected image. Use this to determine the orientation of the cube in the image.	No, turned out to be harder than I expected. It's hard to determine the colors of center squares.
Today plus 1 week	Identify the coordinates of many points in the centers of the squares in the visible edge detected image.	
Today plus 2 weeks	Use this to determine the color of the center squares orientation of the cube in the image.	

## Reflection

The past two weeks, I got my partners edge detection cube to work on my laptop. I tested it with multiple images of cubes: real life and online and it seems to work pretty well. Then I had to identify the colors of the cubes. Originally, I thought it wouldn't be as hard, but after trying it, it's much harder than I expected. I tried different things, did some more research, and came across OpenCV Shape detection. This should help in identifying a bunch of points in the center squares of each visible face. Then, I can average the colors of these coordinates in the original image to figure out the colors of the center squares, and eventually determine the orientation of the cube.

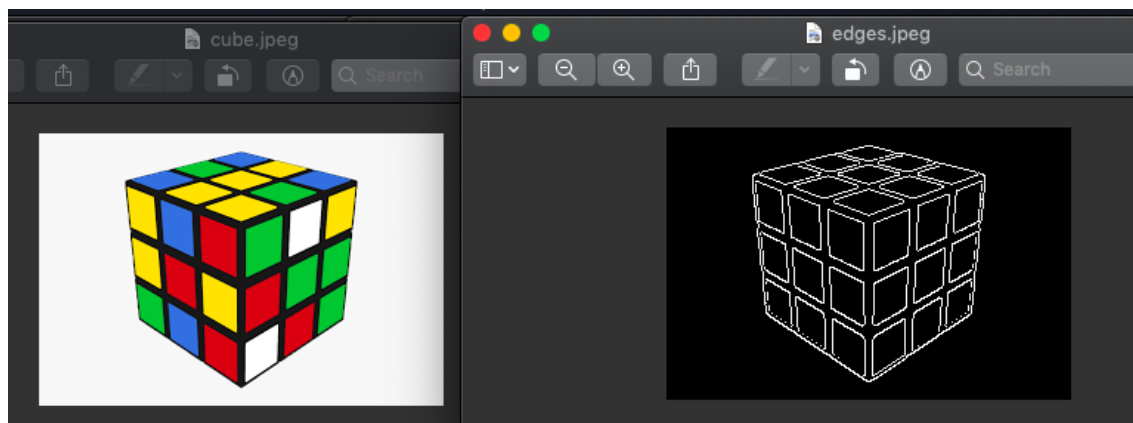


Figure 1: An image of the edge detection program output