



COMPUTER VISION, OBSTACLE AVOIDANCE, AND CONTOURING ON RASPBERRY PI AND OPENCV

Bonnie Akhavan – UMD Aerospace Engineering

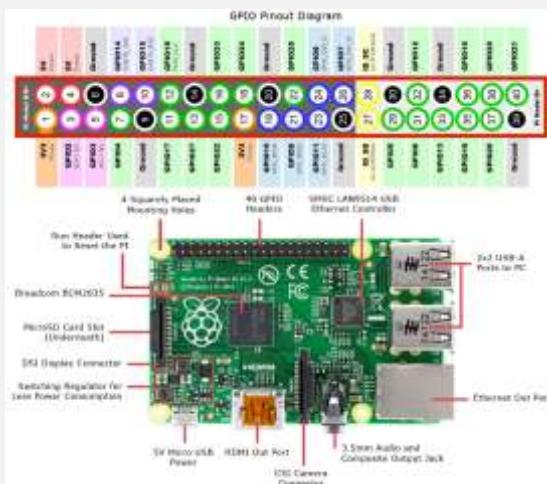
LIFE AS AN INTERN

- My internship: July 5th – August 30th at UMD Aerospace Engineering
- My project: analyzing different methods of computer vision for various tasks.
- My role: a leader in my own project!



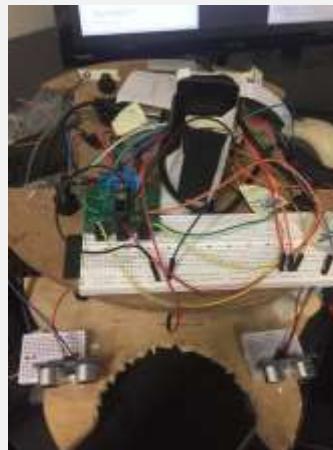
WHAT ARE RASPBERRY PI AND OPENCV?

- A Raspberry Pi is essentially a very bare-bones computer. Its most unique features are the GPIO, Power, and Ground pins, which allow for customization.
- OpenCV is a programming library specifically developed to aid in computer vision. It gives an extended list of commands to use. Many of these extra functions are algorithms or shortcuts in computer vision codes.



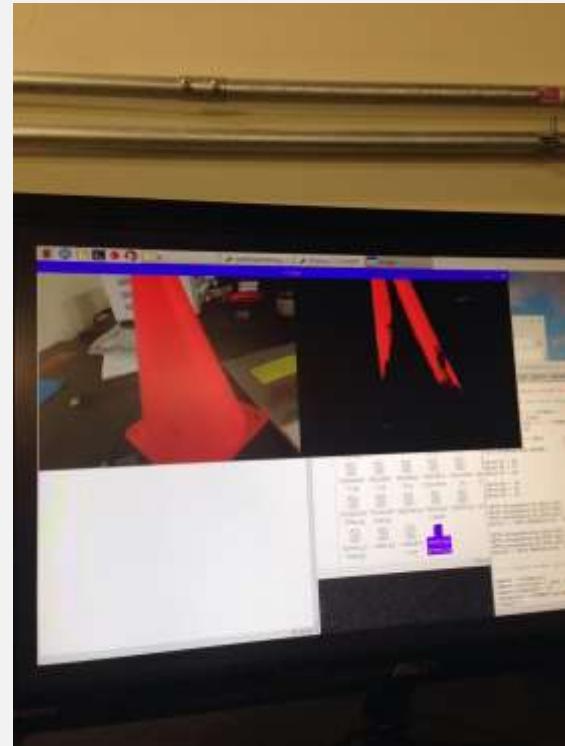
ASSEMBLY

- Terms to remember: breadboards, GPIO pins, Power pins, Ground pins, resistors, and accessories.
- Our goal was to connect everything to the Raspberry Pi and the power source in a way that was stable and correct. Sometimes adding external features was easier said than done!
- Final Result:



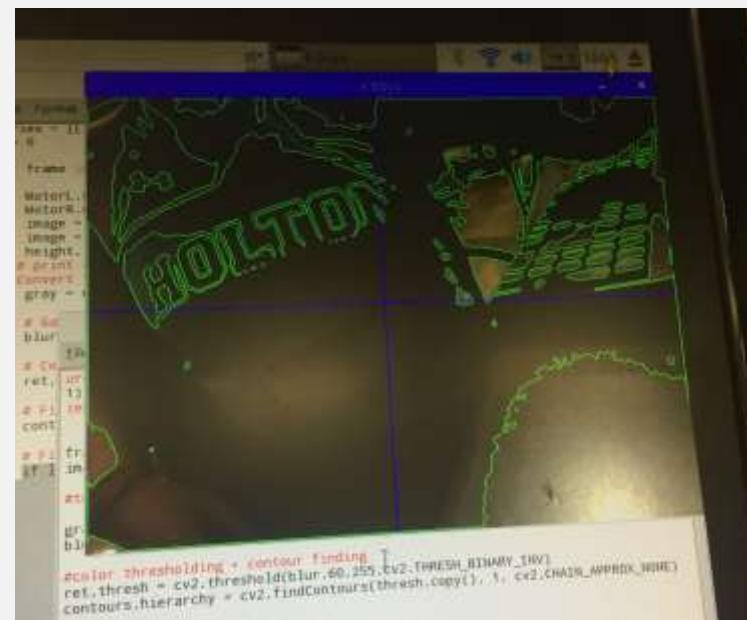
BASIC PROGRAMS

- Color Detection – Using continuous capture and color boundaries, we detected if there was an orange object in the feed, and took a picture of the frame.
- Key Controlled Rover – We created a manual override system to check that all the parts of the rover were working, and that it was possible to access the Raspberry Pi's terminal from wifi
- Ultrasonic Maze Solver – We used 2 ultrasonic sensors to estimate distances from various objects. These sensors were attached to the rover, and their feedback was used to navigate a maze.



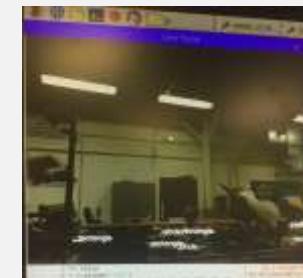
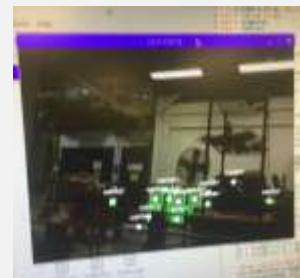
WHAT IS A CONTOUR?

- In OpenCV, contours are a selection of points defined by different color, shade or greyscale boundary. These points are connected to form a closed shape, which can then be analyzed using contour moments.
- Steps to a good contour:
 - Original Image
 - Greyscale
 - Blur
 - FindContours()/DrawContours()



ADVANCED PROGRAMS

- Line Tracking Robot – We used contours formed by color boundaries to follow a colored line. To track the desired direction, we analyzed the pixels in each frame and then moved the robot into an open space.
- Contour Analysis – We analyzed contours both on binary images and in live feed to find the best way to take a contour and use contour moments for analysis. The two parameters I screened for were the center and vertices of each contour.



LESSONS LEARNED

- Learning to use a programming library such as OpenCV is very hard, especially since there is no clear application guide.
- A lot of times we had code errors, but sometimes the more interesting issues were design flaws or mechanical failures that we had to address.
- Self motivation is a huge part of being in an internship environment. Especially in my lab, research was very independent.
- Lithium Polymer batteries are dangerous and unpredictable! So is using a resistor incorrectly.

CONCLUSION

- Computer vision was much more convenient for most of our programs, even if it was more complex and time consuming.
 - Today we see applications of this conclusion in drones as optical flow and mathematical relationships
- My next project would have entailed some kind of facial recognition using contours and shapes.

ACKNOWLEDGMENTS

First, I would like to thank my PI, Dr. Xu, for giving me the opportunity to use her lab and for taking me under her wing during my internship.

Additionally, without my grad student supervisors, Biko and Chris, I don't think our group would have come as far as it did in 8 weeks.

Finally, I also would like to thank my fellow interns, Marissa He and Patrick Wang, who worked with me on this project.

REFERENCES

- “About .” *OpenCV*, OpenCV Team, opencv.org/about.html.
- Marshall, Gary, and Nick Peers. “Raspberry Pi: Everything You Need to Know.” *TechRadar*, TechRadar, 15 Dec. 2015, www.techradar.com/news/computing/pc/raspberry-pi-everything-you-need-to-know-1069241.
- “Optical Flow.” Centeye , Centeye, www.centeye.com/technology/optical-flow/.