Nathan Penfield
14 July 2025
FURSCA, Summer 2025

End of Summer Report

Introduction

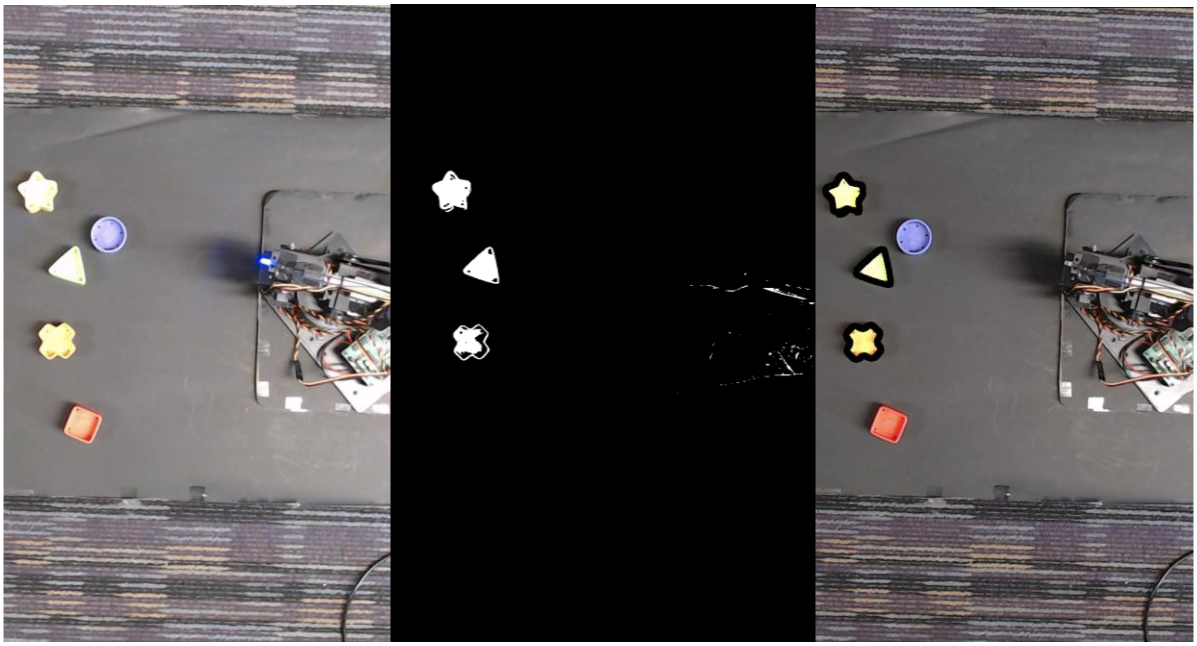
My name is Nathan Penfield, and I am a rising sophomore at Albion College. My project for the summer was to program a shape-sorting robot. The goal of this project was to program the robot arm to play a *Fisher-Price* shape-sorting toy. To do this, a webcam was used to recognize the blocks. Then the computer would calculate the position the robot would need to move to. This project required me to learn and implement computer vision and robot kinematics. This project is important because artificial intelligence and robotics is becoming more and more common in society. Automated robotics specifically plays a major role in many industries and is rising quickly in others. This project is my first step into getting experience into the field of automated robotics.

In this image to the left you can see my setup for the summer. The shape toy and robot were on a table top on the ground. Attached to the table is a webcam that overlooks the whole scene. This webcam is what took the pictures used in the computer vision discussed later.

Methodology

The implementation or methodology for my project involves three main components: Computer vision, robot kinematics, and communication.

The first step in this project was to develop a computer vision system to identify the blocks and their location. Computer vision can be defined as a field of computer science that enables computers to analyze and interpret visual data. In layman's terms, computer vision is any task involving computers "seeing" and making decisions based on images or video. For my project, I used a webcam looking down on the robot and shapes to give a top view. As seen in the images below from left to right, using a color range the program would convert the image to a binary image or black and white. From that image I used contour detection and approximation to recognize each of the blocks. I also relied on the size of the contour to ignore noise. It proved challenging to get the system consistent because of changes in lighting and not perfectly accurate contour detection, but eventually, using all of these techniques, I got a system that worked.

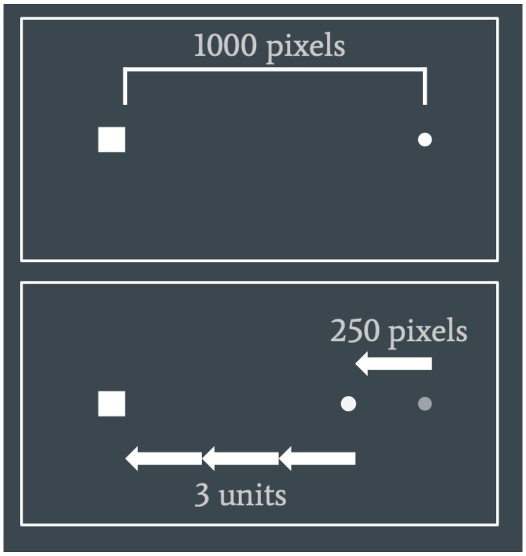


The other half of the computer vision was to recognize the position of the robot. At first, I attached a red eraser above the gripper on the robot that the computer would recognize in the image to find the robot's position. To make the project more professional and cleaner, I used a blue LED light instead to be detected by the vision system. Detecting the LED light was simpler than previously detecting the blocks. However, powering the LED light was more difficult and took longer than expected. I was originally trying to power the light from the robot's servo controller board. To do this I had to learn more about how the board works which was a good learning experience. Eventually, I decided to use a battery to power the LED light which was more simple. Once I got the LED working, the computer would recognize the LED light the same way as the blocks. At this point the computer vision system was working and would recognize all five shapes and the robot in the image.

The next step was to program the robot to move wherever it was needed. To do this, I used inverse kinematics. Inverse kinematics is the process of computing the angles needed at the robot joints to reach a given position. Basically, if the robot needed to move up 2 inches this would determine how each joint would need to move to accomplish this. The kinematics required complex trigonometry equations to compute the angles. To find these equations, I used an article from Dr. Rainer Hessmer, “Kinematics for Lynxmotion Robot Arm.” This article explained how to compute the equations for two of the five joints and the rest I computed on my own. Some joints were easier to compute than others. One of the more difficult equations is shown below.

$$atan2(\pm\sqrt{1-(\frac{x^2+y^2-l^2-L^2}{2l^2L^2})^2}, \frac{x^2+y^2-l^2-L^2}{2l^2L^2})$$

The last step of this project was to write the main code that would combine the two systems previously explained. This step was the most time-consuming because I often had to return and change the code I had already completed. This “main” program would merge the real-life position of the robot with the position of the blocks and LED light on the image to pick up and move the blocks. The implementation that I used, visualized in the diagram on the right, would first take a picture and then move the robot one inch in either the y or x direction. Then the computer vision would take another picture and determine how far the robot moved. Once this was done the program would calculate how much further the robot needed to move to reach the block. This process would be repeated until the robot was on top of the block.



I used this implementation to move the robot rather than calibrating the camera because the robot would not always move exactly as the kinematics wanted it to. Basically this implementation allowed the computer to adjust to errors in the robots movement. To get the robot to pick up the blocks, the LED light would need to be on the side of the block on the image. This is because the gripper on the robot was too small to grab the whole block. This wasn’t a huge problem but it made the system more complicated. Depending on how far the block was from the camera, the position of the LED light relative to the block on the image would need to be different. This has taken weeks to get working and is still not completely consistent. Another big challenge is the rotation of the blocks to fit in the container which I am still working to improve. The key to this is to always pick up the blocks in the same position no matter the rotation of the block. To get around this I would rotate the blocks intentionally when placing them. I did get a system that worked for the triangle and square because they had straight edges, but this system was not completed at the time of the video I have uploaded.

Results

The current version of the robot at the end of my FURSCA project does work but is not completely consistent. I have uploaded a video of the robot to Youtube and put my code on github. Both of these links can be found in the works cited section of this report. The robot can recognize each block and pick it up, somewhat consistently. Once the robot grabs the block, it is hard coded to fit the block into the container. This means that the act of placing the block in the container is not automated. The automated part of this project is recognizing the shapes and picking them up. All in all, I would say this project was successful in accomplishing my goal of programming a shape sorting robot.

This project could be extended to make the placing of the blocks in the container automated as well as improving its ability to grab blocks at all rotations. To do this the computer vision would need to recognize the holes on the container and move the arm to those positions similar to my project.

Conclusion

My participation in FURSCA with this project was massively beneficial for a future career in computer science. As a rising sophomore I didn't have much, if any, background in computer vision or robotics so this project was perfect for me to learn these skills. Implementing a real robot was a great experience for the practical use of skills learned in the classroom like computer vision. Although playing with a shape sorting toy may not seem very important, the practical and technical skills I learned could be extended to build other robots for more serious implications like automotive manufacturing, robot-assisted surgery, and much more.

I am looking forward to presenting my work at the Elkin Isaac Symposium. Thank you to my advisor, Dr. Mauricio Marengoni, who advised me throughout this summer project. I am extremely grateful for this opportunity that was given to me by the FURSCA program and the donors who make this program possible.

Works Cited

Hessmer, R. (2009, October). Kinematics for Lynxmotion Robot Arm. http://www.hessmer.org/uploads/RobotArm/Inverse%20Kinematics%20for%20Robot%20Arm.pdf

Saeed, O. (2024). *Building and implementing a chess robot*. Foundation for Undergraduate Research, Scholarships, and Creative Activity (FURSCA) Summer Research Program, Albion College. https://www.albion.edu/wp-content/uploads/2024/10/Omar.pdf

Chowdhary, C. L., Alazab, M., Chaudhary, A., Hakak, S., & Gadekallu, T. R. (2021). *Computer Vision and Recognition Systems Using Machine and Deep Learning Approaches: Fundamentals, Technologies and Applications.* Institution of Engineering & Technology. https://doi.org/10.1049/PBPC042E

A video of the robot working can be accessed at the following link https://youtu.be/JSMccHaAj6Y?si=tQosXE4PVwqUWXQ_

My code can be viewed on github at the following link https://github.com/nathanPenfield/Shape-Sorting-Robot/tree/main