Rubik Cube Solving Robot

Initial Project Document

Daniel Truesdell
Electrical Engineering

Corey Holsey
Computer Engineering

Tony Verbano
Computer Engineering

January 29, 2016

1 Project Description

The motivation behind this project is our fascination with the Rubik's Cube puzzle and the process of solving it. Our goal is to build a small robot capable of solving a Rubik's Cube, which is a task that many humans cannot perform. After a Rubik Cube is given to the robot, it will determine the locations of all the colored tiles on the Rubik's Cube and process this information into a matrix. Using this information, the robot will run the color patterns through a series of algorithms to determine how the cube needs to be manipulated in order to be solved. Then, the robot will perform these manipulations in order to match the correct colors on every side.

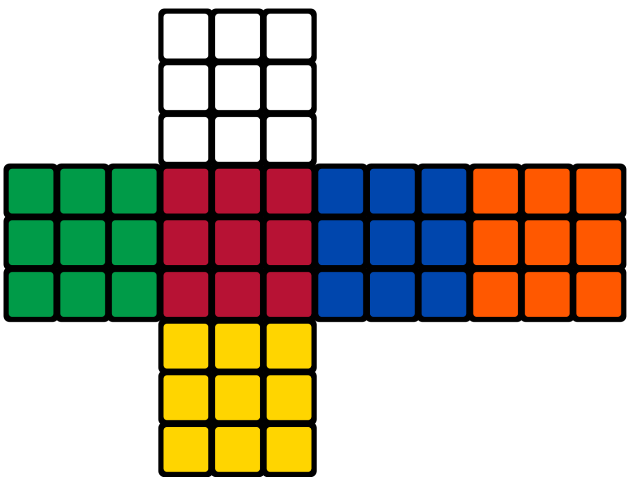
Computer Vision

One of our main goals for our project is using computer vision to look at all sides of the cube and determining the position of the colors in matrices. With this technique we can use an algorithm coding to solve the cube and therefore allow the robot manipulate the cube into its correct color matrices. How we have determined to start the computer vision process is to take a camera and attach it to our raspberry pi. After it is attached we will then position it on the top of the frame looking down at the center of the bottom clamp. When initiated the camera will take a picture of the top of the cube. From there the robot maneuver the cube to an alter side. It will then take another picture of the side it is looking at and saving it into a data bank. It will continue doing this until that camera has taken pictures of all the 3 other sides to the left. Lastly it would flip the cube upside down to determine what the bottom cube's color matrix will be. After all these pictures are taken and save to the data bank, the system will build a 2D form of the cube. This 2D form of the cube will be the basic layout of the whole cube and be used to determine the algorithm process later on discussed in our description.

One issue we look to overcome in the computer vision process is when the cube is being moved and manipulated the other sides will also be rotating and therefore ever picture may not be lined up with how the original cube may look. We look to overcome this part of the process by keeping track of the locations of the cube and rotating pictures where we know the cube will not match up. We also have decided that the easy way to design the layout for the cube in the 2D plane is if each color has its own number or letter in a matrix. That way the cube can solve for each number or letter to be the same on each side making it easier to work with and determine.

Solving Methodology

After gathering the six different matrices from the camera algorithms will be implemented. The algorithms utilized will be the key to solving it. The Rubik's Cube has a couple common way for it to be solved. The most adapted way is the Jessica Fridrich method in which we are inspired to utilize. This method was adapted in the 1980's and is considered the "layered" method. This method takes a scrambled Cube and solves for a cross-section on one arbitrary face. A cross-section is when the top, left, right, and bottom of a face matches the center cube of the same face. This cross-section is the beginning point in which several algorithms will begin to be used. After this cross-section is found the corners are next to be placed on this face. A specific algorithm is used to place each of the four remaining corners of this face in the correct place. Now that one face is solved it is placed face-down towards the ground.



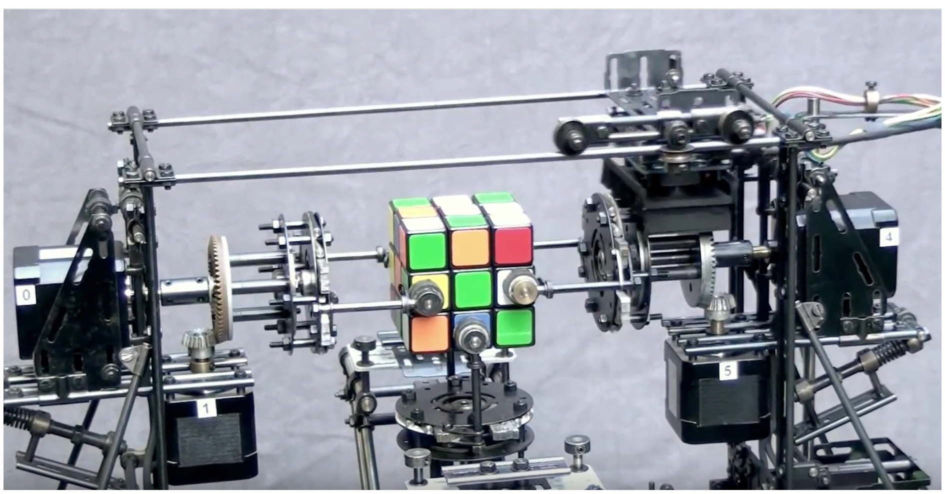
Cube Layout

Now the second layer, being the middle row, is to be solved. There will be a maximum of four cubes to solve for in this layer and two specific algorithms are used to solve it. One algorithm is used to place cube to the left of the center cube and the other is to place cubes to right of the center cube. Next, the top layer, or top row, is to be solved for. To solve the top layer an algorithm will be used to place a cube on the top and left of the cube in the center that are of the same color. Then an algorithm will be used to place a cube on the left and right of the center cube of the same color. Finally, a cross-section will be made on the top layer of the cube. Once this cross-section is made the cubes in the cross section must be placed in the correct place by implementing an algorithm. Once this is done then the corners of the cross-section are placed in the correct position, but can be in the incorrect orientation while using this algorithm. Once they are the correct position an algorithm is used to finally completely solve the Rubik's Cube.

Those are the directions to solving a physical Rubik's Cube via the Jessica Fredrich method however this project entails us to solve the Cube in a series of 2D matrices in code. There are 6 different 2D matrices that will be identified as a Face with each having a name corresponding to: Front, Back, Left, Right, Top, and Bottom. The algorithm's for solving the Cube would be the same, but referenced differently. For example, the Cube below would be similar to the six 2D matrices we would be looking to creating utilizing the camera. For simplicity the Green will be referenced as Top, the Red will be Right, the Blue will be Bottom, the Orange will be Left, the White will be Back, and the Yellow will be Front. Now when solving for the initial cross-section for the Top (Green) we will have to also realize the Front, Back, and Left are all connected to that cross-section even though it is not visually obvious. So the challenge lies in keeping all Faces properly connected and then tweaking the algorithm for Robotic arms. The last part is determining which ways to turn the robotic arms utilizing the algorithms. The solving of the Cube will be the same because the algorithms are what solve the cube and not the layout.

Mechanical Structure

The physical structure for the robot will need to be designed to accomodate the Rubik's Cube and robotic arms. We hope to design and prototype this structure with CAD and 3D printing since this will reduce construction costs as well as prototyping time. Modern Rubik's Cube solving robots that hold world records for fastest solving time rely on drilling holes into the center tiles of each face in order to quickly manipulate each side of the cube. We wish to manipulate the cube without physically altering it. Naturally, this means we are not competing for a speed record. We want our robot to function accurately but we prefer to take a simplistic approach by minimizing the complexity of the structure.



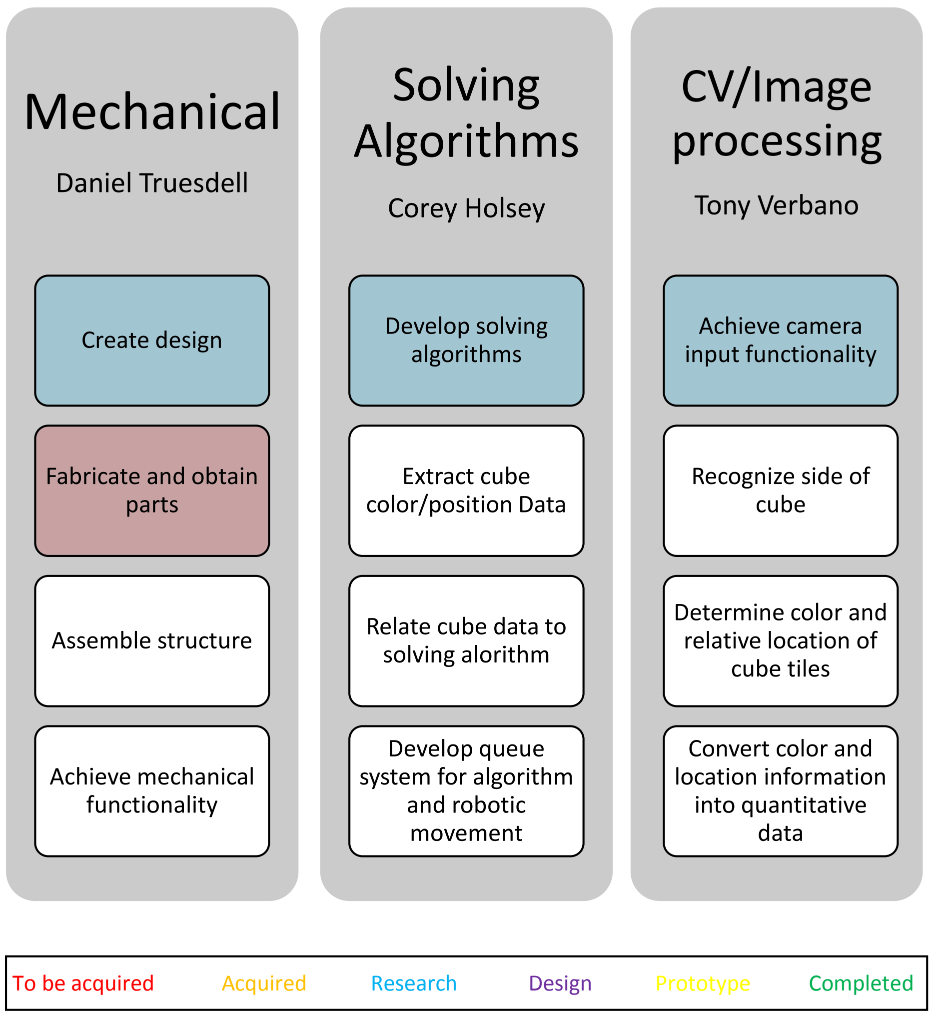
Inspiration for mechanical structure

2 Requirements and Specifications

- The robot should have a fully functional camera.
- The camera should be able to capture all sides of the cube
- The camera should be able to identify the position of the colors in the matrix.
- The code should be able to take in the color matrices identified by the camera.
- The code should be able to determine which algorithm would solve the color matrices.
- The robot will have multiple rotating and clamping (3?) arms to manipulate the cube
- The robot should be powered by a wall outlet.
- The algorithm should be ran through a raspberry pi.
- The primary performance specification is that the robot should solve the cube.
- The robot should solve the cube 95% of the time with full accuracy.
- The robot should fit on a tabletop.
- The robot should only scan the cube once before manipulating the cube.

3 Block Diagram

This project incorporates multiple systems into a single end product. We have broken these systems down into three main categories: Mechanical structure, solving algorithms, and image processing. As long as proper planning is used, each of these systems can be developed independently, e.g. the design and construction of the mechanical structure doesn't depend very much on the software being run on the Raspberry Pi, and the theory behind the solving algorithms doesnt depend on how the computer vision is implemented. Only after these systems reach a certain level of development can they be integrated. The block diagram below highlights the key development milestones for each of the systems.



Key Implementation Steps

4 Project Budget

Camera	$50
Mechanical Components - unknown (Access to a 3d printer)	$15
Servos	$30
Rasberry Pi	$120
Frame - unknown (Access to a 3D printer)	$30
Miscellaneous electronic parts	$10
Total Cost	$255

5 Project Milestones

We have identified several milestones for the overall project that lead to its completion. All of these milestones require cooperation of at least two of the three systems outlined in the block diagram.

- Rubik's Cube algorithm program
- Ability to recognize side of Cube
- Ability to recognize 6 sides of the Cube
- Ability to determine which side of Cube being recognized (Front, back, left, right, bottom, top)
- Building of the robot
- Motion within the robotS
- Ability to turn cube sides
- Ability to move cube to recognize all sides
- Programs ability to determine colors
- Ability to solve Rubik's Cube