Name(s)

Evan M. Gates

Project Number

S1208

Project Title

Software and Hardware Implementation of Rubik's Cube Solving Algorithms

Abstract

Objectives/Goals
The goal was to write the software and create the hardware required to autonomously solve a Rubik's cube.

Methods/Materials
The physical Rubik's cube solver was made primarily out of Plexiglas. A stepper motor is mounted on each of the six sides with a shaft that goes into the center of one face of the cube. In this manner all faces of the cube can be turned. All of the required code was written in C. An implementation of the Thistlethwaite algorithm was written to solve the Rubik's cube. This algorithm uses an iterative deepening search as well as pruning trees to search for moves to solve the cube. It works through four nested subgroups of the cube, restricting moves and solving certain aspects along the way. The state of the cube was originally kept in a file after having a scrambling program scramble it. The use of cameras has recently been implemented to read in the state of the cube. A color recognition scheme was devised, discerning the different colors by comparing RGB values. After a string of moves is produced, either by the solver, the scrambler, or by manual input, it is passed to a program which creates stepper motor commands. The face turns of the cube are translated into turns of the stepper motors, and output through the COM port at a given pace.

Results
There was some trouble stopping the solver from jamming. This was fixed by making all turns clockwise and slightly overshooting the goal so that the cube would self correct if slightly misaligned. Creating consistent lighting for the color recognition to get consistent results was difficult. Lights were added in an attempt to light each face consistently. The solving algorithm solved the cube efficiently and executed quickly.

Conclusions/Discussion
Although the goal was reached, the project can still be improved upon. The main area that could use improvement is the color recognition algorithm. The implementation of neural networks would produce more accurate results. Neural networks could also be used to determine the sampling areas as opposed to manually inputting where the center of the cube is. Another possible extension would be the use of a more efficient solving algorithm. The current algorithm is known as the Thistlethwaite algorithm. Another popular algorithm is the Kociemba algorithm. This algorithm solves the cube on average in ten less moves than the Thistlethwaite algorithm.

Summary Statement
Creation of both the software and hardware needed to autonomously solve a Rubik's cube.

Help Received
Father helped format report; Mother helped with display board.