

Cubr: Cube Puzzle Solver

18500 S19 Team D6

Project Proposal

JT Acheron, Lily Chen, Sam Fazel-Sarjui

Background

- 3x3x3 Rubik's Cube
- 43,252,003,274,489,856,000 unique cube states
- World record time set by Feliks Zemdegs: 4.22 s
- 3 pieces
 - Center, Edge, Corner

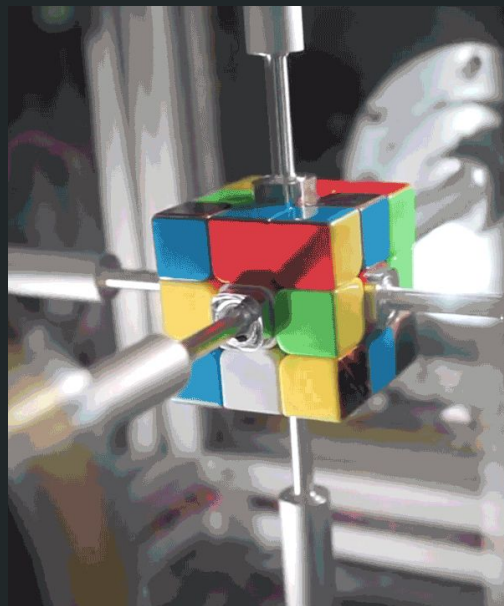


Use Case

- Uses CV to map the faces of the cube (signals & systems)
- Uses software to find the sequence of moves to solve the cube (software)
- Uses an Arduino to interface our solution with stepper motors (hardware)

Hardware and Software Overview

- Webcam
- OpenCV
- Arduino
- 6 stepper motors
- 6 stepper motor drivers
- Power Supply
- 3D printed coupling arms
- Basic breadboard
- 3D printed housing
- 3x3x3 Speedcube



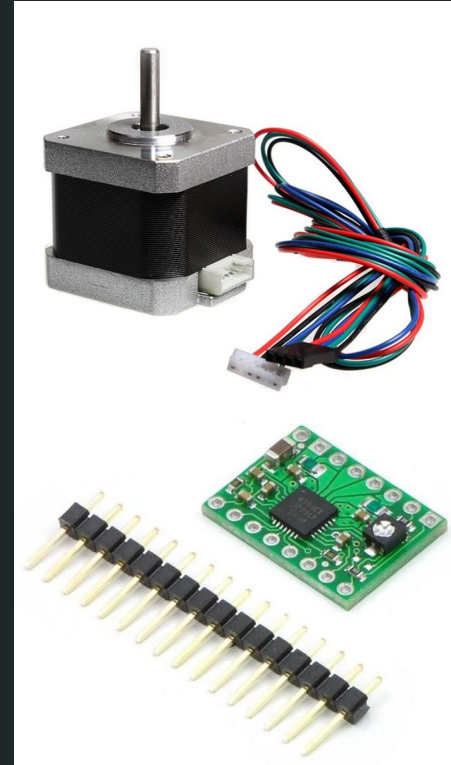
Processing and Execution

- Mac:
 - Cube state detection
 - Logitech - C920 Pro Webcam
 - Webcam compatible with OpenCV
 - Written in C++ or Python
 - Cube solving algorithm
 - Written in Python
- Arduino Uno Rev3:
 - Arduino receives solution string
 - Master-slaves setup to communicate to the 6 stepper drivers
 - Written in C



Stepper Motors

- NEMA-17 Stepper motor
 - 200 steps/rev
 - 12V 350mA
 - \$14
- A4988 Stepper Driver
 - Operates from 8V-35V
 - \$5
- 12 Volt DC wall adapter



Physical Infrastructure & Misc.

- 3D printed housing
- 3D printed cube coupling arms
- Breadboard
- Wires



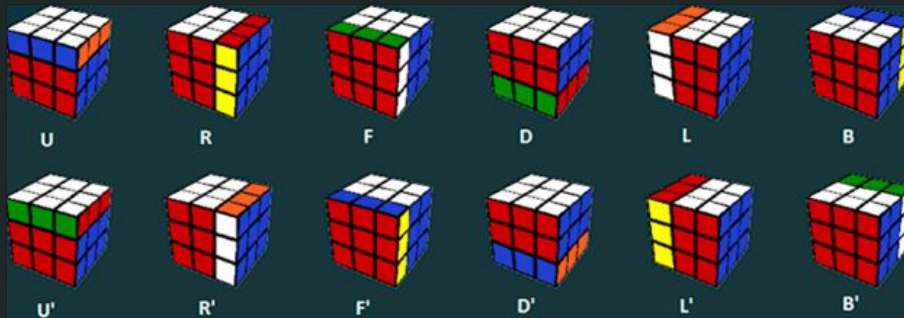
Solution: Solving Algorithms

- Intuitive (Beginner's method)
 - Cube is broken down into states and solved with a set of basic algorithms
- Stretch goal: Non-intuitive (highly efficient)
 - The most optimal solution path
 - Two-phase Algorithm
 - God's Number: 20 Algorithm
- Written in Python



Solution: Physical Execution

- Arduino receives solution string using cube notation (R R' U U' etc)
- Arduino will communicate to each stepper driver individually to drive it's motor
 - 18 possible moves
 - Clockwise or counterclockwise
 - 90 degree or 180 degree turn
 - Motors must be timed and tuned: only one move/turn at a time



Testing, Verification & Metrics

- Cube state detection
 - Verify how accurately we scan the cube using OpenCV
- Algorithm efficiency
 - Number of cube movements to solve the cube
 - Was a solution found?
- Stepper motor precision & timing
 - Verify correct movements are made for any given input
 - Use metrics to record how fast a movement takes
- Is the cube solved?

