Autonomous Rubik's Cube Solver Using Image Processing

Harshad Sawhney
IIT Kanpur

Sakshi Sinha
IIT Kanpur

Anurag Lohia
IIT Kanpur

Prashant Jalan
IIT Kanpur

Priyanka Harlalka
IIT Kanpur

Abstract

Rubik's cube is a 3-D mechanical puzzle in which a pivot mechanism enables each face to turn independently, thus mixing up the colours. The beauty of the cube is that each piece of the cube has only one correct orientation. For the puzzle to be solved, each face must be returned to one colour configuration. The aim of this project is to build an autonomous robot which can solve a Rubik's cube. It uses a gripper design mechanism to hold the cube from four faces which can provide all possible rotations in the cube. The robot is controlled by a microcontroller which is the brain of it and provides commands to the motors to solve the cube . In order to recognise colours on each faces of the cube, Image processing was used. For image processing we used OpenCV. OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision. Dev C++ was used as development environment. Dev-C++ is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language.

1. Introduction

Rubik's cube solver is a robot which requires perfection. As has been known Rubik's cube is one of the most challenging and most famous puzzles of all time .Our aim was to develop a completely autonomous robot which is capable to identify colours on the cube and solve it in minimum time and minimum number of steps.

Solving a Rubik's cube has three major parts. First is identifying the positions of different colours at different positions. Second is to develop a series of steps which can be used to solve the cube and third is to implement these steps on the cube to get the final result.

A camera captures each of the six faces of cube and then it is processed to separate the blob of each colours on each face which is then accounted for and an array of data is produced which contains location and colour of all the pieces. Then a solving algorithm is applied on it which gives the desired solution step. The beauty of this algorithm is that it gives a maximum of only 25 steps to solve the cube where each step comprises of one rotation of respective face. This string of data is then fed into microcontroller which the drives grippers that are attached to each of the motors. These grippers are used to hold the cube and provide it the desired motion.

This paper is organised as follows:
Section 2: Mechanical design.
Section 3: Electronics.
Section 4: Programming.
Section 5: Conclusion.
Section 6: Acknowledgement.
Section 7: References.

Figure 1: Rubik's cube solver photograph

2. Mechanical design

One of the most important parts of the robot was its mechanical design. The mechanical design would require a very high precision .Even the slightest error of few millimetres would result in failure. The mechanical design consists of a cuboidal box made of acrylic material open from one side .On each of the four side walls of the cuboid, One stepper motor is mounted with utmost precision so that motors at opposite faces has their shafts coaxial. On each of these motors a gripper has been mounted which is driven by servo motors .These grippers are the part of the robot which will be in contact with the cube and provides the cube, the necessary rotations. Using the combination of motion of these motors, the robot is able to provide rotation to all the six faces of the cube.

Before the actual fabrication of the required parts, designs of all the part were made on Autodesk Inventor and simulated. After ensuring that the design will work it has been fabricated at 4i Laboratory at IIT KANPUR.

2.1. Grippers

Grippers are like robotic arms which are used to hold the cube. They are hand like mechanism which is able to open and close and remain at any particular angular position as desired. Grippers are made up of different components which are attached to act as one body in which the two front hands are capable of rotating. It consists of two spur gear placed side by side to transfer motion in parallel direction from one hand to another. One of the gears is attached to the rotating head of the servo motor. Hence when servo motor rotates, both the gears rotates with equal angular velocity but with opposite direction which essentially give the motion of two fingers of a hand to hold any object. On each of the gears there are finger like object attached which acts as holding fingers. A threaded screw passes through the centre of each spur gear and the holding fingers which makes them act as one rigid body. This system is the covered by acrylic by three sides giving it a protective covering as well as providing a 3 dimensional rigid structure for the gripper. The rear side of the gripper has a hole which is placed at the geometric centre of the gripper mechanism so that when rotating the cube, it
rotates about its centre axis. Through this hole, shaft of stepper motor is attached to the gripper through bearings. This design makes it possible for the robot to hold the cube and give it proper rotations. The acrylic parts of the gripper are attached to each other using acrylic.

2.2 Motors

There are two kinds of motors used in this project-namely stepper motors and servo motors.

Servo motor is a rotatory actuator that is used for precise control of the angular positions. They are a special kind of motors that have the ability to rotate to a particular angle relative to its initial position. They are based on position feedback mechanism to control its motion and final position .Due to this mechanism the get locked at a particular angle through this feedback mechanism but they can maximum provide 180 degree rotation in one direction and its further rotation is prevented by a physical barrier. Thus they provide rotation through a particular angle. Servo motors can be easily operated and can be used to provide a particular angular motion ranging from o to 180 degrees. Therefore we can use this to control the angle to which the gripper should be opened or closed, hence are used for holding the cube.

The other kind of motors is the stepper motors Stepper motors moves a known angular motion for each pulse of power provide to it. Generally a stepper motor has a least count of 1.8 degrees for one pulse of power. Due to this property it provides precise rotations and hence is required to rotate the entire gripper by exact 90 degrees. These motors contain coils of electromagnet and convert electrical pulses to discrete mechanical movements. At any given moment of time two coils in the motor are provided power .there is a tendency of a permanent magnet to rotate until it comes into the orientation of the magnetic field applied to it. In stepper motor these coils are provided power in wave form due to which the permanent magnet bars in the motor moves from one orientation to another thus proving a continuous motion. Stopping the input of this electrical pulse at any time stops the motor hence controlling this time factor we can give precise angular motion to the motor. Due to this property of the stepper motor, it has been used to rotate grippers by exact 90 degrees in turn rotating the cube by 90 degrees. There are four such motors attached to four faces of the main frame which can provide the rotation of four faces at that particular instant. For the rotation of the other two faces namely the top and bottom faces, two opposite grippers can be rotated simultaneously converting the

top face to front face which can be rotated by one of the grippers by 90 degrees.

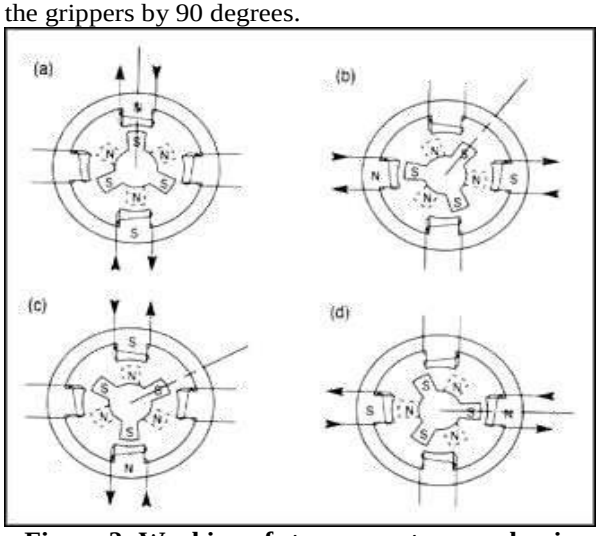

Figure 2: Working of stepper motors mechanism

3. Electronics

The electronic part basically contains a microcontroller which is the brain of the robot. The microcontroller used is "Arduino Mega" and the motors are connected to the pins of the microcontroller which gives commands to motors based on a code stored inside the chip or Atmega of the microcontroller. The stepper motors are connected through a L298 motor driver which acts as a switch. The stepper motors are 6 wired motors two of which are the 12V and ground terminals while the rest four wires correspond to the ends of the two electromagnets. The motor driver pins are connected to the microcontroller which provides appropriate instructions to it. They have PWM (pulse width modulation) pins which can be used to control the speed of stepper motors. The servo motors are 3 wired, two of which are ground and 5V while the third is the PWM which is used to control the speed of servo.

4. Programming

The programming consists of image processing, algorithm for finding the solution of the scrambled Rubik's cube and coding of microcontroller to direct the motors accordingly.

4.1. Image processing

Image processing is a technique to store the colours of any object in a format that can be processed through codes and programs. The images of the six faces of the cube are taken to obtain the solution or the steps which can solve the cube. This is done through a code to obtain the colours of each block in a matrix. The images are taken externally in a black box since the lighting conditions are very important to avoid errors and unnecessary calibrations to detect the colours of the cube correctly.

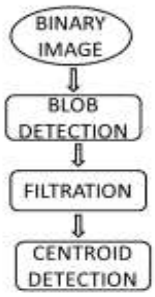

Figure 3: Flowchart depicting the process of image processing

4.1.1 SOFTWARE ARCHITECTURE. Image is being processed by the code of Dev C++.The main algorithm has been explained by the flowchart shown in Figure 3 . We are taking the images of all the faces of the cube. The sequence in which the image is taken is determined by the kociemba algorithm (section 4.2).After clicking the image; conversion of the captured image to a binary image is done. We are dealing with all the six colours separately. After that, blob detection is done followed by determination of the centroid of each blob. Sequence of colours on that face of the cube is obtained by sorting the centroids of the blobs. A string is generated by the sequence obtained for the Kociemba algorithm.

4.1.2 BINARY CONVERSION OF IMAGE. A 3-channel RGB image is captured by the camera. Each colour has its own three values (hue, saturation and value) which distinguish it from other colours. Using this property, a binary image can be generated for all the six colours of the cube. Binary image can be represented as a matrix where each pixel can have value either value 0 (black) or 1 (white) which is a 1-channel image.

4.1.3. Centroid Detection & Filtration. Blob detection is applied on the binary image of each colour. cvBlob library is provided by OpenCV which is used for detecting the blobs. Detecting the connected components of the binary image is the purpose served by this library. Area of each blob in pixels can be easily found and the background filtration can be done on this basis i.e. area greater than or less than the area of a blob is in the background and should be removed. A noise-free binary image of each colour is obtained along with its centroids. These centroids are sorted to get the location of each colour. A string of colours is generated in a proper format taken by the Kociemba algorithm.

4.2. Koceimba's algorithm

After finding the matrix that stores the colours of the individual blocks of the six faces of the cube, an algorithm are required which provides the steps that can solve the cube. Koceimba's algorithm is used to find this solution that is given to the microcontroller autonomously which further instructs the motors according to it and unscrambles the cube. This algorithm can solve the cube in maximum twenty-five steps and hence provides a good optimal solution for descrambling the cube.

4.3. Arduino coding

The Arduino is coded in Arduino interface and it provides rotation to each face of the cube. The code ensures that the grippers do not collide with each other at any point of time while unscrambling the cube. The solution calculated through the Koceimba's algorithm is converted into appropriate characters that are then passed to the Arduino through serial communication between the computer and Arduino which then commands the motors to solve the cube. The characters that are passed to the Arduino in the string are 'U', 'V', 'D', 'E', 'R', 'S', 'L', 'M', 'F', 'G', 'B', 'C'.

'U' denotes the rotation of first stepper clockwise while 'V' denotes it anticlockwise. 'D' denotes the rotation of second stepper clockwise while 'E' denotes it anticlockwise. Similarly 'R' and 'L' denotes the rotation of third and fourth stepper motor clockwise respectively while 'S' and 'M' denotes it anticlockwise. Since there are four motors the rotation of the left two faces requires that the entire cube is rotated by 90 degrees. Therefore for 'F' move the cube is rotated by third and fourth stepper motor simultaneously and then the first stepper is rotated by 90 degrees in the clockwise direction while in the anticlockwise direction for 'G' move. Similarly it is done for the rest two cases. A flowchart depicting the above algorithm is shown on the next page.

5. Conclusion

The method for solving the Rubik's cube autonomously is described in the paper. The image processing on appropriate calibrations can correctly determine the blobs of the cube and hence completely gives the initial state of the cube. Finally the use of Koceimba's algorithm gives the solution of the Rubik's cube which can be finally used to instruct the motors to solve the cube through serial communication.

6. Acknowledgement

We would like to acknowledge the efforts of Robotics Club of IIT Kanpur for providing appropriate guidance and facilities which helped in the successful completion of the project. We would also like to acknowledge Abhijit Verma, Ayush Varshney and Mohammed Farid Ahsan for their continuous support through the entire project. Finally, we would like to thank the staff members of 4i Laboratory of IIT Kanpur for the successful fabrication of mechanical design.

7. References

[1] Herbert Koceimba, "Koceimba's algorithm and Cube Explorer 5.00", 2011
[2] Thomas E. Kissell "Industrial Electronics", 2006
[3] Takashi, Kenjo, and Sugawara Akira, "Stepping motors and their microprocessor controls." (1994): 45-70.
[4] Gary Bradski and Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library".

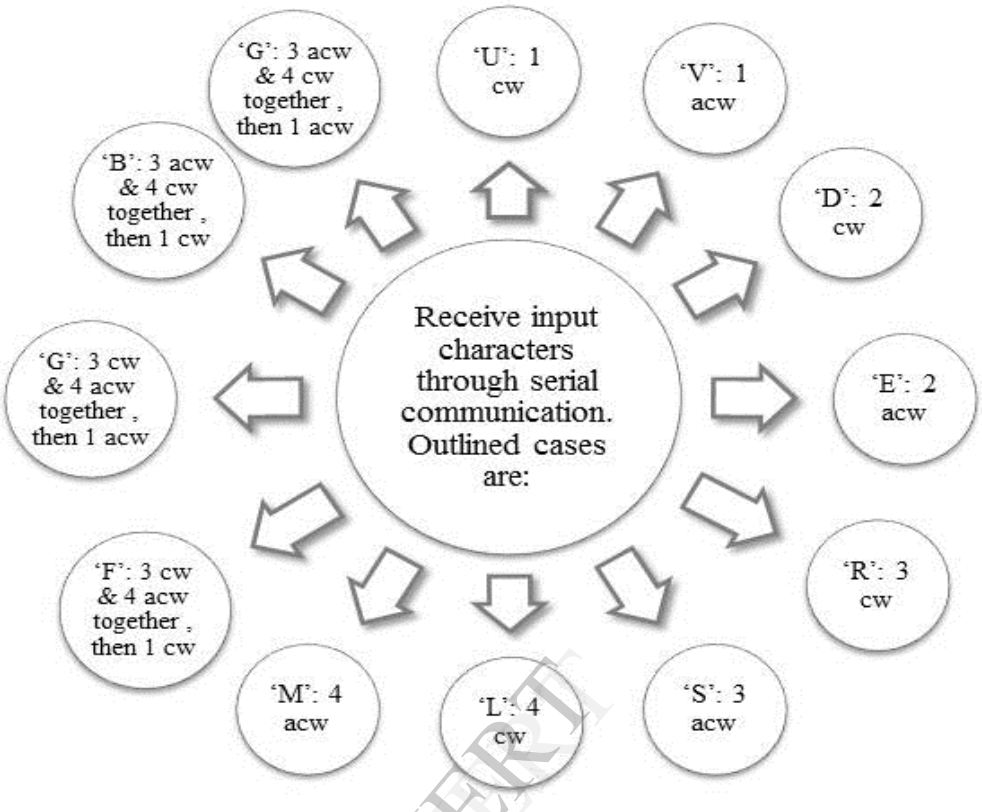

Figure 4: Flow chart for Arduino Coding
Note: cw-clockwise, acw- anticlockwise; 1, 2, 3, 4 denotes the stepper motors

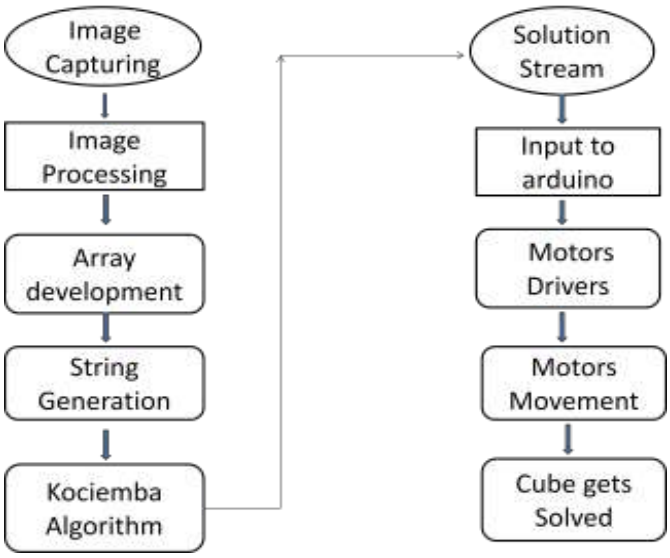

Figure 5: Flow chart for complete processing