Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140
Volume 14 Issue 01, 2025

Automatic Rubik's Cube Solver with Advanced Colour Detection

Prof. Shital Patrakar[1], Mr. Sadique Khan [2], Mr. Ashish Dadmal[3], Mr. Hanumant Narwade[4], Mrs. Seema Raut[5]

[1]Assistant Professor, Department of Computer Engineering, SCET, Nagpur, Maharashtra, India
[2, 3, 4, 5] UG Student, Department of Computer Engineering, SCET, Nagpur, Maharashtra, India
Email: [1]shitaldurudakar13@gmail.com, 7773937429
[2]sadiquekhan2309@gmail.com, 9923059611
[3]dadmalashish2002@gmail.com, 8010012637
[4]hanumantnarwademh22@gmail.com,7666944346
[5]raut271203@gmail.com, 9209376702

Peer Review Information

Submission: 05 Feb 2025
Revision: 17 Mar 2025
Acceptance: 18 April 2025

Keywords

Rubik's Cube
Robotics
Computer Vision
Artificial Intelligence
Kociemba Algorithm

Abstract

Rubik's cube is a 3-D mechanical puzzle in which a pivot mechanism enables each face to turn independently. Solving it manually requires a deep understanding of patterns and algorithms, but the advent of robotics has opened new possibilities for autonomous solutions. The aim of this project is to build an autonomous robot which can solve a Rubik's cube. This project proposes the development of a Rubik's Cube solver robot that integrates computer vision, robotics, and artificial intelligence to efficiently solve the puzzle. The system comprises three main subsystems: vision, computation, and actuation. The vision subsystem uses cameras and image processing techniques to capture and analyze the cube's state, while the computation subsystem employs advanced algorithms like the Kociemba algorithm to determine the optimal solving sequence. The actuation subsystem, consisting of a robotic arm and motors, executes the computed moves with precision. Currently, the project is in the implementation phase, with the mechanical design and assembly of the robot nearing completion. The vision subsystem has been tested using OpenCV for colour detection, and the Kociemba algorithm has been successfully integrated into the computation subsystem. The next steps involve fine-tuning the motor control system and conducting extensive testing to achieve a fully autonomous and efficient Rubik's Cube solver robot.

INTRODUCTION

The Rubik's Cube is one of the most popular and enduring puzzles in modern history, known for its simple design yet complex solutions. Solving the cube manually involves understanding intricate patterns and algorithms to manipulate the colored tiles into a single uniform configuration. While human "speedcubers" can solve the puzzle in mere seconds through sheer skill and practice, the advent of robotics has introduced an entirely new domain of exploration: developing machines that can solve the cube autonomously. A Rubik's Cube solver robot demonstrates the remarkable intersection of mechanical engineering, artificial intelligence, and computer vision.

Rubiks cube puzzle has continually been as a hot topic in intelligence competition for child/adult. While in service robot fields, efficient solution of Rubiks cube puzzle is a challenge for computer vision. A software scheme to solve Rubiks cube puzzle includes detection, color recognition and solve method of a randomly scramble cube. Rubiks cube puzzle can be also considered as a sequential manipulation problem for service robot. For example, optical time-of-flight pre-touch sensor are used for grasp Rubiks cube to achieve a high precise sequential manipulation.
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.

OBJECTIVES

The primary goal of this project is to design and develop an advanced robotic system capable of autonomously solving a Rubik's Cube in real-time using cutting-edge color detection and image processing techniques. The system will integrate high-precision robotics with efficient algorithms to analyze the cube’s current state, compute the optimal solution, and execute the required movements seamlessly. This project aims to bridge the gap between theoretical problem-solving and practical implementation by leveraging innovative technologies to create an efficient, reliable, and user-friendly solution for solving the Rubik's Cube under diverse conditions. Here is some key objectives are:

1. Efficiency Develop an Advanced Color Detection Mechanism: Implement a robust color
detection system to accurately identify the six colours of the Rubik's Cube, ensuring consistent performance under varying lighting conditions and color shades. Maintenance: Ensure solar panels operate at peak efficiency by keeping surfaces clean from dust and grime.
2. Design a Real-Time State Analysis System: Create a reliable algorithm to capture and analyze the current state of the Rubik's Cube in real-time by processing data from all six faces with minimal latency.
3. Integrate an Optimized Solving Algorithm: Implement a solving algorithm capable of calculating the most efficient solution for the cube’s state while minimizing the number of moves and ensuring quick execution.
4. Develop a High-Precision Robotic Manipulator: Engineer a mechanical system equipped with precise actuators to handle and rotate the Rubik's Cube accurately, ensuring smooth and error-free operations.
5. Ensure Seamless Hardware-Software Integration: Achieve a seamless connection between hardware components like cameras, sensors, and robotic arms with software modules for color detection, state analysis, and algorithm execution to ensure efficient real-time performance.

LITERATURE REVIEW

Year	Authors	Title	Description
2024	Gan Robot	Intelligent Cubing Pioneer, Gan Cube.	GAN Cube, a pioneer in intelligent cubing, revolutionizes the classic puzzle with cutting-edge technology. Known for their premium speed cubes, GAN combines precision engineering, advanced materials, and innovative features like adjustable tensions, magnetic positioning, and AI-driven tracking. The GAN Smart Cube series offers real-time performance analysis, tracking, and battle modes, enhancing the cubing experience for beginners and professionals alike. With sleek designs and smooth turning mechanisms, GAN cubes deliver speed, control, and consistency, making them a top choice for competitive cubes worldwide.
2023	J. Terra, Keras vs Tensorflow vs Pytorch.	Key Differences Among Deep Learning,	This provides a comparison of major deep learning frameworks, including **Keras, TensorFlow, and PyTorch**.

		Simplilearn, Aug 2023, Available	It explains their unique features, ease of use, performance, and best applications: **Keras**: A user-friendly, high-level API that simplifies deep learning model development. **TensorFlow**: A powerful and scalable framework widely used for production and deployment. **PyTorch**: A flexible, research-oriented framework known for its dynamic computation and ease of debugging.
2021	Dan, V., Harja, G., & Naşcu.	Advanced Rubik's Cube Algorithmic Solver. In *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*	It discusses various algorithmic approaches, including **heuristic methods, optimization techniques, and artificial intelligence-based solutions**, to improve solving speed and accuracy. The study also highlights the integration of these algorithms into robotic systems, contributing to advancements in **automation, robotics, and AI-driven puzzle-solving technologies**.
2020	Barucija, E., Akagic, A., Ribic, S., & Juric, Z	Approaches in solving Rubik's cube with Hardware-Software Co-design. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*	It discusses **algorithmic techniques** for efficient cube solving and their integration with **robotic or embedded systems** to enhance performance. The study highlights the advantages of **hardware-software co-design**, balancing computational efficiency with mechanical precision, and demonstrates practical implementations for automated cube-solving applications.
2019	Liu, S., Jiang, D., Feng, L., Wang, F., Feng, Z., Liu, X	Color Recognition for Rubik's Cube Robot.	The study **"Color Recognition for Rubik's Cube Robot"** focuses on the development of a **color detection system** for an automated **Rubik's Cube-solving robot**. It explores techniques for accurately identifying cube face colours using **computer vision, sensors, or machine learning algorithms**.

PROPOSED METHODOLOGY

To develop a Rubik's Cube solver robot, it is essential to understand the problem at hand and identify the components and systems necessary for its solution. The development process can be divided into three main subsystems: vision, computation, and actuation. Each of these subsystems involves specific hardware and software components, described in detail below:

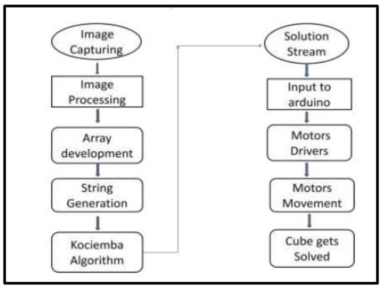

Fig 1: Block Diagram

1. **Vision Subsystem:**

- **Camera:** A high-resolution camera is required to capture images of the cube's faces. Multiple cameras or a single movable camera can be used to view all six faces of the cube.

- **Lighting:** Adequate lighting ensures that the images are free from shadows and color distortions.
- **Image Processing Software:** OpenCV, an open-source computer vision library, is commonly used for color detection and segmentation to identify the arrangement of colored tiles on the cube.

2. **Computation Subsystem:**

- **Microcontroller/Processor:** A microcontroller like Arduino or a single-board computer like Raspberry Pi processes the image data and computes the solution. Advanced systems may use GPUs for faster computation.
- **Algorithm Implementation:** The solution algorithm, such as the Kociemba algorithm or Thistlethwaite's algorithm, determines the sequence of moves required to solve the cube. These algorithms minimize the number of moves, optimizing both time and mechanical efficiency.

3. **Actuation Subsystem:**

- **Robotic Arm/Grippers:** Servo motors or stepper motors control robotic arms or grippers to manipulate the cube. These components must be capable of precise and rapid movements.
- **Motor Controllers:** Motor drivers or controllers regulate the power supplied to the motors, ensuring smooth operation.
- **Chassis and Frame:** A sturdy frame holds the cube and the robotic components in place.

Additional Components:

- **Power Supply:** A stable power source is required for all electronic and mechanical components.
- **Software Framework:** A programming environment such as Python or C++ integrates the vision, computation, and actuation systems.

PROJECT REQUIREMENTS

Hardware Requirements

1. Microcontroller/Processor
   A microcontroller or processor is the central unit responsible for executing commands, processing images, and controlling motors. It acts as the brain of the Rubik's Cube Solver robot. Arduino can be used for motor control if external processing is done on a separate computer.

Figure 1. Arduino UNO

2. Camera
   A high-resolution camera is used to capture images of all six faces of the Rubik's Cube. The camera must have good color accuracy to detect different cube colours precisely. A USB camera or a built-in camera module like the Raspberry Pi Camera Module is commonly used for this purpose.

Figure 2. USB Camera

3. Motors
   Motors are required to manipulate and rotate the cube's faces. Servo motors provide precise angular control, making them ideal for this application. Alternatively, stepper motors offer higher precision in movement but require additional driver circuits for operation.

Figure 3. Stepper motor

4. Motor Drivers
   Motor drivers regulate the power supply to the motors and control their movements. Since microcontrollers cannot directly supply the required voltage and current to

motors, L298N motor drivers (for stepper motors) or PCA9685 servo drivers (for servo motors) are used to manage motor control effectively.

Figure 4. L298N motor drivers

5. Sensors
Sensors, such as rotary encoders, help track motor movements and ensure accurate cube rotations. These sensors provide real-time feedback to detect misalignment and correct errors during cube manipulation.

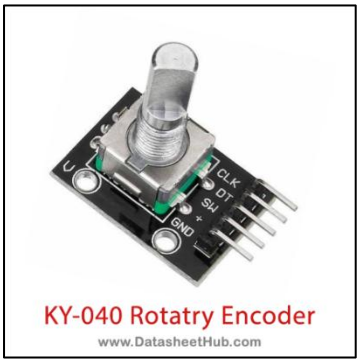

Figure 5. Rotatory Encoder Sensor

6. Frame and Grippers
The frame is the structural component that holds the Rubik's Cube in place, ensuring stability while the motors operate. 3D-printed or CNC-machined components are used to build a durable and precise frame. The grippers securely grip and rotate the cube without misalignment or damage.
7. Power Supply
A stable power source is required to operate the robot's components. This can be in the form of a rechargeable Li-ion battery, which provides portability, or an AC adapter, which offers continuous power supply for prolonged operations.

Software Requirements

1. Programming Languages: The robot's software is developed using programming languages such as Python and C++. Python is primarily used for image processing, algorithm implementation, and system control, while C++ is used for microcontroller firmware to handle real-time motor control efficiently.
2. Image Processing Library: OpenCV (Open-Source Computer Vision Library) is used to process images captured by the camera. It helps detect cube colours, segment different faces, and analyze the cube's current state for solving. Functions like cv2.cvtColor() and cv2.findContours() are used to process images and identify colours.
3. Solving Algorithm Library: The Rubik's Cube Solver uses a mathematical solving algorithm to determine the optimal sequence of moves required to solve the cube. Kociemba's algorithm is widely used due to its efficiency in finding the shortest possible solution. It is implemented using the kociemba Python library or similar algorithmic approaches.
4. Motor Control Software: Motor control software is responsible for sending movement commands to the motors. It includes microcontroller firmware that generates PWM (Pulse Width Modulation) signals to control the speed and direction of motors. Libraries like RPi.GPIO (for Raspberry Pi) and Adafruit_PCA9685 (for servo motors) are used to manage motor movements accurately.
5. User Interface: A user interface allows interaction with the robot. It can be a Command Line Interface (CLI), where users can enter commands to start the solving process, or a Graphical User Interface (GUI) built using Tkinter or similar frameworks. The GUI can display cube states, solving steps, and system status for a better user experience.

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. Beyond its technical contributions, this project serves as an educational tool, inspiring students and enthusiasts to explore robotics and computational problem-solving. Furthermore, the technologies and methodologies applied in this project extend beyond puzzle-solving, offering valuable applications in industries such as manufacturing, logistics, and healthcare, where automation and intelligent decision-making are essential.

References

Gan Robot – Intelligent Cubing Pioneer, Gan Cube, Available (Accessed Jan 15,2024): https://www.gancube.com/gan-robot/

J. Terra, Keras vs Tensorflow vs Pytorch: Key Differences Among Deep Learning, Simplilearn, Aug 2023, Available (Accessed Jan 15, 2024): https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article#what_is_pytorch

Top 16 computer vision libraries, SuperAnnotate, May 2023, Available (Accessed Jan 15, 2024): https://www.superannotate.com/blog/computer-vision-libraries.

Dan, V., Harja, G., & Naşcu, I. (2021, February). Advanced Rubik's Cube Algorithmic Solver. In *2021 7th International Conference on Automation, Robotics and Applications (ICARA)* (pp. 90-94). IEEE.

Barucija, E., Akagic, A., Ribic, S., & Juric, Z. (2020, September). Two approaches in solving Rubik's cube with Hardware-Software Co-design. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)* (pp. 128-133). IEEE.

Sawhney, H., Sinha, S., Lohia, A., Jalan, P., & Harlalka, P. (2013). Autonomous Rubik's Cube Solver Using Image Processing. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY*, 2(10).

Li, T., Xi, W., Fang, M., Xu, J., & Meng, M. Q. H. (2019, December). Learning to solve a rubik's cube with a dexterous hand. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 1387-1393). IEEE.

Liu, S., Jiang, D., Feng, L., Wang, F., Feng, Z., Liu, X., ... & Cong, Y. (2019). Color Recognition for Rubik's Cube Robot. *arXiv preprint arXiv:1901.03470*.

T. Li, W. Xi, M. Fang, J. Xu, and M. Q.-H. Meng, "Learning to solve a Rubik's Cube with a dexterous hand," arXiv preprint arXiv:1907.11388, 2019.

I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al., "Solving Rubik's Cube with a robot hand," arXiv preprint arXiv:1910.07113,2019

A. Akagic, E. Buza, R. Turcinhodzic, H. Haseljic, N. Hiroyuki, and H. Amano, "Superpixel accelerator for computer vision applications on arria 10 soc," in 2018 IEEE 21st International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp. 55–60, IEEE, 2018.

M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," in Advances in Neural Information Processing Systems, 2019.