BAXTER SOLVES THE RUBIK'S CUBE

A PREPRINT

Abaho Katabarwa
Department of Computer Science
Brown University
Providence, RI 02912
abaho_katabarwa@brown.edu

August 18, 2019

ABSTRACT

The Rubik's cube presents a unique challenge for robotics in both manipulation and computing given its small size (measuring 5.7 centimeters), it's 6 manipulable faces, and it's immense state space (4.3e19). While this problem has been solved with task-specific robots such as BriKuber [1], it has been shown to be difficult for complex multi-purpose robots. The following paper details an approach and a work-space setup to solve a standard sized Rubik's cube using a Baxter robot. Using a carriage the Baxter is able to turn the faces of the cube with a single arm. Both robot arms are used to perform face-switching. This approach to manipulating a cube enables the Baxter robot to solve the cube from any arbitrary state given an initial start state. Additionally, this start state can be deciphered using the on-board cameras on the robot.

1 Introduction

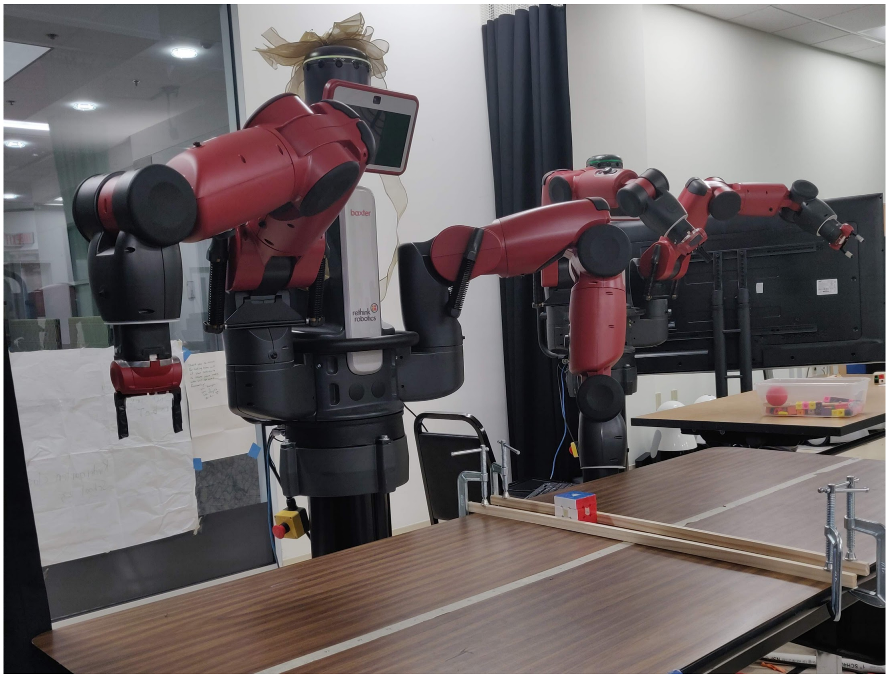
The Rubik's cube is a popular and a relate-able puzzle game developed by Erno Rubik in 1974. For this reason, solution algorithms to it are widely researched, available, and understood. While the Baxter Research Robot has demonstrated it's capability for a varied set of tasks (picking, pushing, pulling, etc), it's limits can be challenged by highly dexterous complex tasks. Solving the Rubik's cube is such a task for the Baxter robot. Cube manipulation is difficult for three following reasons. Firstly, the cube needs to maintain a location that is roughly within the "green" zone (The "green" zone is the area within the robot's work-space in which the robots joint position can be easily determined with inverse kinematics calculations given the end-effector position). Secondly, the cube requires the robot arms to horizontally and vertically interact with the cube. And thirdly, cube rotations/cube moves must be completed with minimal error in order to be reproduce-able. For these reasons, demonstrating the Baxter solving the Rubik's cube would contribute to the understanding of the dexterity capabilities and limitations of the Baxter robot.

There have been multiple previous successful approaches to solving this problem. Two such approaches are highlighted. One is a demonstration by Rethink Robotics group developers of the Baxter robot,a more detailed elaboration of this demonstration is done in the next section [2], and the second is a course project completed at University of California Berkeley[3]. While both projects are successful at solving the cube, the first approach makes use of a external scanner to determine the state of the cube. The second approach, in light of the difficulties of manipulating a standard cube, makes use of an oversized cube [4].

The technical approach draws from the high level steps taken by the previous approaches detailed above. It breaks the tasks into three phases: state determination, solution finding, cube manipulation. The state determination is done by scanning the cube using the Baxter's hand camera. Solution finding is done using a solver that executes the Kociemba algorithm. Finally, cube manipulation is performed using hand-crafted moves programmed using ein[5] and rospy[6] libraries.

Evaluation of the approach shall be based on four criteria: firstly whether the robot is able to solve the Rubik's cube, secondly, the degree of autonomy in doing so, thirdly, the efficiency in accomplishing each move in the solution, and finally, the reproduce-ability of the end-to-end solution.

Figure 1: *Image of the environment at the final state with a solved cube and the manipulating arm returned to home position*



2 Related Work

The Baxter research robot has been used to solve the Rubrik's cube before [2]. This demonstration used four steps to solve the cube. Firstly, there was a detection phase using a scanner which provided an initial suggestion of the state of the cube. Then there was a verification phase which determined that the suggested state was a valid cube state. This was followed by the solve phase in which the Baxter ran the Kociemba algorithm to solve the cube [7] which seeks to reach a subset of pre-defined cube states and then runs an iterative deepening search to the find the path to the goal state. Finally, the Baxter enters the manipulation phase in which it physically solves the cube using the plan generated by the previous phase. The proposed approach seeks to solve the cube solely using the Baxter robot. The approach detailed in this paper is different from the described approach because it accomplishes face-turns with a single arm therefore decreasing error and complexity. Furthermore, it does not require the use of an external scanner to determine the cube state. Instead, it attempts to use Baxter's on-board cameras to determine the cube state.

3 Technical Approach

A formal description of the problem requires a technical description of the environment. The environment is primarily constituted of three objects: the Baxter robot, a standard Rubik's cube ($3x3$, 6-faced, with sides measuring 5.7cm), and a work table. The work table is supposed to act as a platform for Rubik's cube and elevate the cube to a height at which it can be easily picked and placed by the Baxter robot. This height is approximately 0.75m.
In the initial state of the problem, the cube, in a scrambled state, must be resting on the work table. From this position, the robot should determine the state of the cube, compute a solution, and finally execute the arm movements required to move the cube into a solved state. At the end of successful completion of a run, the cube should in a solved state in the same position as it was in the initial state.

The technical approach solves the problem in two phases. Cube manipulation is important in both phases.
The first phase is a state determination. In this phase, all 6 sides of the cube's faces are exposed to one of the Baxter's hand cameras. This is done by *face-turning* one of two Baxter cube manipulations explained in more detail below. The image produced is then processed to determine the local color state. Each of these local states is aggregated to determine the global state of the cube. The second phase is cube solving. In this phase, *face rotation* is also used in tandem with *cube rotation* to manipulate the cube from its initial state to its solved state.

Figure 2: *Intermediary state in cube rotation, the hand-off*

3.0.1 Cube manipulation strategy

Face rotation:
Cube face rotation is done with a single Baxter arm. In order to enable this, a carriage is added to the work-space. When the cube is placed within the carriage, a single rotation of the Baxter arm from a grasping position will rotate the downward oriented face of the cube.

Cube rotation:
In order to get a global view and in order to rotate a face of the cube, it is required that the appropriate face is exposed or engaged within the carriage. Cube rotation is a two armed action in which the cube is passed from one arm to another. In the process of this exchange the cube is rotated and placed back into the carriage with a different face exposed.

4 Evaluation

The goal of the technical approach was three-fold. Firstly, it was important that the complexity of the problem be decreased to the detriment of the total duration of the run. This was achieved by using a single arm whenever possible and only adding the use of the second arm when it was absolutely required. The second goal was promoting the modularity of the problem. Being able to approach the problem in phases was preferable because each phases could be developed in a non-linearly un-ordered fashion. Furthermore, it was possible to complete a run without having necessarily solved all the intermediary parts of the problem. Finally, the goal was increase the autonomy of the task. While it was not always possible, it was important that the robot accomplish the sub-tasks of the problem without human help or correction wherever possible.

The primary benchmark for goal success was determined by first all of whether the Baxter was able to solve the Rubik's cube from a scrambled state (*https://drive.google.com/open?id=1q74bxMWrpkhh7uJEl4xtlmpwjC98yDd_*). The secondary benchmark was the degree of autonomy present in the run. While the primary benchmark was achieved, it was important to note that complete autonomy wasn't achieved. Cube rotation as well as state determination had to done with human intervention. This was primarily because error in the position of both hands would increase to unworkable magnitudes as each arm tried to compensate for the others error and vice-versa (*https://youtu.be/kflntGidnAY*). Furthermore, human intervention was required to error-correct the cube position in between face-rotations.

5 Conclusion

The problem statement was to solve a standard Rubik's cube given a Baxter robot and a work table environment. While the Baxter robot proved dexterous enough to solve the cube given simple single arm manipulations, two arm manipulations proved difficult given the error accumulation from both end-effectors.

Future work will focus on alternative methods to manipulate the cube. While the approach in this paper was simple, I believe that even simpler methods can be defined (for example, attempting to cube rotate with a single arm). Additionally, future work will focus on how to reduce the end-effector error. From the Rethink Robotics demonstrations, it is apparent that two Baxter arms can hold then same object without error accumulation that was observed. This work will be focused on how to recreate that scenario. Finally, on a more aspirational note, it will be useful to explore how to guide the Baxter in manipulating the cube based on the difference between its current state and the intended next state instead of hard-crafted movements. I believe that this approach will eliminate error and move towards complete autonomy.

References

[1] BricKuber - How To Build a Raspberry Pi Rubiks Cube Solving Robot. https://www.dexterindustries.com/projects/brickuber-project-raspberry-pi-rubiks-cube-solving-robot-project/.

[2] Baxter Solves Rubiks Cube. http://sdk.rethinkrobotics.com/wiki/Baxter_Solves_Rubiks_Cube.

[3] Lichang Xu Gongbo Yang Lingyao Zhang Shijie Gao, Yunfan Song. Smart Baxter, EE106A-Proj. https://github.com/LichangLovesCS/EE106A-Proj.

[4] Gongbo Yang. https://www.youtube.com/watch?v=pctwe89aSVg.

[5] Stefanie Tellex. Ein 1.0. http://h2r.github.io/ein/.

[6] Ken Conley. ropsy. http://wiki.ros.org/rospy.

[7] Herbert Kociemba. The Two-Phase-Algorithm. http://kociemba.org/cube.htm.