Prospects for Combining Task and Motion Planning for Bi-Manual Solution of the Rubik's Cube

University of Washington – Sensor Systems Laboratory

Boling Yang, Patrick Lancaster, Joshua R. Smith

Introduction

The Rubik's cube is a puzzle that requires a long sequence of manipulation operations to solve. Changing the cube state requires two hands (one to constrain part of the cube, and one to rotate the remainder of the cube). Also, solving the cube requires re-grasping, and (while it may not be necessary) it is very natural to use bimanual re-grasping.

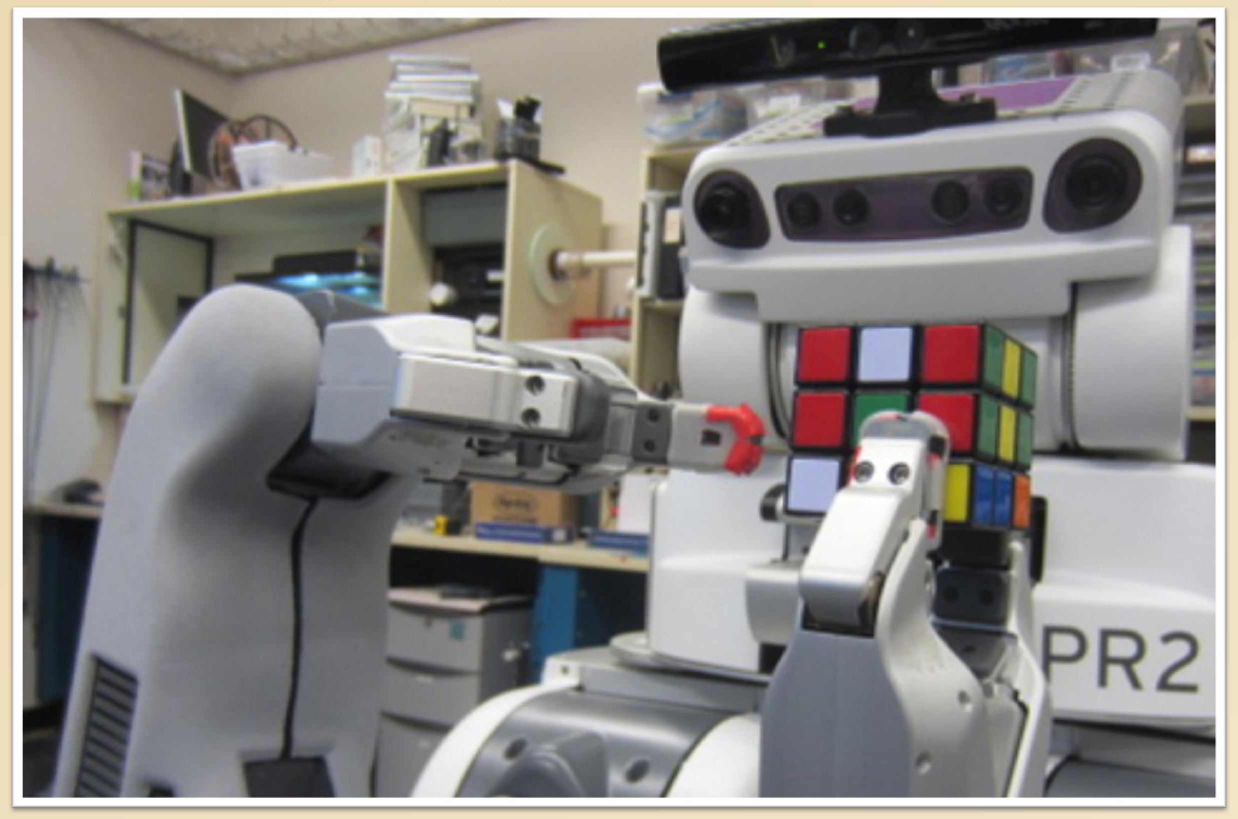

Fig. 1: By combining task planning with pre-touch sensor aided motion planning, the robot is able to solve the Rubik's cube.

Our initial motivation for the robotic solution of the Rubik's cube was to explore the use of pre-touch sensing during the course of long manipulation sequences to correct actuation errors as they arise. Indeed we found that without pre-touch sensing [3] , the PR2 typically makes a catastrophic error after an average of 8 moves; with pre-touch sensing, an indefinite sequence of moves is possible. We developed a system that demonstrates the end-to-end functionality using a task-based planner to choose a sequence of cube moves, and a single-purpose planner to choose bi-manual hand actions to execute the cube plan.

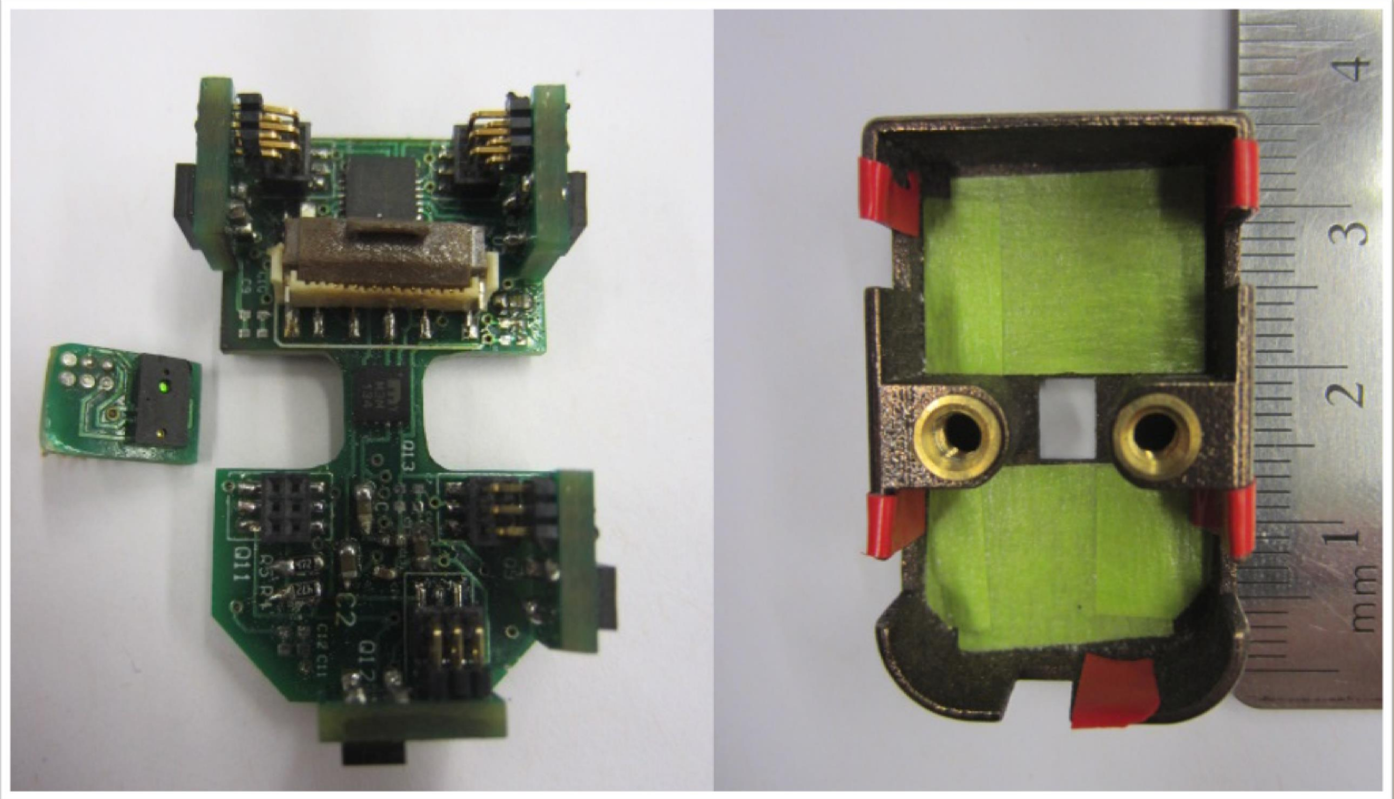

Fig. 2: Left: The PCB that compose the sensor. Right: The 3D printed sensor casing. The hole in the middle of the case is for a sensing module soldered to the bottom of the main PCB.

Separate Task & Motion Planning

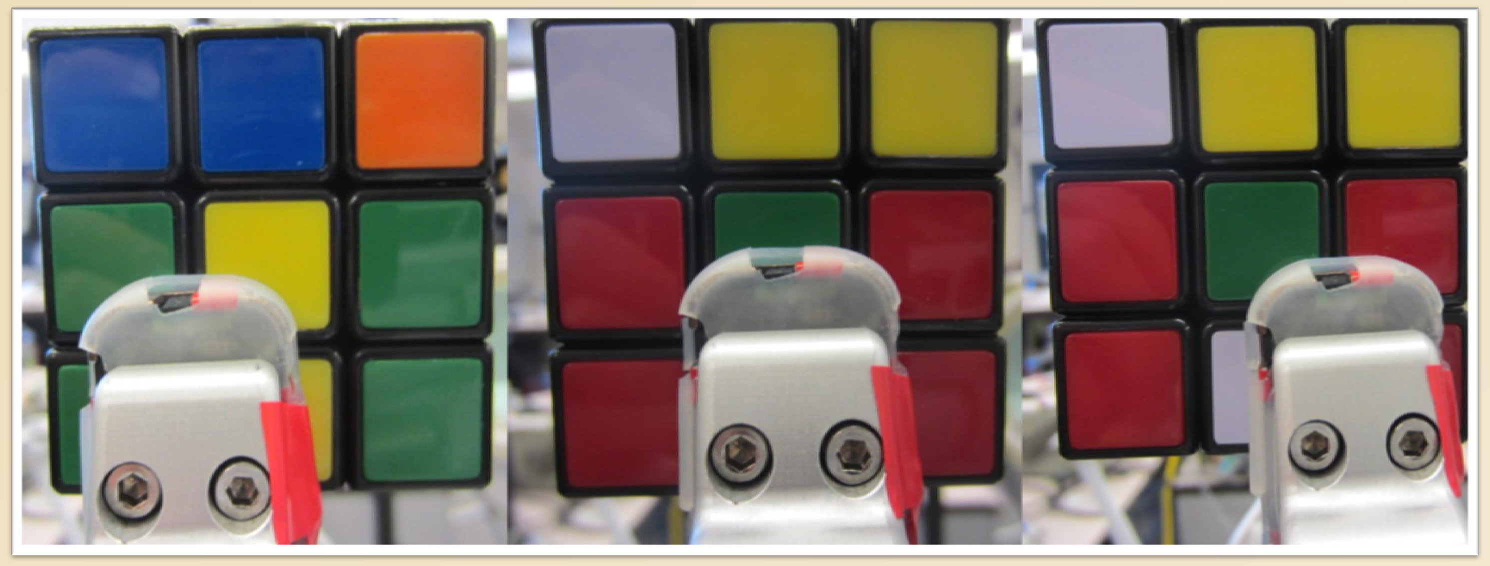

Fig. 3: Three grasp points used by the current system. These correspond to the grasp points labeled 3, 5, and 7 in Fig. 4.

Task Planning: Given the initial state of the Rubik's cube, the robot generates a solution using Kociemba's Two Phase algorithm [1][4], which first uses an iterative deepening A∗ algorithm to search for maneuvers that transform the scrambled cube to a state that is an element of a particular group G with special properties that correspond to a partial solution with corners and edges in the right locations but possibly with incorrect orientation. In phase 2, the 8 corners and all edges will be set to the correct orientations. The algorithm searches to find a solution that requires a minimal number of cube face rotations. It is known that any cube can be solved with at most 20 moves.[4]

Motion Planning: The space of the general grasp planning problem is surprisingly large. There are up to 120 feasible grasp points for one hand on a cube and no more than 120x100 = 12,000 two handed grasp poses. We simplified the motion planning problem enough that it can be handled by a finite state machine. The current system only considers a subset of the grasp points shown in Fig. 4. As a result, it disregards many possible motion plans that may require less execution time.

Fig. 4: The proposed system will use ten unique grasp points. Assuming the gripper is approaching from the bottom of the cube, each white box represents a possible grasp point.

Joint Task-Motion Planning

Our future work will reformulate the Rubik's cube problem as a search through a larger state space that is the product of cube states and the relevant robot states. The more general state space we are contemplating should capture the interaction between the robot's manipulators and the object of interest. The original Rubik's cube state will be augmented with variables describing the state of each gripper. These variables include:

1) The position of the gripper with respect to the cube. When the robot positions either of its grippers prior to making contact with the cube, there is a discrete set of grasp points that it can choose. Along each parallel pair of edges of the cube, there are ten unique grasp points, as shown in Fig. 4.
2) The orientation of the gripper. The gripper's orientation consists of a heading (north, east, south, west, up, down) and a roll that rotates the gripper in 90 degree increments.
3) Whether the gripper is open or closed.

It might be possible to generate the table of valid transitions between states by 'compiling' a small set of rules (that models the physics of the system) into this explicit state transition table. The set of valid transitions can be viewed as the edges in a bi-partite graph, where initial states are represented by nodes on the left and final states are represented by nodes on the right. If we start with an initially empty set of valid transitions, and add only transitions that are known to be valid, then we will never make incorrect or kinematically infeasible moves.

Alternatively, we could grow an initially empty set of state transitions by validating candidates with an offline motion planner. This results in full integration of motion and task planning: motion planning constrains the task plans that can be generated through state transition validation, and task planning specifies which types of motion plans are needed when the task is actually attempted. We are particularly interested in how these ideas can be applied to sequential tasks that require precision, such as object construction [2]. We believe that by augmenting the object states with the relevant robot state, more effective plans can be produced.

References

[1] pr2 rubiks solver, 2011. URL https://github.com/ uu-isrc-robotics/pr2 rubiks solver.
[2] Ross A Knepper, Todd Layton, John Romanishin, and Daniela Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 855–862. IEEE, 2013.
[3] Brian Mayton, Louis LeGrand, and Joshua R Smith. An electric field pretouch system for grasping and comanipulation. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 831–838. IEEE, 2010.
[4] Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. The diameter of the Rubik's cube group is twenty. SIAM Review, 56(4):645–670, 2014.