Interactive Robot Knowledge Patching Using Augmented Reality

[1]Center for Vision, Cognition, Learning, and Autonomy - University of California, Los Angeles

Hangxin Liu[1], Yaofang Zhang[1], Wenwen Si[1], Xu Xie[1], Yixin Zhu[1], Song-Chun Zhu[1]

Introduction

We present a novel Augmented Reality (AR) approach, through *Microsoft HoloLens*, to address the challenging problems of diagnosing, teaching, and patching interpretable knowledge of robot.

The proposed system is able to:

- Display knowledge structure in addition to robot state visualization
- Reveal robot's decision making during its task execution.
- Allow user interactively patches robot's knowledge structure.

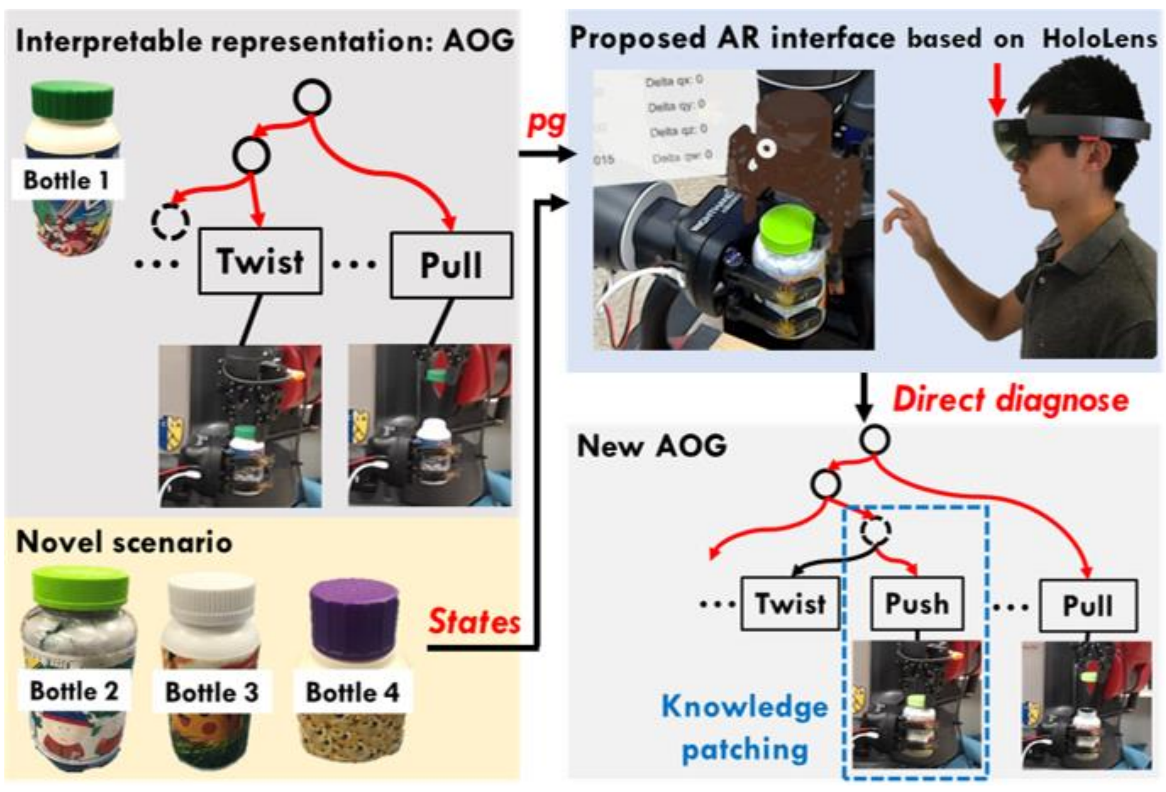

Fig 1. Given a knowledge represented by a T-AOG of opening normal bottles, user patches the knowledge through the proposed AR interface to allow robot to open medicine bottles with safety lock.

Knowledge Visualization

We reveal the robot's inner functioning and knowledge structure through the AR interface. The knowledge structure is represented by a T-AOG, which encodes a repertoire of opening the bottles.

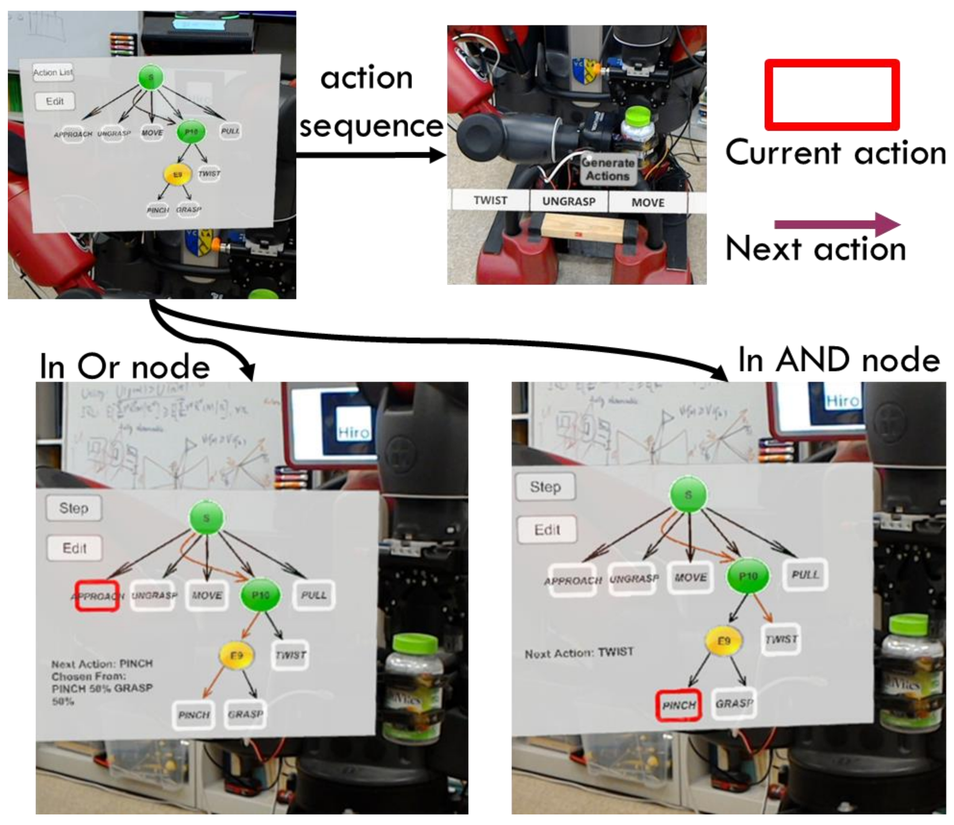

Fig 5. The knowledge representation, T-AOG, generates valid action sequences for opening bottles. The robot decides next action probabilistically in at OR-node, and deterministically at AND-node.

Parsing the T-AOG produces a sequence of atomic actions that the robot can execute to fulfill the task. By closely monitoring the dynamic parsing process, the users can supervise the decision making process of the robot. Next action is selected with 100% at an AND-node and its child nodes are deterministically executed in a temporal order. An OR-node indicates a switching configuration among its child nodes; one of its child nodes is selected based on the branching probability.

Robot Imitation Learning

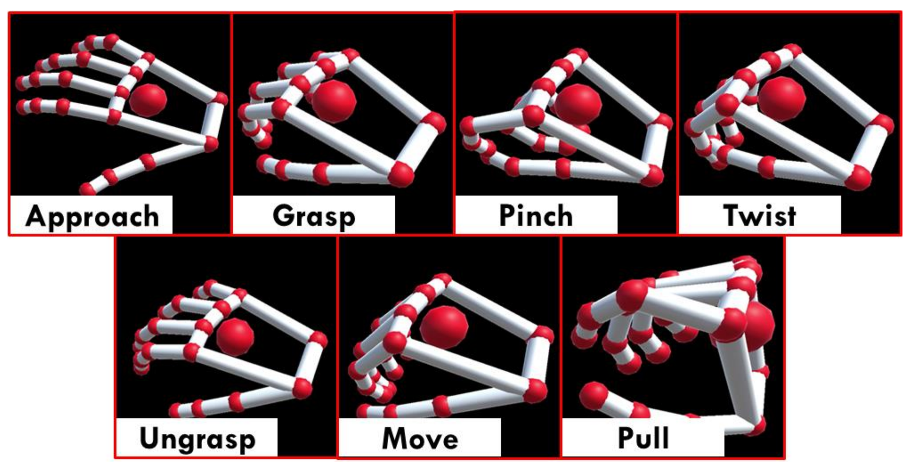

Fig 2. Average hand skeleton of 7 atomic actions..

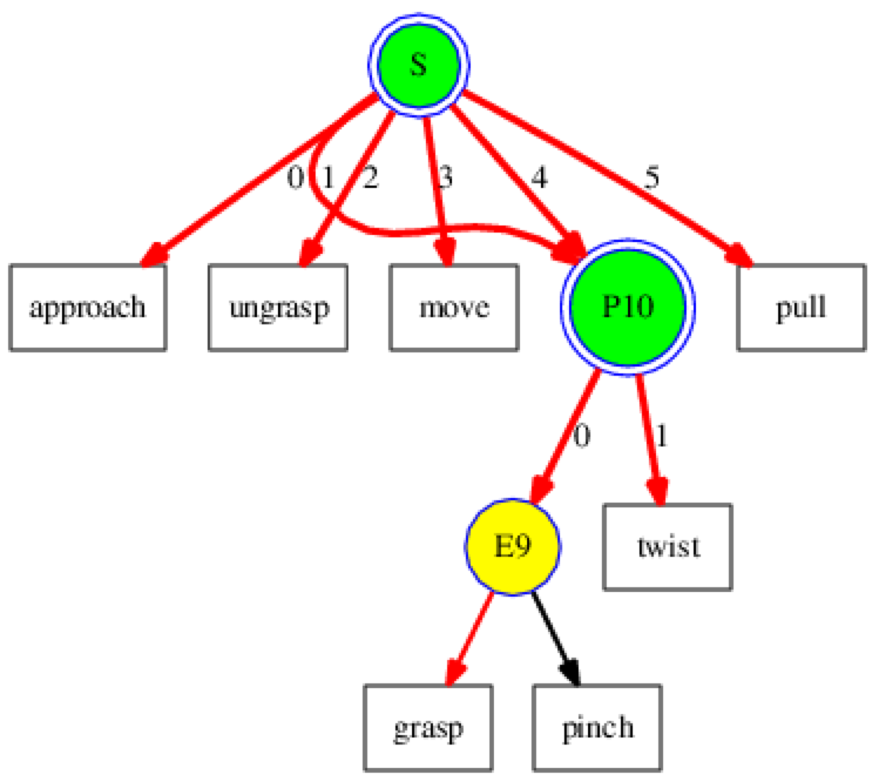

Fig 3. Learned AOG. The green and yellow nodes are And-nodes and Or-nodes, respectively. The red edges indicates a possible *pg*.

Human Data Collection: Twenty manipulations of opening medicine bottles are collected using hand tracking by *LeapMotion* sensor.

Robot Platform: A Rethink Baxter robot. It is equipped with a ReFlex TackkTile gripper on the right wrist, and a Robotiq S85 parallel gripper on the left. The entire system runs on Robot Operating System (ROS), and arm motion planning is computed using *MoveIt!*.

Mirroring Human Actions to Robot: We endow the robot with a dictionary of atomic actions corresponding to the human's manipulative actions. Specifically, each action is represented by the change of robot's end-effector pose or the open/close of the gripper.

State Visualization

We use *HoloLens* to track an AR tag on robot and overlay the state information of interest at the corresponding reference frames. Using gesture control, the user can easily turn on or off the sensory information, as well as control the robot.

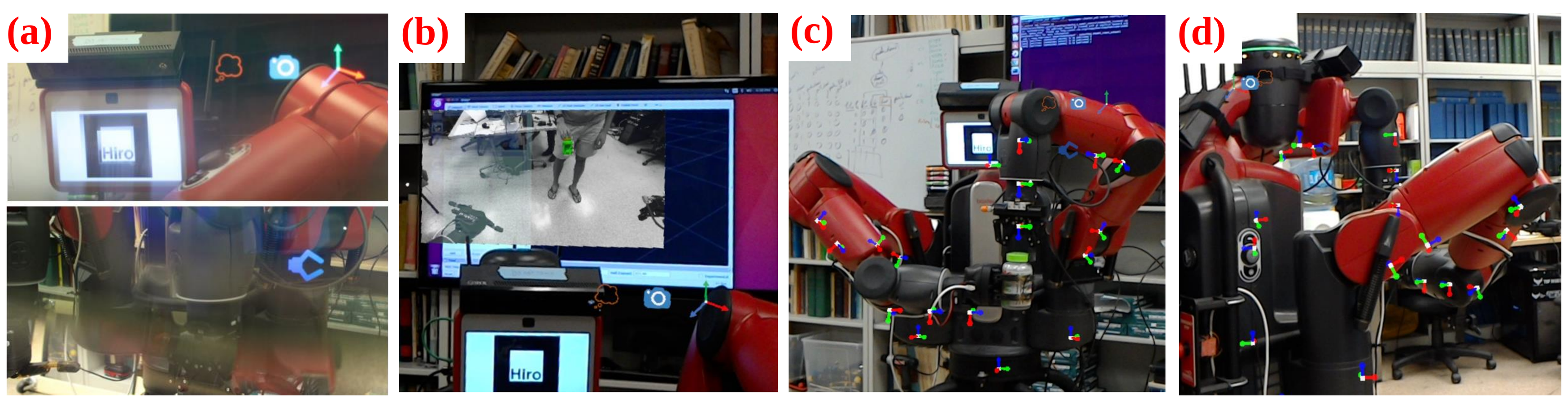

Fig 4. (a) Users can use gesture control to turn on or off sensory information (b) the camera icon is designed to turn on or off the Kinect camera. (c) robot's joints positions are displayed according to the tracked AR tag. (d) The frames remain in place although the AR tag is lost during tracking later.

Diagnosing

The robot's force readings of its left end-effector when performing the *grasp*, *pinch* action in opening conventional *Bottle 1* are identical, and one of the actions can be removed.

Wrong actions can also be discovered. In opening a medicine bottle with child-safety lock that requires an additional pressing down action on the lid (*e.g.*, *Bottle 2-4*), the *twist* action only applies small downward force, while later taught *push* action produces larder downward force.

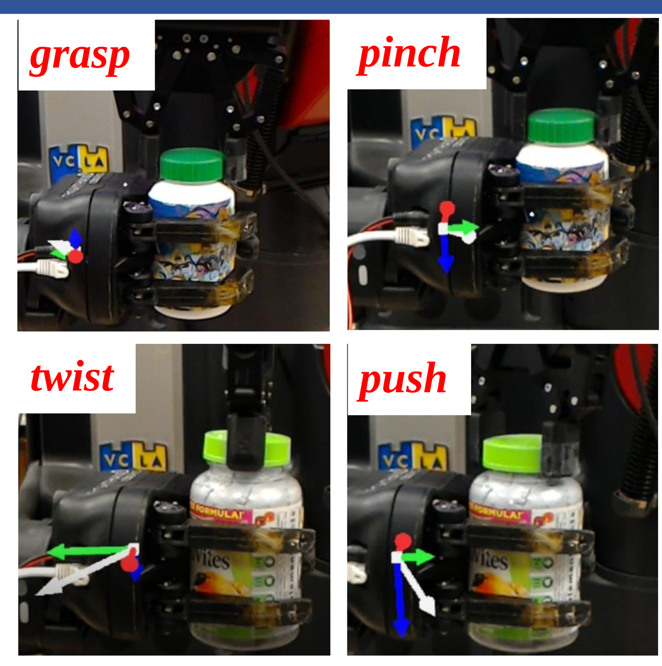

Fig 6. The R, G, B arrows indicate the canonical X, Y, Z direction and the white arrow is their vector sum.

Knowledge Patching

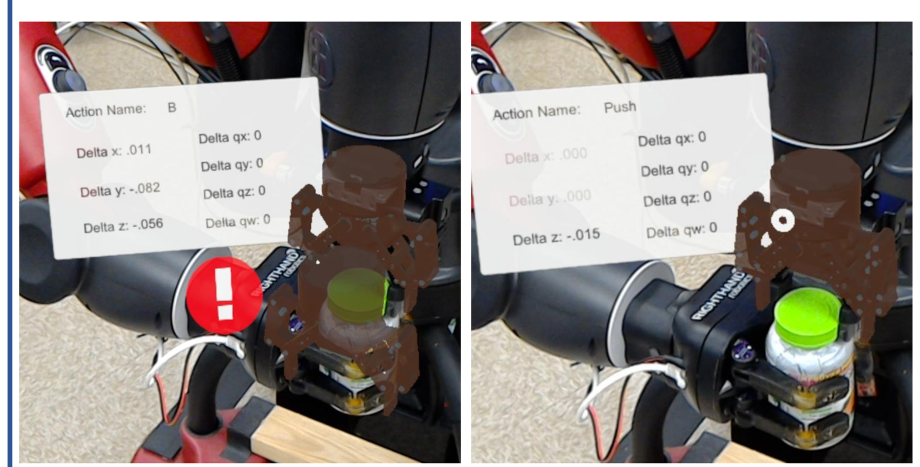

Fig 7. Robot control interface.

The AR interface allows users to easily provide new guidance by dragging the virtual gripper hologram to a new pose.

The user can define a new action through modifying the existing *twist* action by moving the end-effector downward to produce pressing force, namely *push* action.

The following figure depicts the robot's step-by-step execution. Before the user patches its knowledge, it successfully open a conventional bottle *Bottle 1*. But in its attempt failed in opening a medicine bottle *Bottle 2* with safety lock due to the lack of pressing down action. After patching its knowledge, the robot succeeds in opening the medicine bottle *Bottle 2*.