Robot Programming by Situated Illustration

Yuxiang Gao[1] and Chien-Ming Huang[1]

***Abstract*— As robots begin to provide daily assistance to individuals in human environments, their end-users, who do not necessarily have substantial technical training or backgrounds in robotics or programming, will ultimately need to program and "re-task" their robots to perform a variety of custom tasks. To support this need for accessible end-user robot programming, this research explores an innovative method of robot programming through situated specification and simple illustration of tasks. In this work, we present PATI, an implementation of this new programming method, and an empirical evaluation of its effectiveness and usability. Our results show that participants needed significantly less training time before they felt confident in using PATI than they did for a state-of-the-art method. Moreover, participants were able to program a robot manipulator to complete a pick-and-place task significantly faster with PATI. This work indicates a new direction for end-user robot programming.**

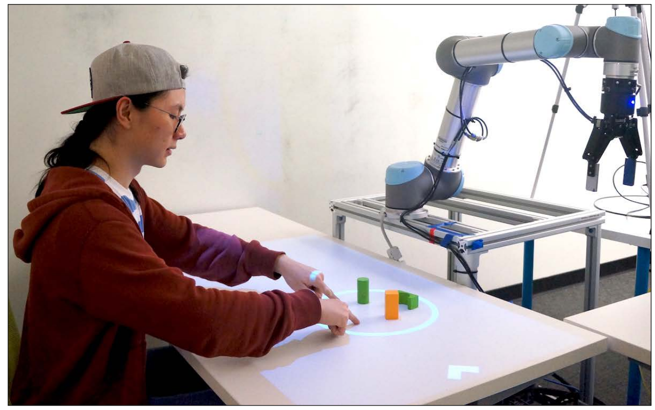

Fig. 1. PATI supports the augmented and situated specification of robot tasks, allowing users to teach their robots new tasks by directly referencing and interacting with the environment through tools (e.g., shape tools) and common touch screen gestures, such as pinch, tap, or drag-and-drop.

I. INTRODUCTION

Robots are beginning to transform the future of work. They hold great promise to accelerate flexible manufacturing, enable collaborative construction, and assist in patient care. To succeed in these domains and beyond, robots need to provide custom assistance to accommodate individual needs, environmental constraints, and task requirements. Different from research that seeks to build general intelligence into autonomous robots, our research focuses on enabling everyday people, who do not necessarily have technical backgrounds in engineering and related fields, to be able to program robots to perform a variety of custom tasks.

Programming by Demonstration (PbD), or *Learning from Demonstration*, [2], [3] is a common method for end-user robot programming. Kinesthetic teaching, a particular method of PbD, has been explored to facilitate flexible manufacturing, allowing production workers to reskill collaborative robots for custom situations and task requirements. Common kinesthetic teaching involves little perception of the environment and relies on replaying demonstrated action trajectories or tracing recorded key waypoints (e.g., [4]). Though limited perception constrains the generalizability and scalability of PbD, perceiving task-relevant objects and the environment usually requires specialized processes for training objects for visual recognition and robot manipulation (e.g., [5]), which consequently creates additional technical barriers for end-users.

To address these limitations and enable accessible robot programming, in this work, we seek to streamline the processes of visual tracking, object referencing, and task specification in robot programming by enabling users to reference and annotate the environment directly through gestural input (Fig. 1). In particular, we developed a Projection-based Augmented Table-top Interface (PATI) for robot programming [1], a prototype system designed to explore the possibilities of robot programming by situated illustration.

II. AN OVERVIEW OF PATI

The PATI system allows users to program a robot manipulator directly on a table-top surface through an intuitive tangible interface. The system involves the use of a UR5 robot manipulator, a top-down projector, and a Kinect2 RGB-D camera mounted on the ceiling. It further consists of four software modules: *Visual Perception*, *Tangible User Interface (TUI)*, *Program Synthesis*, and *Robot Control*.

The Visual Perception module detects and tracks objects in the environment, as well as a user's touch input, through the use of depth data and RGB color images from the Kinect2 camera. The user's input is then sent to the TUI module, in which different types of gestures are recognized. Once the user completes his/her program specification via the interface, the Program Synthesis module translates the user's task-level specifications into a set of robot commands, including gripper actions and the waypoint specifications of intended trajectories. The commands are then forwarded to the Robot Control module, which subsequently generates motion plans based on the input commands for robot execution.

*A detailed version of this work is presented in [1].

[1]Department of Computer Sicence, Johns Hopkins University, Baltimore, MD 21218, USA. Email contact: yuxianggao@jhu.edu

[1]This project is available at https://intuitivecomputing.github.io/PATI.

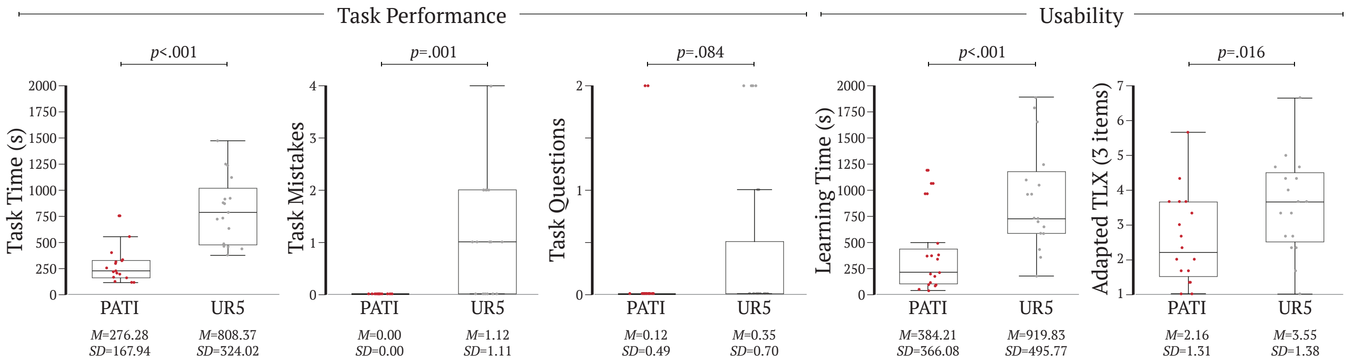

Fig. 2. Box and whisker plots of data on the objective measures of task performance and the subjective measure of usability. The top and bottom of each box represent the first and third quartiles, and the line inside each box is the statistical median of the data. The length of the box is defined as the interquartile range (IQR). The ends of the whiskers are the first quartile minus 1.5 IQR and the third quartile plus 1.5 IQR.

III. EVALUATION

A. Methods

We conducted a user study to compare PATI with the built-in programming method provided by the manufacturer of the UR5 robot. Our evaluation focused on simple manipulations tasks (pick-and-place) fundamental to a variety of common functions that robots are envisioned to assist people in performing. Our central hypothesis is that the PATI system will help participants achieve greater task performance in simple manipulation tasks such as pick-and-place, and offer higher usability to participants than the built-in PbD method for the UR5 robot manipulator.

B. Results

We recruited 17 participants (9 females) on a college campus for this study. The participants had a variety of educational backgrounds, including engineering, psychology, writing, and medicine. Fig. 2 summarizes the results of our evaluation study.

1) Task Performance: Our data showed that participants needed significantly less time and did not make any mistakes when using the PATI system, whereas they made more than one mistake on average when using the UR5 interface.

2) Usability: Our data showed that the participants needed significantly less time before they became familiar with the programming interface and felt ready for the experimental task earlier when using PATI than they did when using the UR5 interface. Moreover, self-report data suggested that the participants experienced less task load when using PATI than when they used the UR5 interface.

IV. CONCLUSION AND FUTURE WORK

To enable accessible robot programming, we proposed a new paradigm of *robot programming by situated illustration* where end-users can use simple tools and intuitive gestural commends to provide task specifications. In this work, we present PATI, an open-source system that implements this new paradigm of robot programming. An empirical evaluation comparing PATI with a state-of-the-art method for robot programming shows the effectiveness and potential of PATI for end-user robot programming. In particular, our results reveal that participants were able to learn how to use the programming interface in a shorter period of time, achieved a greater task efficiency, and had less task load when using PATI than they did when using the state-of-the-art PbD system. While promising, the present work has limitations that highlight directions for future research.

Future work needs to investigate the potential and limitations of robot programming by situated illustration beyond simple manipulation tasks. New task tools, action operations, and input styles (e.g., augmented reality and tangible instruments [6]) should be explored and evaluated. Moreover, in addition to task-level specification, future work should explore how this novel programming paradigm may be used to craft motion-level and goal-driven social robot behaviors. Future research should also investigate ways to minimize environmental modifications (e.g., integrating the capabilities of projection and 3D sensing into the robot) and the needed system calibration to allow people to use it with ease in custom contexts.

V. ACKNOWLEDGMENT

This work is supported by the Johns Hopkins University.

REFERENCES

[1] Y. Gao and C.-M. Huang, "Pati: a projection-based augmented tabletop interface for robot programming," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*. ACM, 2019, pp. 345–355.

[2] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[4] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 391–398.

[5] J. Huang and M. Cakmak, "Code3: A system for end-to-end programming of mobile manipulator robots for novices and experts," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 453–462.

[6] Y. S. Sefidgar, P. Agarwal, and M. Cakmak, "Situated Tangible Robot Programming," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 473–482.