Development of a Hand Motion Tracking based Teleoperation System for Imitation learning

Jinchul Choi
Autonomous IoT Research Sec., ETRI
Daejeon, Korea
spiders22v@etri.re.kr

Heechul Bae
Autonomous IoT Research Sec., ETRI
Daejeon, Korea
hessed@etri.re.kr

Chan-Won Park
Autonomous IoT Research Sec., ETRI
Daejeon, Korea
cwp@etri.re.kr

***Abstract*—Imitation learning is emerging as one promising approach for robots to acquire skills. Since imitation learning provides a way to learn policies by imitating the behavior of experts, it requires a sufficient amount of sophisticated expert behavior trajectories. However, current interfaces, such as kinesthetic teaching or remote manipulation, have significant limitations in efficiently collecting diverse demonstration data. To address this issue, this work proposes an alternative interface for imitation that simplifies the demonstration acquisition process using a hand tracking solution while facilitating the transfer of human actions to the robot. Performance results show that the proposed system is effective for imitative learning, not only imitating expert demonstrations with low latency, but also helping to collect elaborate demonstration data.**

***Index Terms*—Teleoperation, Mimetic interfaces, Motion tracking, Imitation learning**

I. Introduction

Imitation learning is a form of supervised learning, a machine learning technique in which an agent derives a policy for performing a task by imitating the actions and decisions of an expert. The goal of imitation learning is to derive a map of policies or sequences of actions based on the state of the expert that a skilled expert would take to successfully complete a task [1]. This characteristic of imitation learning has a number of advantages over other types of machine learning [2]. First, it can reduce the time and effort required to derive a policy for a specific task. Second, it can learn from the behavior of an expert that cannot be captured by rewards alone. Third, by leveraging the expertise of human demonstrators, agents can perform tasks more efficiently and effectively.

A major bottleneck in current imitation learning is the use of interfaces such as kinesthetic training or teleoperation to acquire expert demonstrations [3]. Kinesthetic teaching, in which an expert physically guides the robot by applying force, is an effective way to enable non-experts to configure and manipulate robots for demo collection [4], [5]. However, this method is somewhat cumbersome and requires manipulating each action one by one, so it is not suitable for complex manipulation tasks [6]. Teleoperation, in which an expert operates a system or machine from a distance using a control interface, has been successfully applied to a variety of robotic tasks such as robot navigation [7], object grasping [8], car driving [9], and even humanoid robots [10]. However, devising such interfaces for robot manipulation remains a challenge. Another alternative is to use motion capture solutions that can record the movements of objects or people. However, the recording process often requires customized equipment and software, which is costly and time-consuming [11]. Consequently, these interface issues represent a serious bottleneck for most robotics applications, especially in deep learning environments, which typically require long training periods and large amounts of empirical data. To address these issues, this study proposes a novel interface utilizing a real-time hand tracking solution. The proposed system simply captures and processes an expert's hand movements and then reproduces the movements by a corresponding robotic manipulator. Performance results show that the proposed system is effective in imitating expert demonstrations with low latency and acquiring sophisticated demonstration data.

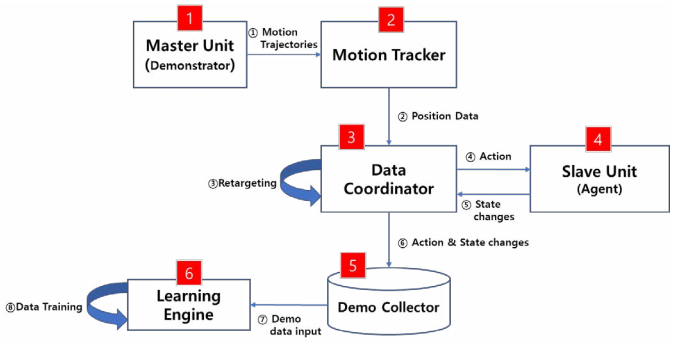

Fig. 1. The workflows of the proposed system.

II. The Proposed Demo Collection System

Figure 1 shows the workflows of the proposed system, which tracks, collects, and transforms task demonstration data to learn how an expert handles a task, has an agent reproduce it to obtain demonstration data for training, and performs imitation learning based on that data. In the figure, the master unit demonstrates the behavior of handling a task as an expert with the ability to handle manipulation or manufacturing tasks. The motion tracker tracks specific keypoints and extracts location and orientation information for the keypoints while the master demonstrates their behavior. The data coordinator handles the

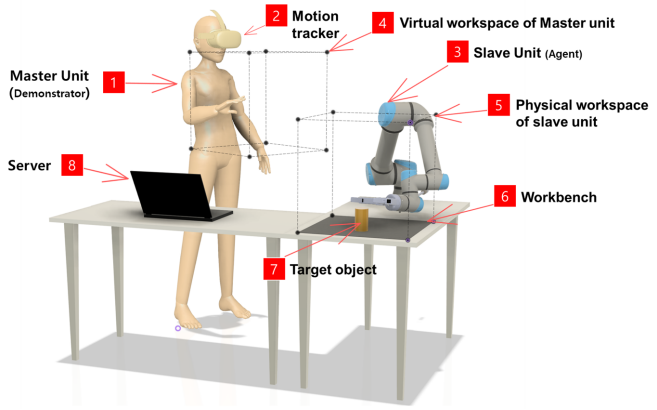

Fig. 2. Configuration overview of the proposed system.

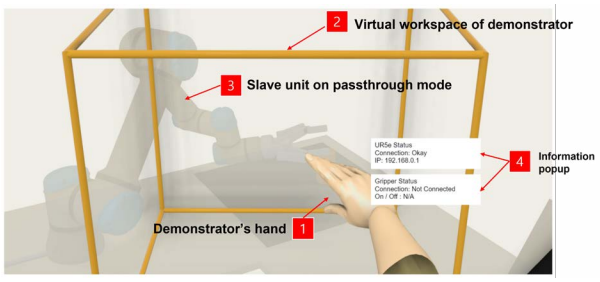

Fig. 3. Layout of user view.

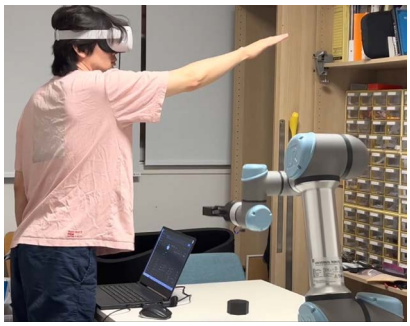

Fig. 4. Participant performing robot manipulation.

retargeting process, which converts the information collected by the motion tracker into a form suitable for controlling the slave unit so that it can act in a way that mimics the behavior of the master unit. The slave unit is an agent, such as a robot or robotic manipulator, that replicates the behavior of the master unit using data processed by the data coordinator. The status and behavior information of the slave unit is stored in the data collector. The learning engine takes the stored demonstration data as input and learns policies that mimic the master unit's behavioral policy.

Figure 2 depicts the configuration and operating environment of the proposed system. The proposed system is simple to use. The demonstrator simply wears a motion tracker such as a head-mounted display (HMD) with passthrough capabilities, and manipulates with their hands in a virtual workspace visualized through the HMD. The motion tracker basically utilizes passthrough augmented reality (AR) and 3D hand tracking solutions as human-robot interface. Passthrough AR [12] is a technology that uses stereo cameras and a standard virtual reality display to display the real-world environment around the user. The motion tracker captures images of the virtual workspace through attached cameras and analyzes them in real-time to track the spatial coordinates and orientation of the demonstrator's hand keypoints, such as fingertips and finger joint, as well as hand gestures. The proposed system converts keypoint coordinates of the tracked hand into joint position parameters using an inverse kinematic solver [13]. The server delivers the converted data to the agent, and the agent reproduces the same behavior as the user based on the behavior control data received.

Figure 3 shows the layout of user view that is visualized to the user when demonstrating task processing. During a demonstration of an action, if the user demonstrates the action in a blank space without any additional information, the behavior may be unnatural or inaccurate. To compensate, a real agent that mimics the user's behavior performs the action demonstration, and the user can see the action in real time. The information visualized to the user is a kind of mixed reality content that includes a 3D image of the virtual workspace, state information from the agent, and real-time images from the motion tracker's camera and the user's hand. For example, if the user's hand makes a shifting or grasping gesture, the agent will quickly do the same. These visualization applications can be used to increase the intuitiveness and accuracy of object manipulation when demonstrating tasks to users.

III. Implementation and Performance

The implemented system works as follows. The demonstrator wears a motion tracker such as a HMD with passthrough capabilities. Then, the attached camera collects images of the workspace and the demonstrator's hand tracking information and sends them to the server. Based on the acquired information, the server generates movement control commands that cause the slave unit to mimic the demonstrator's movements. These commands are delivered to the agent in real time and the agent operates in the physical workspace and can push, pull, grab, and move target objects on the workbench.

During these task demonstrations, the agent's behavioral data (e.g., tip position of the robot hand, angles, angular velocities, spatial coordinates, orientation, rotation, etc.) and observation data (e.g., angles, angular velocities, spatial coordinates, orientation, rotation, etc.) are stored on the server. These system features were implemented in Unity 3D, and a 6-DoF robotic manipulator with a two-finger gripper was used for the mimic agent. The proposed system is capable of controlling joint angles up to 500 times per second.

Figure 4 shows a participant demonstration of the implemented system. Demonstration videos of the proposed system are available at https://sites.google.com/view/ictc2023choi.

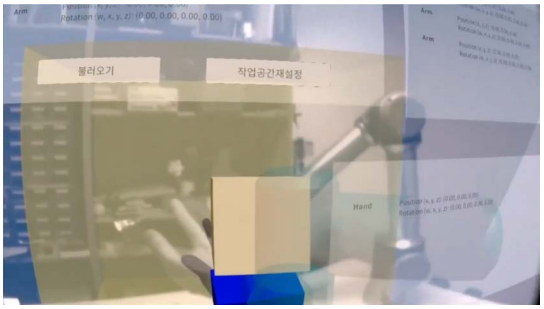

Fig. 5. User view of the implemented system.

Fig. 6. Workflows of the proposed method.

Unrestricted mimicking of user behavior can cause safety issues. Therefore, to ensure safety, the implemented system utilizes hand gesture recognition to synchronize the slave agent when the user performs a certain hand gesture. Specifically, the implemented system required that the user's hand be fully extended and that the hand and the real robot's gripper overlap for a period of time to synchronize, as shown in Figure 5.

Since the demonstrator is performing the demonstration while being aware of the slave agent's movements, the demonstrator is sensitive to motion delays. The implementation system is driven by a variety of processing, including perception, tracking, transformation, networking, and device control, and these complex processes will have a significant impact on the delay between the demonstrator's behavior and the slave agent's behavioral imitation. The end-to-end behavioral delay of the system was estimated from the demonstration video. Specifically, we took a 30fps video of the system running and measured motion latency as the difference between the frame where the human starts to move and the frame where the robot actually moves. In a wired local network environment, as shown in Figure 6, the robot was found to move after an average of 7 or 8 frames, which translates to a delay of about 233 to 267 ms. It is known that humans are generally uncomfortable with delays of more than one second [14]. Consequently, the proposed system can be utilized not only as a demo collector for imitation learning, but also as a manipulative task execution avatar operating in a remotely accessible place.

IV. Conclusion

This work proposed an alternative demo collection interface for imitation. Performance results showed that the proposed system not only mimics expert demonstrations with low latency, but is also effective in collecting sophisticated demonstration data. The presented work has the advantage of being able to easily transfer human motions to robots and to learn a wider range of behavioral policies more flexibly, but has the disadvantage of requiring a HMD with motion tracking and passthrough capabilities. Potential extension of the presented work is to extend it to a humanoid robot and apply it to a real-world settings with noisy trajectories.

Acknowledgment

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways]

References

[1] T. Osa, et al., "An Algorithmic Perspective on Imitation Learning," arXiv preprint, abs/1811.06711, 2018.
[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in Proc of ICML, 2004.
[3] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel and L. Pinto, Visual imitation made easy, in Proc. of *Conference on Robot Learning (CoRL)*, 2020.
[4] M. M. Coad, L. H. Blumenschein, S. Cutler et al., Vine robots: design, teleoperation, and deployment for navigation and exploration, *IEEE Robotics and Automation Magazine*, 2019.
[5] J. D. Sweeney and R. Grupen, A model of shared grasp affordances from demonstration, in Proc. of *IEEE-RAS International Conference on Humanoid Robots*, pp. 27–35, 2007.
[6] A. Santara, A. Naik, B. Ravindran, D. Dipankar, D. Mudigere, S Avancha and B. Kaul, RAIL: Risk-Averse Imitation Learning, in Proc. of *NIPS*, 2017.
[7] C. Mutzenich, S. Durant, S. Helman, and P. Dalton, Updating our understanding of situation awareness in relation to remote operators of autonomous vehicles, *Cognitive Research: Principles and Implications*, vol.6, no.1, pp.9, 2021.
[8] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo and S. Ivaldi, A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot, *IEEE Robot. Autom. Mag.*, vol.26, no.4, pp.73-82, 2019.
[9] S. Wrede, C. Emmerich, R. Grunberg, A. Nordmann, A. Swadzba and J. Steil, A user study on kinesthetic teaching of redundant robots in task and configuration space, *Journal of Human-Robot Interaction*, vol.2, no.1, pp.56–81, 2013.
[10] I. Lenz, R. Knepper and A. Saxena, Deepmpc: Learning deep latent features for model predictive control, *In Robotics: Science and Systems*, 2015.
[11] S. Sharma, S. Verma, M. Kumar, and L. Sharma, Use of motion capture in 3D animation: Motion capture systems, challenges, and recent trends, in Proc. of *IEEE COMITCon*, pp. 289–294, 2019.
[12] G. Chaurasia et al., Passthrough+: Real-time Stereoscopic View Synthesis for Mobile Mixed Reality, in Proc. of *ACM Comput. Graph. Interact. Tech.*, vol.3, no.1, article 7, 2020.
[13] Universal Robots RTDE C++ Interface, Available at: https://sdurobotics.gitlab.io/ur_rtde/
[14] S. Ellis, K. Mania, B. Adelstein and M. Hill, Generalizeability of latency detection in a variety of virtual environments, in Proc. of *the Human Factors and Ergonomics Society Annual Meeting*, vol.48, 2004, pp. 2632–2636.