Deploying Imitation Learning using VR Hand Tracking in Robot Manipulation Tasks

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

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

Jun Hee Park
Industry and IoT Intelligence Research Dep.,
ETRI, Korea
juni@etri.re.kr

Abstract

Imitation learning is emerging as one of the promising approaches for enabling robots to acquire abilities. Since imitation learning provides methods of learning policies through imitation of an expert's behavior, it requires sophisticated and sufficient expert behavior trajectories. However, current interfaces for imitation such as kinesthetic teaching or remote operation considerably restrict the ability to efficiently collect diverse demonstration data. To address this challenge, this work proposes an alternative interface for imitation, which can easily transfer human motions to robots while simplifying the demo collection process using a VR-based hand tracking solution. In addition, a novel method that performs data augmentation on expert trajectories is proposed to improve imitation performance. Experimental results showed that the proposed method is effective in collecting expert demonstrations and augmenting the expert trajectories and successfully completing robot manipulation tasks.

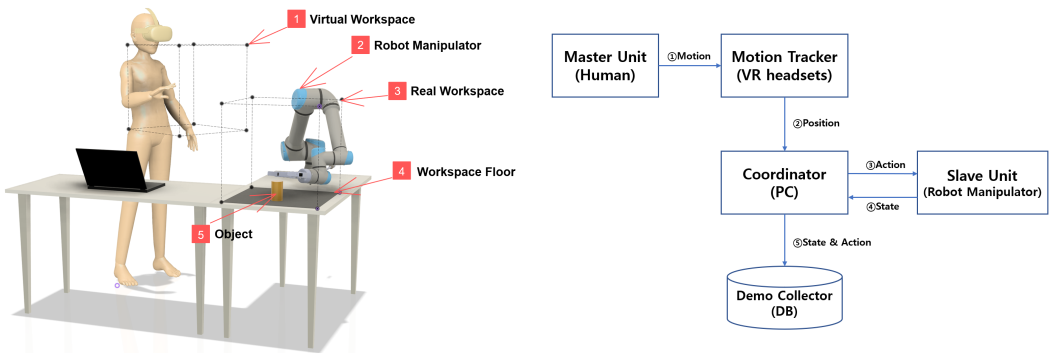
1 Introduction

Deploying imitation learning models is the process of imitating the way real-world problems are solved and placing the trained imitation model into a real-world environment where it can be used for its intended purpose [1]. A major bottleneck in current imitation learning deployment is the use of interfaces such as kinesthetic teaching or remote operation [2].

Kinesthetic teaching, in which experts physically guide the robot by applying force to it, is an effective way for non-experts to enable robot configuration and manipulation for demonstration collection [3-4]. However, this method is rather cumbersome and requires the operation of manipulating each movement one by one, so it is not suitable for complex manipulation tasks [5]. Remote operation (or teleoperation), in which an expert remotely controls the robot using control interfaces, has been successfully applied to a variety of robotic tasks, including navigating robot [6], grasping objects [7], driving cars [8] and even humanoid robots [9]. However, it is still challenging to devise such interfaces for robotic manipulation.

To deal with this challenge, this study proposes a novel imitation learning system to effectively derive imitation models in a real-world environment. First, this work proposes an alternative interface using a VR-based real-time hand tracking solution. The expert's hand movements are simply captured and processed through the proposed system, and subsequently the movements are reproduced by

NeurIPS 2022 Workshop on Challenges in Deploying and Monitoring Machine Learning Systems.



(a) System layout

(b) General workflow

Figure 1: Overview of the proposed robot manipulation system for demo collection

the corresponding robot manipulator. In this process, the movement trajectory of the robot is recorded as learning data for imitation. Second, this work proposes a novel method that performs data augmentation on expert trajectories. Imitation learning generally tends to succeed with large amounts of demo data [10]. This often results in a serious bottleneck for most robotic manipulation applications, especially in deep learning environments that typically require a long learning period and a large amount of experience data [11]. To deal with this challenge, the proposed method combines behavior cloning (BC) and inverse behavioral cloning (IBC) in a complementary manner and thus augments state-action pairs on a given expert trajectory.

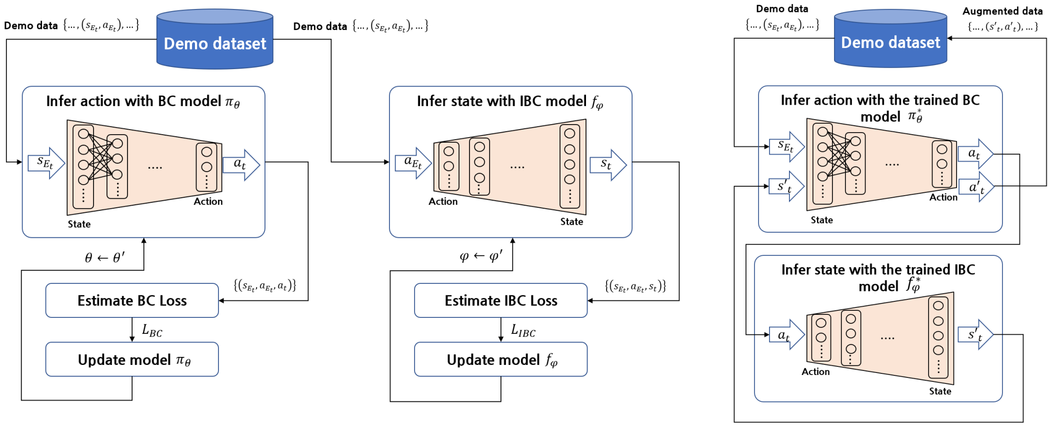
2 The Proposed Demo Collection System

Figure 1 shows an overview of how the proposed system controls the robot manipulator by replicating human movements. The expert wears a VR headset that supports passthrough features [12] and performs hand manipulations in a virtual workspace. Passthrough provides a real-time 3D visualization of the physical environment in the VR headset. The proposed system applies the passthrough features so that experts can simultaneously demonstrate while recognizing that the physical robot is working.

Furthermore, to drive AR experiences, the proposed system utilizes a real-time hand tracking solution [13] that allows hands to be used as input methods. Using four monochrome cameras mounted on the VR headset, the hand tracking solution detects to each hand in every input image and tracks the position of a specific keypoint of the hand, such as the fingertip or finger joint, in real time. The proposed system converts keypoint coordinates of the tracked hand into joint position parameters using an inverse kinematic solver [14]. The joint position parameters to move are sent to the robot manipulator and consequently the demonstrator's movements are replicated by the robot. The proposed system is capable of joint angle control up to 500 times per second. At regular time intervals, the current joint position values and the input control commands are stored as expert's state-action pairs for imitation. 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.

3 The Proposed Demo Augmentation Method

Although the proposed system helps to transfer human motions to robots, it is still challenging to obtain a sufficient amount of data through human demonstrations. To deal with this, this work proposes a novel demo augmentation method consisting of two stages. In stage 1, BC and IBC algorithms are trained separately on given demo data. BC can provide a reasonable imitation policy that updates the policy model to reduce the difference between the inferred action and the expert action according to the state-action pairs in the demo data [15]. However, BC has a limitation in that it cannot efficiently cope with various environmental states because the scope of learning is limited to the expert trajectories included in the demo data [16]. To deal with this, the proposed method introduces IBC, which provides a stochastic model that restores a given state from the expert's


Demo data $\{\dots,(s_{E_t},a_{E_t}),\dots\}$
Demo dataset
Demo data $\{\dots,(s_{E_t},a_{E_t}),\dots\}$
Infer action with BC model π_θ
s_{E_t}
a_t
State
Action
Infer state with IBC model f_φ
a_{E_t}
s_t
Action
State
$\theta \leftarrow \theta'$
$\varphi \leftarrow \varphi'$
$\{(s_{E_t},a_{E_t},a_t)\}$
Estimate BC Loss
L_{BC}
Update model π_θ
$\{(s_{E_t},a_{E_t},s_t)\}$
Estimate IBC Loss
L_{IBC}
Update model f_φ
Demo data
$\{\dots,(s_{E_t},a_{E_t}),\dots\}$
Demo dataset
Augmented data
$\{\dots,(s'_t,a'_t),\dots\}$
Infer action with the trained BC model π^*_θ
s_{E_t}
s'_t
a_t
a'_t
State
Action
Infer state with the trained IBC model f^*_φ
a_t
s'_t
Action
State

(a) Stage 1: BC and IBC model training

(b) Stage 2: Demo augmentation

Figure 2: Two-stage architecture of the proposed demo augmentation method

action. In terms of network structure, BC and IBC have inputs and outputs in opposite directions. Consequently, IBC can provide imitators with some flexibility to experience new states that are not included in the demo data, allowing them to learn a rather extended coverage of behavioral policies.

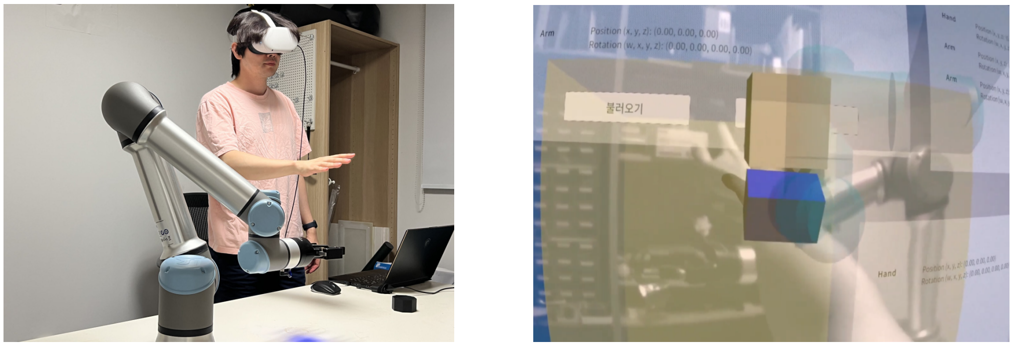
Figure 2 illustrates the two-stage architecture of the proposed method. In Figure 2a, the BC model π_θ and the IBC model f_φ are neural network modules representing the action policy and the state restoration, respectively. In BC, the learner follows the randomly sampled mini-batch of demo trajectories, concurrently selecting the action a_t to be taken in the demo state s_{E_t} using π_θ, followed by the calculation of L_{BC} which is the difference between the inferred action a_t and the demo action a_{E_t}. Thus, L_{BC} is defined as $L_{BC} = \sum_{(a_{E_t},a_t)\in A} \|a_{E_t} - a_t\|_2^2$. Then, the parameter θ of the BC model π_θ is updated to reduce the loss L_{BC}. In contrast to BC, IBC is the process of updating the parameter of the state restoration model to reduce the loss L_{IBC} between the demo state s_{E_t} and the restored state s_t from the model. Thus, L_{IBC} is formulated as $L_{IBC} = \sum_{(s_{E_t},s_t)\in S} \|s_{E_t} - s_t\|_2^2$. Then, the parameter φ of the IBC model f_φ is updated to reduce the loss L_{IBC}.

Figure 2b describes the demo augmentation process using the trained models. It starts with the inference of action according to the demo state s_{E_t} using the trained BC model π^*_θ. The next step is the state restoration, in which the state s'_t caused the inferred action is predicted using the trained IBC model f^*_φ. Finally, using π^*_θ once again, the process of determining the action a'_t to be executed in the state s'_t is performed. Upon completion of this procedure, consequently, a new state-action pair (s'_t, a'_t) different from the existing demo data (s_{E_t}, a_{E_t}) is obtained.

4 Experimental Results

The performance of the proposed system was tested using a 6-DoF robot manipulator with a two-finger gripper and analyzed for two types of robotic manipulation tasks, i.e., reach-target and pick-and-place tasks. The task environments for evaluation were modeled using the robot simulator CoppeliaSim (formerly V-REP [17]) and PyRep, a toolkit for robot learning. Each task state is represented by the manipulator's six joint angles and the x, y, and z coordinates of the target. Each control action of the manipulator is represented by the next joint angle, which is a six-dimensional vector in the range of -2π to 2π radians, and the amount of gripper opening between 0 and 1.

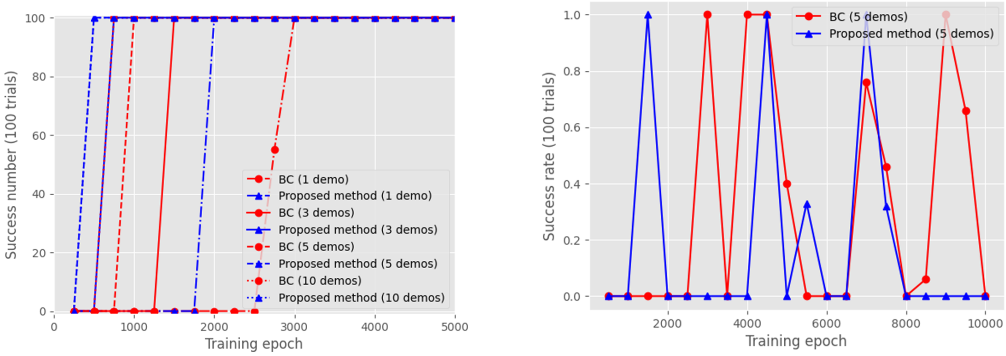
Figure 3 shows a participant demonstration of the implemented system. Demo data for training were collected through the proposed robot manipulation system. Demonstration videos of the proposed system are available at `https://sites.google.com/view/dmml2022choi`. As shown in the video, in a wired local network environment, the robot followed the user's behavior within an acceptable low latency. Consequently, the proposed system can be effectively used to collect responsive and precise demo data.


Arm
Position (x, y, z): (0.00, 0.00, 0.00)
Rotation (w, x, y, z): (0.00, 0.00, 0.00, 0.00)
불러오기
Hand

(a) Participant performing robot manipulation

(b) User's view with the passthrough feature

Figure 3: Demo collection using the proposed system



(a) Reach-target task

(b) Pick-and-place task

Figure 4: Training epochs required to obtain a task completion model

The collected demo data was trained with the pure BC algorithm. In order to show that the proposed demo augmentation is effective in reducing the training epoch required for a well-trained model, the augmented data is also trained with the same BC algorithm. The dataset for training consisted of several task execution demos. A single demo trajectory for the tasks consisted of between 32 and 78 state-action pairs. A simple MLP network was used as the learning model and trained at a learning rate of 10^{-3} using the Adam optimizer. The BC and IBC models for demo augmentation were trained up to 10^4 epochs. It was tested 100 times to evaluate policies trained for the tasks.

Figure 4 shows the training results for the reach-target and pick-and-place tasks. The goal of the reach-target task is to move the gripper tip of the manipulator from its initial position to a predetermined target position. Consequently, both the pure BC algorithm and the proposed method derived well-trained policy models as shown in the figure 4a. The figure also shows that the proposed method improved the performance level rapidly compared to the pure BC algorithm. The goal of the pick-and-place task is to grab a target placed on the table and move it from an initial position to a predetermined target position. Compared to the reach-target task, this task is more challenging because it has a much larger state space and a longer task trajectory. Figure 4b shows that the pure BC algorithm for the task did not provided any noticeable improvement for the first 2500 epochs. On the other hand, the proposed method derives a task completion model with the first 1750 epoch training.

5 Conclusion

This work proposed both an alternative demo collection interface for imitation and a novel method for performing data augmentation on behavioral trajectories. 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 VR devices for hand tracking and computational efforts for data augmentation. The effectiveness of the proposed system was demonstrated through evaluation in two robotic manipulation tasks. 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. [22ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways]

References

[1] A. Paleyes, R. Urma and N. D. Lawrence, Challenges in Deploying Machine Learning: a Survey of Case Studies, *ACM Comput. Surv.*, vol. 1, no. 1, article 1. 2022.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] I. Lenz, R. Knepper and A. Saxena, Deepmpc: Learning deep latent features for model predictive control, *In Robotics: Science and Systems*, 2015.

[10] J. Ho and S. Ermon, Generative adversarial imitation learning, in Proc. of *NIPS*, pp. 4565–4573, 2016.

[11] G. Qi and J. Luo, Small data challenges in big data era: a survey of recent progress on unsupervised and semi-supervised methods, *arXiv:1903.11260*, 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] S. Han et al., MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality, in Proc. of *ACM Trans. on Graph.*, vol.39, no.4, pp.87:1-13, 2020.

[14] Universal Robots RTDE C++ Interface, Available at: `https://sdurobotics.gitlab.io/ur_rtde/`

[15] J. Choi, H. Kim, Y. Son, C. Park and J. Park, Robotic Behavioral Cloning Through Task Building, in Proc. of *IEEE ICTC*, 2020.

[16] E. Jin and I. Kim, Hybrid Imitation Learning Framework for Robotic Manipulation Tasks, *Sensors*, vol. 21, no. 10:3409, 2021.

[17] E. Rohmer, S. P. Singh and M. Freese, V-rep: A versatile and scalable robot simulation framework, in Proc. of *IEEE IROS*, pp. 1321–1326, 2013.

[18] R.D. Madder et al., Network latency and long-distance robotic telestenting: exploring the potential impact of network delays on telestenting performance, in *Catheterization and Cardiovascular Interventions*, 2019.

[19] P. Farajiparvar, H. Ying and A. Pandya, A Brief Survey of Telerobotic Time Delay Mitigation, in *Front Robot AI.*, 2020

A Notation and Problem Description

The proposed work considers robot manipulators acting within the broad framework of a Markov Decision Process (MDP), consisting of the tuple (S, A, D, T, π). S and A represent the state space and the action space, respectively, and both are defined as high-dimensional continuous spaces. A dataset of expert demonstrations is given as a set of trajectories $D = \{\tau_{E_1}, ..., \tau_{E_m}\}$, where each trajectory $\tau_E = \{(s_{E_1}, a_{E_1}), ..., (s_{E_n}, a_{E_n})\}$ is a sequence of state-action pairs. In this work, it is assumed that the state and the action spaces of the tasks demonstrated by the expert are the same as those of the robotic learning tasks. Thus, any state s and action a in D are assumed to $s \in S$ and $a \in A$. T denotes the assigned robotic manipulation task, which is expressed by pair (c_{init}, c_{goal}) of the initial state condition c_{init} and the goal state condition c_{goal}. π denotes the stochastic policy that determines the action a_t to be taken by the manipulator in the state s_t at time step t to achieve the assigned task T. The goal of this work is to effectively train a policy π that can imitate the expert behavior by augmenting a given amount of demonstration.