IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 8 Issue 11, November 2021
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

Interaction in an Augmented Reality Environment Using Reinforcement Learning

Sung-Jun Lee[1] and Sang-Young Cho[1]

[1] Major in Software Convergence, Hankuk University of Foreign Studies,
Yongin, Gyeonggi 17035, Korea

Abstract
The biggest feature of Augmented Reality (AR) technology is that it can simultaneously express reality and virtuality in one place. AR technology allows real-world users to interact with virtual objects, which creates complex user experiences that were limited to real or virtual space. This paper uses a hand-based interaction technique to enrich user experience in augmented reality, and reinforcement learning is conducted to allow objects in augmented reality to learn hand movements and take specific actions. By the reinforcement learning, the complex interactions between users and virtual objects in augmented reality can be achieved. This paper shows that the hand movements of users can control the behaviors of virtual objects in various ways.
Keywords: *Augmented Reality, Object Interaction, Reinforcement Learning, Learning Environment.*

1. Introduction

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory [1]. The application of AR technology, which is widely known to people through Pocketmon Go, is expanding across different sectors [2]. Recently, there has been an increase in the application of AR in various fields such as architecture and interior design [3]. Furthermore, studies have been carried out to integrate AR to enhance the game experience [4].

Recently, the need for interaction with augmented objects has been increasing, and a lot of interaction studies have been underway [5]. Of the many interactive methods, the most intuitive is using the hand. The hand, along with the eyes, is one of the most frequently used body parts and functions to share emotions or induce specific behaviors through interactions such as grabbing, stroking, or touching objects. For example, petting a dog makes a dog wag its tail and pointing to a certain sign leads people to look at the sign.

In this paper, hands, the easiest and most effective way to interact in augmented reality, were used as a medium for interaction. We augment the hands of the real world to represent them as physical objects in the virtual environment and implement them so that the hands of the real person can directly access the augmented reality to experience the manipulation of virtual objects. In order to express a person's hand in a virtual environment, a part of an open source game produced using Google's MediaPipe was used [6]. Google's MediaPipe is an open-source program that helps you easily extract the shape of your hand from an image [7].

Reinforcement learning refers to the method in which the computer learns to maximize rewards by observing specific data in a given environment rather than to artificially designed behavior [8]. It does not force people to take specific actions in the process of reinforcement learning. The actions the machine learns by itself act as an active action on the situation in the environment. This active action enables interaction with the machine using the designed input and allows it to feel two-way communication with the machine. We aim to take advantage of the interaction aspects of these reinforcement learning. In this paper, we implemented reinforcement learning using Unity's ML-agent package [9]. It is designed to observe the value of the hand position vector in reinforcement learning so that the interacting object can learn in a way that rewards the movement of the hand.

This paper is organized as followings: Section 2 describes the learning environment of reinforcement learning including the observation, behavior and reward system of learning targets. It also presents the learning progress method and augmented reality system implemented. Section 3 shows the experimental methods and results. Finally, the conclusions and future research directions are described in Section 4.

2. Design of Reinforcement Learning and Augmented Reality System

In augmented reality, hand behavior is used as input. In response to this input, the interaction in which the animal performs a specific motion is tested. Reinforcement learning is used to learn the performance of animal movements corresponding to hand movements. Figure 1 shows the process of constructing an augmented reality environment by applying reinforcement learning.

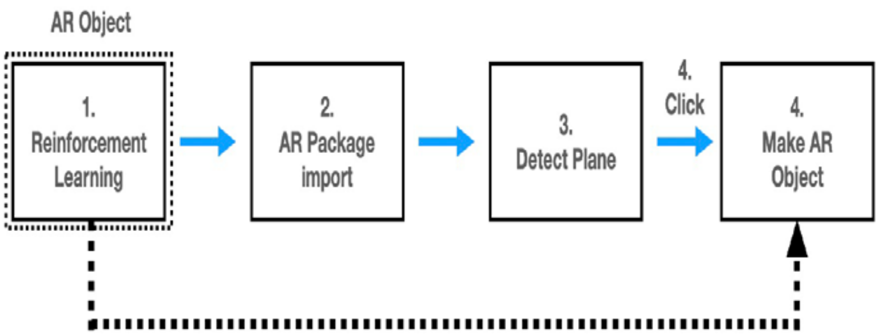

Fig. 1 System design process

Reinforcement learning is conducted by creating objects for hand and receiving commands within the game environment using the Unity game engine. After the learning to be implemented is completed, Unity's AR Foundation 3.1.3 and ARCore 3.1.3 packages are used to convert the game environment into an augmented reality environment [10]. Afterwards, augmented reality camera was made to recognize the hand object. When the recognized hand object clicks on one of the command objects, the hand and the command object are augmented.

2.1 Reinforcement Learning Design

2.1.1 Design of Learning Environment

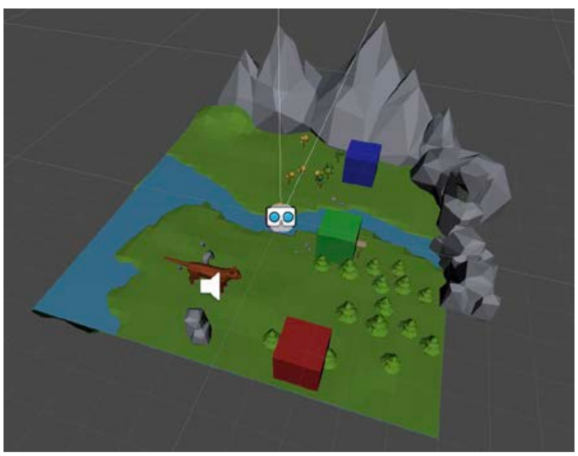

Fig. 2 Learning Environment of hand and command objects

Reinforcement learning design is largely divided into two parts: 1. learning environment and 2. design for observation, behavior, and reward of objects to be learned. In order to recognize the hand when augmented reality is applied, information about the hand must be generated in the learning environment in advance. To learn hand information, we create a cube with a physical engine and tag it 'Hand'. The animal to interact with using hands is a cat. In order to teach the cat three commands: sleeping, running, and sitting, three cube objects larger than the "Hand' object are prepared. If a cube with the 'Hand' tag contacts an object for a specific command, the corresponding integer value of 1(sleeping), 2(running), and 3(sitting), respectively, is stored in the 'order number' variable, and in other cases, 0 is stored. The learning environment is shown in Figure 2. There are three command box of red, green, and blue and also a cat. When a colored box is touched by hand in the actual AR environment, the red box is set to run, the green box is set to sleep, and the blue box is set to sit.

2.1.2 Design of Observation, Behavior, and Reward of Objects

The cat 3D model of the Unity Store was used as the object to be learned, and additional animations of sleeping, running, and sitting were used. The behavior of the cat is related to a total of eight values such as triggers that activate sleep, run, sit motions, a Boolean value that verifies the execution of animation, the 'order number' variable, and x, y, and z coordinate values of the 'Hand' object. The location values for the 'Hand' object are used to determine whether the object touches the command objects. In the case of cat behavior, if the animation execution state is False, it is possible to take one of four actions: sleeping, sitting, running, and staying still.

For the reward system of the reinforcement learning, if the 'order number' value is 1, i.e., the box for sleeping command is touched, the reward system gives 1 point reward when the sleeping animation is executed and –0.1 point reward if other actions are taken. The same reward value is applied to the 'order number' value 2 (running) and the value 3 (sitting). It was designed to give 0.2 point reward for executing the staying still animation when the order number is zero and –0.1 point reward for other actions. The reason why the compensation for staying still was designed low to 0.2 is to induce the natural movement of the cat by setting it up to perform various actions without instructions.

2.1.3 Learning Progress and Parameter Design

The Boolean value, which can indicate whether the cat's animation is executing or not, is activated six seconds after the animation was executed and induced not to simultaneously execute the animation by command. The object with "Hand" tag was allowed to contact command-inducing objects with the 3/7 probability every 7 seconds and were placed in other areas with the 4/7 probability. If the 'Hand' object contacts a box that gives instructions, it is programmed to contact any random inner part of the box set at 5cm so that it can recognize whether there is contact or not. The learning process is designed to facilitate the PPO algorithm [11] that is simple compared to other algorithms. The total step of reinforcement learning is more than 3 million times, and the hyperparameters used for reinforcement learning are as follows.

Hyperparameter
- batch size : 128
- buffer size : 1024
- learning late : 0.0003
- beta : 0.01
- epsilon : 0.2
- lambd : 0.95
- num epoch : 3
- learning rate schedule : linear

Figure 3 shows a snapshot of the learning process in the developed learning game environment.

2.2 Implementation of Augmented Reality

The objects of the Unity game environment in Figure 2 where reinforcement learning was conducted are produced as Prefab so that they can be used as objects in an AR environment. (Prefab is a concept used in Unity and means a finished assembly.) These Prefab objects are used to construct our AR environment.

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 8 Issue 11, November 2021
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

Fig. 3 The snapshot of the learning process in the learning game environment

AR Foundation and ARCore packages are imported to the development environment to build an interactive AR environment using hands. Then, we create AR Session Origin with AR Camera as a sub-component to identify the location where AR can be configured. AR Session Origin has a coordinate value of (0, 0, 0) in the augmented reality environment. Afterward, an AR Session is created to manage state changes, such as frame rates for image recognition of augmented reality, to configure an environment in which AR can operate.

In order to become a desired augmented reality, a virtual environment in which reinforcement learning has been conducted must be added to the actual image shown through the camera. To this end, a plane in which a virtual environment may be installed on the displayed image must be selected. When AR Camera recognizes a plane, visualization is performed so that users can check whether the recognized plane through AR Point Cloud is able to be used for the ground of the virtual environment. When the user clicks on the recognized plane, Prefab, which was produced in the game environment where reinforcement learning was conducted, and Prefab, the shape of the hand using Google's Media Pipe, are implemented on the real view of AR camera.

3. Experiment Method and Result

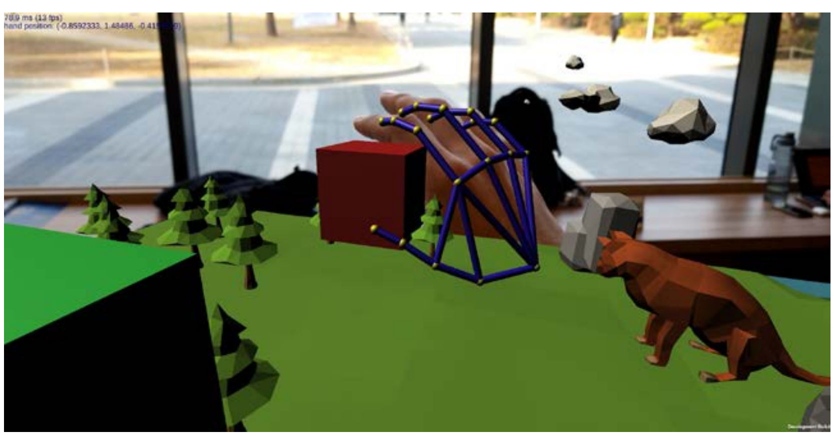

Fig. 4 The augmented reality environment for experiment

The experiment was conducted with a Samsung Galaxy 8 smartphone model that uses the Android operating system. The developed augmented reality appears on the camera display and recognizes the hand for interaction. The experiment used two methods: not giving commands for more than 10 seconds and giving commands every 3 to 10 seconds. In the first experiment method, if the command was not issued for 10 seconds, it was seen that the cat basically was motionless and randomly ran, slept, or sat down. Figure 4 shows the experimental screen of the first method. For the second method, if the green box was touched, i.e., instructed to sleep, the actual sleeping action was performed, and if the other box was touched within a short time less than six seconds, the following instructions were not carried out. However, if user touched another box with user's hand after more than six seconds, it would function normally. It can be seen from the data of TensorBoard in Figure 5, which shows the results of reinforcement learning, that the actual learning took place normally.

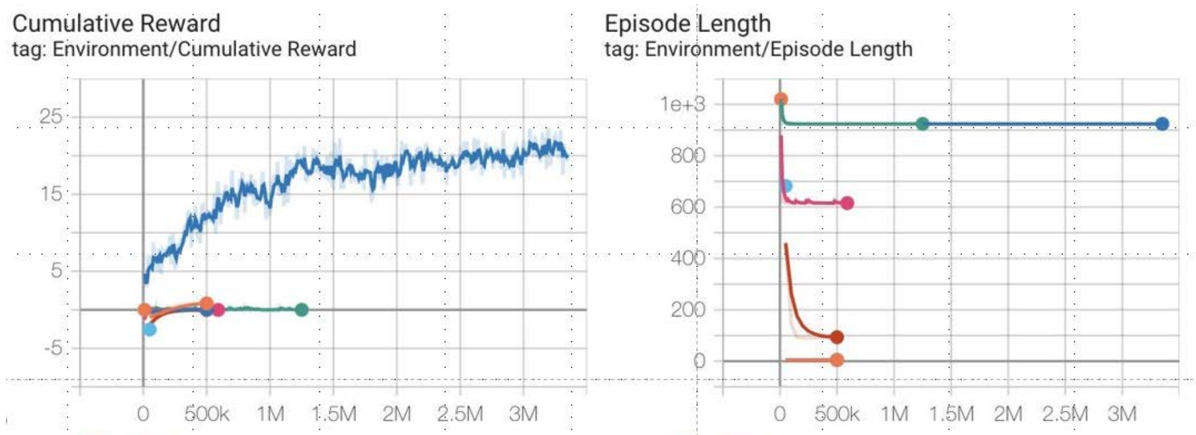

Fig. 5 The result of reinforcement learning

4. Conclusions

The purpose of this study is to incorporate reinforcement learning into augmented reality technology to enable people to interact with digital objects and have a better user-experience. In the previous researches, reinforcement learning and augmented reality have not often been used at the same time because of their different purposes and directions. However, in terms of interaction, augmented reality with visual representation as to its main function and reinforcement learning, which entrusts autonomous selection to machine behavior, are interlinked fields that can narrow the gap between people and machinery.

This paper confirmed the possibility of linking augmented reality with reinforcement learning in terms of interaction. In future studies, visually improved interactions can be expected by utilizing the function of the Lidar sensor or the function of recognizing the depth of augmented reality. In addition, it is expected that a more natural user experience can be derived by applying various algorithms used for reinforcement learning.

Acknowledgments

This work was supported by the Ministry of Science and MSIT (ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP (Institute of Information & communication Technology Planning & Evaluation) (2019-0-01816). This work was supported by Hankuk University of Foreign Studies Research Fund.

References

[1] I. A. Sicaru, C. G. Ciocianu, and C.-A. Boiangiu, "A Survey on Augmented Reality", Journal of Information Systems & Operations Management, Vol. 11, No. 2, 2017, pp. 263-279.
[2] D. W. F. Van Krevelen and R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations", The International Journal of Virtual Reality, Vol. 9, No. 2, 2019, pp. 1-20.

[3] H.-J. Kim, and J.-H. Sung, "Spatial Analysis of Mobile Augmented Reality Games", Journal of Korea Game Society, Vol. 19, No. 4, 2019, pp. 5-14.
[4] S. Razzaq, F. Maqbool, M. Khalid, I. Tariq, A. Zahoor, and M. Ilyas, “Zombies Arena: fusion of reinforcement learning with augmented reality on NPC”, Cluster Computing, 21, 2018, pp. 655–666.
[5] S. Lee, D. Baik, S. Choi, and S. S. Hwang, "Interactive Mobile Augmented Reality System using Muscle Sensor and Image-based Localization System through Client-Server Communication", Journal of the HCI Society of Korea, Vol. 13, No. 4, 2018, pp. 15-23.
[6] Single Hand Tracking: https://gitlab.com/JackieNek/single-hand-tracking
[7] MediaPipe: https://google.github.io/mediapipe/.
[8] B. J. Pandian and M. M. Noel, "Control of a bioreactor using a new partially supervised reinforcement learning algorithm", Journal of Process Control, Vol. 69, 2018, pp. 16-29.
[9] Unity ML-agent package: https://docs.unity3d.com/Packages/com.unity.ml-agents@1.0/manual/index.html
[10] Unity AR Foundation: https://unity.com/kr/unity/features/arfoundation
[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms”, arXiv preprint arXiv:1707.06347, 2017.

Sung-Jun Lee received his B.S. degree in Division of Chinese Language, Literature and Culture and Convergence Software as a second major from Hankuk University of Foreign Studies, Korea, in 2021. His research interests include reinforcement learning, intelligent UI/UX, and augmented reality.

Sang-Young Cho received his B.S. degree in Control and Instrumentation from Seoul National University, Korea, in 1988, and M.S. and Ph.D. degrees in Electrical Engineering from KAIST, Daejeon, Korea, in 1990 and 1994, respectively. He is currently a full professor with the Division of Computer Engineering in Hankuk University of Foreign Studies. His research interests include embedded system, computer architecture, and software optimization.