ICCEPM 2024 https://dx.doi.org/10.6106/ICCEPM.2024.0665
The 10th International Conference on Construction Engineering and Project Management
Jul. 29-Aug.1, 2024, Sapporo

Superimposing 3D Models on Real Scenes Based on The Reinforcement Learning using Visual Observations

Yong-Ju Lee[1]*, Linh Nguyen[2], Man-Woo Park[3]

[1] *Department of Civil and Environmental Engineering, Myongji University, Yongin 17058, Korea,* E-mail address: leetoday@mju.ac.kr
[2] *Department of Civil and Environmental Engineering, Myongji University, Yongin 17058, Korea,* E-mail address: linngn@mju.ac.kr
[3] *Department of Civil and Environmental Engineering, Myongji University, Yongin 17058, Korea,* E-mail address: mwpark@mju.ac.kr

Abstract: This research presents a method for Augmented Reality (AR) object superimposition leveraging reinforcement learning techniques to significantly reduce manual input during the exploration of digital information on construction sites. A distinctive feature of this approach is the application of a reinforcement learning neural network, trained with pairs of real and virtual view images, for AR superimposition. This approach enables the precise adjustment of the virtual camera's position and orientation within a virtual scene, aiming to seamlessly integrate AR objects into real-world views. This research initially focuses on simpler scenarios involving 2 and 3 degrees of freedom for orientation and position adjustments. The purpose is to explore the feasibility of the application through those experiments, with the expectation that the results would be interpreted positively. These initial findings highlight the promise of the suggested method in improving AR applications, especially within the construction sector, by enabling more natural and precise merging of virtual and physical objects, without requiring user intervention.

Keywords: Augmented Reality (AR), Superimposition, Reinforcement Learning

1. INTRODUCTION

As the construction industry embraces digital transformation, the demand for effective information visualization techniques increases. Alongside these developments, a notable shift towards paperless projects signifies a wider trend of change within the sector, indicating a broad spectrum of changes within the sector. In line with the establishment of digital technologies like digital twins and paperless projects, there is a critical need for effective methods for workers to access digital information from construction sites. In particular, the ability to visualize 3D models, such as BIM, is crucial for intuitive access to construction site information. The increasingly used Augmented Reality (AR) technology serves as an excellent alternative for the next-generation user interface. AR technology provides an environment where users can interact with 3D models. Visualizing 3D models through AR technology offers an intuitive experience that is incomparable to checking the shape of a structure through traditional 2D drawings or even viewing 3D models on a computer screen. These advantages can also apply to AR devices with flat displays, such as smartphones or tablets, and are especially enhanced in commercial HMD AR devices like the HoloLens or Magic Leap.

However, accurately superimposing virtual 3D models onto real scenes with AR technology presents significant challenges. For superimposition, the AR device must obtain or be able to calculate accurate relative position and orientation between the real scene and the device. Approaches for calculating position and orientation can be broadly divided into two categories: one is based on image processing that utilizes 2D information, and the other uses various sensors of the AR device to utilize 3D information. Image processing-based methods can be further divided into more detailed approaches.

The most traditional and widely used method is the photogrammetric approach, which estimates position and orientation by extracting features from 2D images. This method allows for the calculation and adjustment of the AR object or camera's position and orientation by using image features extracted from captured images. It requires setting common feature points (control points) between the real object and the 3D model or some other geometric features for similarity comparison.

A more refined method involves the use of planar image markers, such as QR Codes or ArUco, to enhance accuracy and ease of use in applications. This method employs the photogrammetric method by recognizing markers to ascertain position and orientation. It simplifies the process by eliminating the use of control points, thanks to the employment of planar markers. However, these methods still require attaching control points or markers to each structural component, which presents challenges in terms of maintenance and leads to considerable expenses for implementation on construction sites [1][2][3].

Direct approaches that utilize 3D information for calculating relative position and orientation for superimposed AR objects allow for direct comparison, which is a significant difference from methods that use 2D information. Their functionality relies on various sensors that capture the shape of the environment around the AR device and ascertain the device's position and orientation, making their performance largely dependent on sensor precision. Given the AR device's advantage of being more portable, there is a trade-off between the performance of the sensors and the size of the device. Typically, data from GNSS (Global Navigation Satellite System), IMU (Inertial Measurement Unit), ToF (Time of Flight) sensors, etc., could be utilized, but their accuracy and precision are often lower than industrial requirements for construction [4][5][6].

So, this research proposes a method that can overcome the limitations of existing methods. The proposed method utilizes a neural network trained by reinforcement learning to perform AR superimposition. That neural network is trained to take virtual view and real view images as inputs and calculate the pose information of the virtual camera displaying the AR object. It can be seen more accurately adjust the position and orientation calculated from inaccurate sensor data using the image information of the object viewed through the camera. It aims to achieve its purpose without user intervention or using control points or markers. This research is still in its early stages and will be a study to verify its applicability. The performance of the trained neural network will be reviewed to examine the applicability of the proposed method.

2. METHODOLOGY

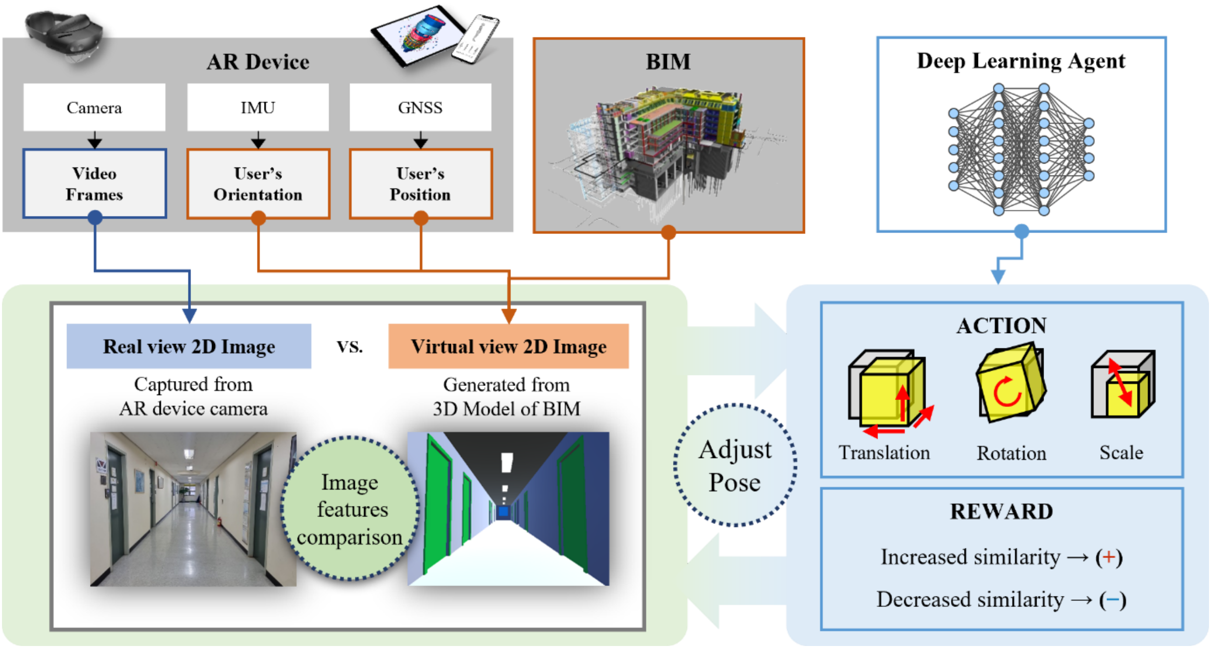


Figure 1. The framework of the proposed method

The framework of the proposed method for the automated AR superimposition based on reinforcement learning in this study is illustrated in the Figure 1. The user's scene (which the AR object will display) is captured as an image (or video) by the camera mounted on the AR device, along with synchronized sensor data such as IMU and GNSS. These sensor data are used to settle the user's position and orientation, i.e., they are utilized to construct the observation. The user's position and orientation can be used to define the user's view in ~~the~~ virtual space. Within this virtual view, a 3D virtual model generated from BIM can be placed. If the sensor data were highly accurate, this virtual model could be perfectly superimposed on the real one without additional adjustments. However, as previously noted, the precision of these sensors is insufficient, implying that they had significant errors that need to be corrected. Using a reinforcement learning network trained with image data for this error correction process is the primary approach of this research. The neural network is trained by reinforcement learning with two types of image data for training: One reflects the real view, and the other reflects the virtual view. The neural network generates a vector as output for the input images, which adjusts the virtual camera to move to the correct position (called an *action*). Therefore, it is necessary to assess whether the output of the neural network has generated the accurate results required for AR superimposition. Based on this assessment, rewards are allocated during the reinforcement learning process.

2.1. Learning process

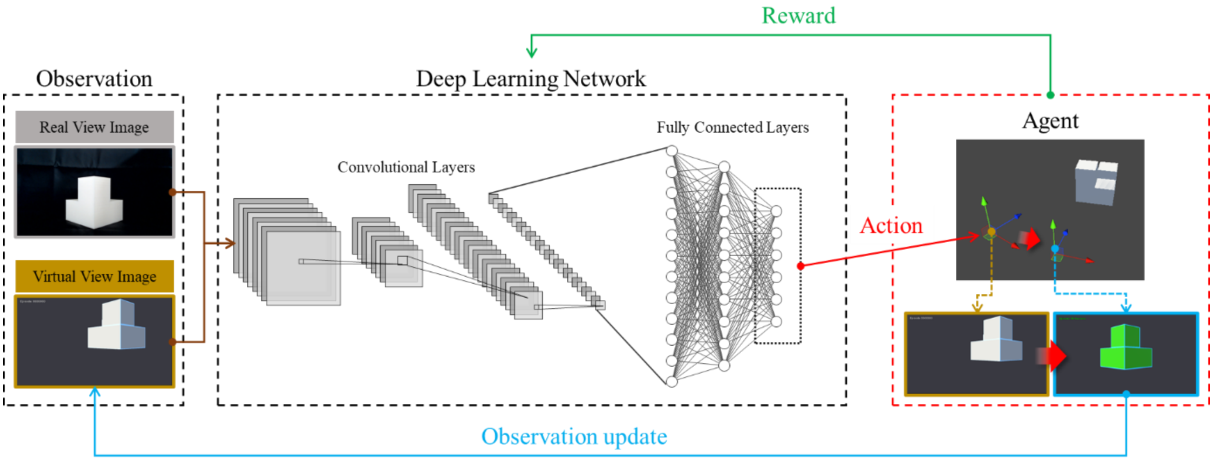


Figure 2. The process of reinforcement learning for AR superimposition

The proposed framework of the reinforcement learning-based AR superimposition method is as shown in the Figure 2. The neural network receives two images corresponding to the observation as inputs and generates a series of actions through convolutional layers and fully connected layers. These actions are used to control the agent, and the observation is updated from the changed environment, with this process repeating to adjust the weights of the network. During this process, it is necessary to determine the reward scheme, defining when and how feedback is provided for network updates. The details of each step are described below.

2.2. Observation

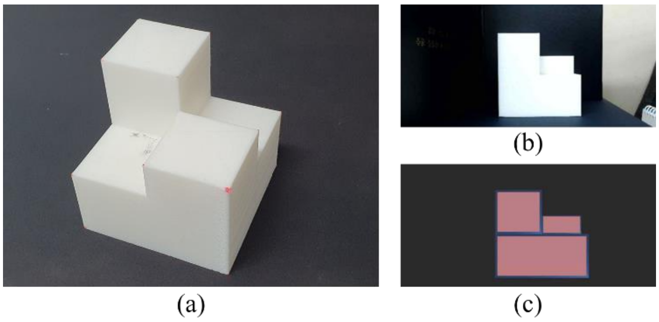


Figure 3. Real object and its images captured from webcam and virtual camera

In this proposed AR superimposition method, the images used as input to the neural network are of two types: *real view image* and *virtual view image*. In this research, real-view images are captured using the AR device's camera, which was replaced by a webcam for the experiments. Virtual view images are created from a virtual camera that mimics the real camera in a digitally constructed environment, utilizing the Unity engine for this virtual reconstruction.

The AR object for superimposition was created as a 3D model and printed using a 3D printer (as shown in Figure 3. (a)), and the same model was imported into the Unity environment to be used as a virtual object. The real view (Figure 3. (b)) observed through the webcam and the virtual view (Figure 3. (c)) implemented are shown in the figures.

The image captured through the webcam is projected with camera intrinsic parameters, so to accurately project the virtual view image, the same intrinsic parameters must be set for the virtual view camera. To achieve this, the camera calibration for the webcam was conducted to obtain intrinsic parameters, which were then applied to the virtual view camera. After this process, each image will be used as input to the neural network.

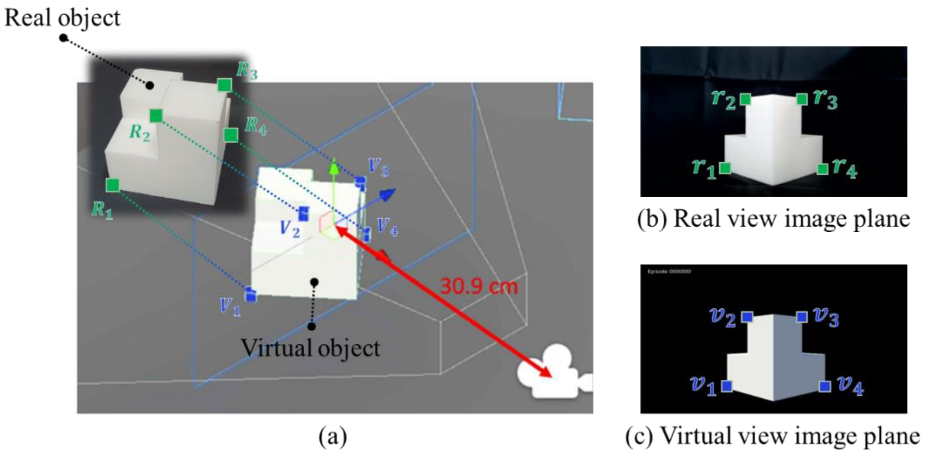
2.3. Agent and action

In reinforcement learning, the agent refers to the entity that performs actions based on the state within a given environment. In this case, the Agent could either be the virtual object or the virtual camera observing that virtual object. The primary objective of the agent is to establish a corresponding spatial relationship between the virtual object and camera, mirroring that of the real object and camera.

In this study, the virtual camera was set as the agent to calculate the pose, or TRS (Translation, Rotation, Scale) matrix of the virtual camera based on the center coordinates of the virtual object. Therefore, the actions generated according to the observation are designed to adjust the pose of the virtual camera. Actions can have a total of 9 degrees of freedom (DOF) since Translation, Rotation, and Scale each have 3 DOFs. However, assuming that the scale error can be ignored when utilizing BIM models to an AR object, we only consider changes in 6 DOFs for Translation and Rotation.

2.4. Terminal state

The situation where the agent reaches the final objective or fails, resulting in the end of the episode, is referred to as the *terminal state*. The actions of the agent leading up to the terminal state are evaluated based on their impact on reaching the terminal state and can be rewarded accordingly. Deciding when and how much reward to distribute is a crucial aspect of reinforcement learning. In this study, the agent successfully completing the episode and reaching the terminal state is described as arriving at the *goal*. The role of the agent is to translate and rotate the virtual camera so that the projection of the AR object appears identical in both the real view and the virtual view. This will be the final goal of the agent. Therefore, it is necessary to have a process for determining whether the agent has reached its goal. While user-driven observation through the device would provide the best evaluation, it eliminates automation, rendering reinforcement learning itself impractical. For reinforcement learning, the process of determining the terminal state must be automated, so that numerical values or criteria are required.



(b) Real view image plane

(a)

(c) Virtual view image plane

Figure 4. Control points on each objects and images

A method was utilized to assess alignment by designating specific points on the AR object as control points. As shown in Figure 4. (a), common points between the real and virtual objects were established, and the *goal* or *failure* was determined based on whether these points, projected onto each image (Figure 4. (b) and (c)), matched each other. The control points on the real object are denoted as R, and the control points on the virtual object as V. And their pixel coordinates on the image plane are represented as r and v. The sum of the distances between each r and v is named as *distance error* (D_E). So, when this value is ~~at~~ 0, it is considered that the two objects are perfectly aligned. Due to various noises that can introduce errors, converging this value to 0 is necessary to achieve our objective. The threshold for this value is defined as D_T. Therefore, the agent is considered to have reached the goal in the following case:

$$D_T > D_E = \sum_{i=1}^{n} |r_n - v_n| \tag{1}$$

2.5. Reward and update feedback

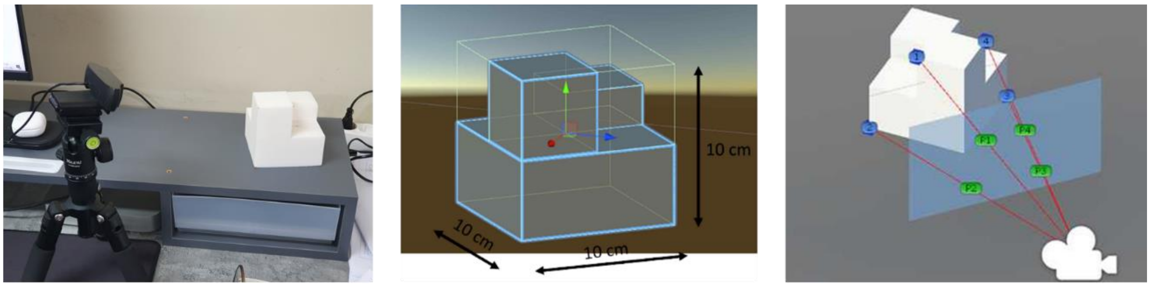
Giving rewards or penalties based on the agent's actions, whether the agent reaches the goal or during the process of reaching it, is a core element of reinforcement learning. The training performance and speed can be influenced by the reward function. In this study, various methods of giving rewards were devised and applied, and the differences in training performance and speed were examined. The reward functions set in the experiment are as follows:

1. (+1) reward is given only when the goal is reached.
2. (-1) penalty is given only in case of failure.
3. (-1 / Max step) penalty is given at each step.
4. Reward gradient

(The reward can be given in single episode range from [-1, 1].)

3. EXPERIMENT AND DISCUSSION

3.1. Experimental setup



(a) Experiment environment example (b) Target AR object 3D model (c) Control points

Figure 5. Experimental setup

The experiment was conducted on a small object measuring 10x10x10cm^3. The target object was created as a 3D model and then printed using a 3D printer (as shown in Figure 5. (a) and (b)). A webcam was assumed to be the camera of the AR device. Figure 5. (a) is an example picture of the positional relationship between the webcam and the target object. As depicted in Figure 5. (c), four control points located on the surface of the object were selected.

Agent’s actions were set as discrete actions, allowing for a maximum movement of up to 1 cm per step in each axis, and rotation was set to allow for a maximum rotation of up to 0.1 degrees per step.

Training was carried out in stages, commencing with simpler tasks involving only 2 DOFs out of a possible 6.

- 2 DOF: y, z-axis translation
- 3 DOF: x, y, z-axis translation

3.2. Experimental results and discussion

The training results were as follows:

Table 1. Training results table

Actions	Reward method	Training time (s)	Success rate	Distance Error (cm)		
				Average	Max	Min
2D	(Goal) +1 reward	15162.510	100%	0.122	0.141	0.100
	(Fail) -1 reward	14697.626	100%	0.122	0.141	0.100
	Penalty for each step	14898.325	100%	0.125	0.141	0.100
	Use distance gradient	13000.000	100%	0.114	0.141	0.000
3D	(Goal) +1 reward	14856.111	100%	0.206	0.800	0.000
	(Fail) -1 reward	12232.062	100%	0.253	0.800	0.000
	Penalty for each step	14749.303	100%	0.240	0.800	0.000
	Use distance gradient	10544.721	100%	0.313	0.800	0.000

The success rate is 100% for all training results, with the distance error averaging around 0.1 to 0.3 cm. Although the results show good performance, it is important to consider that these results only concern 2 and 3 DOFs, with low levels of training difficulty. There were some differences in training time, but no clear differences were observed in success rate and superimposition performance by different reward functions.

The results achieved from training using still images and small objects, specifically concentrating only on translation excluding rotation, are quite constrained. To successfully achieve the ultimate goal of AR superimposition, it is imperative to ensure that performance is robust enough for real-time video, in addition to still images. Additionally, in construction sites, it must be applied to much larger objects than those considered in this study. Further research is needed to overcome these limitations.

The primary objective of future research is to diversify and enhance the criteria for determining the terminal state. The current configuration method using control points has many limitations, with the most problematic being that it can hinder the automation of reinforcement learning. To apply the current method to real-time video, it must be possible to detect and continuously track control points from video streams using computer vision algorithms. However, the loss of tracking may require significant user intervention. Furthermore, if physical control points are not visible in the real-time streaming video, it could halt further training. Therefore, an alternative method for defining the terminal state is necessary. While the marker-based method mentioned in the introduction allows for solid tracking and direct use of pose information, some solutions are still necessary to overcome problems such as marker occlusion and the maintenance of markers. As an alternative, using scan data obtained through ToF sensors from the AR device is also being considered. After refining the criteria for terminal state determination, the next objective is to expand the degrees of freedom of the agent in the training for 6 DOFs.

4. CONCLUSION

This study presents a method to achieve AR superimposition through reinforcement learning using visual observations, with the aim of minimizing user intervention in accessing construction site information. An artificial neural network is trained for this purpose, utilizing two types of images as inputs: a real view image capturing the appearance of the AR object in the real scene, and a corresponding virtual view image. The neural network adjusts the position and orientation of the virtual camera within the virtual scene using these images, essentially identifying the correlation between the images and the pose of the virtual camera. To train the network, a criterion is needed to define the terminal state, and control points are used as the criterion in this study. The pose of the virtual camera encompasses a total of 6 DOFs, comprising 3 DOFs for position and three for orientation. Considering the preliminary nature of this research, training initially focused on the more manageable 2 and 3 DOFs, with experiments for 4 or more DOFs currently underway. For the training limited to 2 and 3 DOF, the goal achievement rate is 100%, with an average distance error of 0.1 to 0.3 cm. Various reward strategies were explored during the reinforcement learning ~~process~~, yet no significant differences in learning outcomes were observed. However, as the training becomes more challenging and complex, differences are expected to emerge. This study is empirical research aimed at assessing the feasibility of automated superimposition methods for various object shapes. Future research will extend to learning up to 6 DOFs and will incorporate more advanced methods for defining the terminal state beyond control points.

ACKNOWLEGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1074604).

REFERENCES

[1] E. Marchand, F. Chaumette, “Virtual Visual Servoing: A Framework for Real-Time Augmented Reality” Computer Graphics Forum, vol. 21, no. 3, pp. 289–97, 2002.

[2] A. I. Comport, E. Marchand, M. Pressigout, F. Chaumette, “Real-time markerless tracking for augmented reality: the virtual visual servoing framework” IEEE Transactions on Visualization and Computer Graphics, vol. 12 no. 4, pp. 615–628, 2006.

[3] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, N. Vitzilaios, “Fiducial Markers for Pose Estimation” Journal of Intelligent & Robotic Systems, vol. 101, no. 4, 2021.

[4] J. Polvi, T. Taketomi, G. Yamamoto, A. Dey, C. Sandor, H. Kato, “SlidAR: A 3D positioning method for SLAM-based handheld augmented reality” Computers & Graphics, vol. 55, pp. 33–43, 2016.

[5] M. Tamaazousti, S. Naudet-Collette, V. Gay-Bellile, S. Bourgeois, B. Besbes, M. Dhome, “The constrained SLAM framework for non-instrumented augmented reality” Multimedia Tools and Applications, vol. 75, no. 16, pp. 9511–9547, 2015.

[6] J. Liu, Y. Xie, S. Gu, X. Chen, “A SLAM-Based Mobile Augmented Reality Tracking Registration Algorithm,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 34, no. 01, p. 2054005, 2020.