



Natural embedding of live actors and entities into 360° virtual reality scenes

Vuthea Chheang¹ · Sangkwon Jeong² · Gookhwan Lee² · Jong-Sung Ha³ · Kwan-Hee Yoo¹

Published online: 22 September 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

This paper is concerned with techniques for *directly embedding moving objects in the real world into 360° virtual reality scenes* captured by 360° camera, DSLR camera, smartphone, and others. For more natural embedding, we present a *cylindrical mapping methodology* based on a proposed international standard model for mixed and augmented reality, in which the living physical objects are called live actors and entities. Our experiments illustrate the realistic movements and interactions of live actors and entities that are embedded into 360° virtual reality scenes.

Keywords Live actor and entity · Mixed and augmented reality · Head-mounted display · Virtual reality · Cylindrical embedding

1 Introduction

With the advent of new sensory devices which are more advanced than headsets, they replace the real world with a totally different one. The camera technologies have

✉ Kwan-Hee Yoo
khyoo@chungbuk.ac.kr

Vuthea Chheang
chheang.vuthea@gmail.com

Sangkwon Jeong
ceo@joyfun.kr

Gookhwan Lee
ghlee@joyfun.kr

Jong-Sung Ha
jsha@woosuk.ac.kr

¹ Department of Computer Science, Chungbuk National University, Cheongju, South Korea

² JoyFun Inc., Seoul, South Korea

³ Department of Computer Engineering, Woosuk University, Wanju-gun, South Korea

evolved to the point for making possible to capture a 360° photograph, record a 360° video [1], and record a 3D visual experience using virtual reality (VR) video cameras. That is, a VR camera is required to be affordable, lightweight, robust, and easy to use. Tan et al. [2] proposed a VR camera that challenges these issues and proposed the techniques for video compression and delivery. Besides that, the streaming technologies have also evolved to an acceptable point that enables 3D high-efficiency video coding (3D-HEVC) to stream real time on wireless. Kokkonis et al. [3] proposed a high-level system architecture for a network-adaptive transmission protocol for a reliable, real-time, and multisensory surveillance system. Furthermore, the image restoration [4] and quality evaluation of virtual reality [5] are researched by many researchers.

The virtual world can be integrated into the real world with the techniques of *augmented reality* [6–10] that have grown up in order to enhance reality perception, and the opposite techniques of *augmented virtuality* [11] merge real-world objects into virtual worlds. Mixed and augmented reality (MAR), as illustrated in Fig. 1, is a comprehensive concept that deals with mixing content between the real world and the virtual world containing augmented reality [12, 13] and augmented virtuality excluding the pure real and the pure virtual world. Recently, an international standard MAR model [14] has been proposed in order to represent a *continuum* encompassing all domains or systems that utilize a combination of reality and virtuality representations.

If a live actor and entity (LAE) moving in the real world can be embedded more naturally into the MAR world, we can provide many advanced applications such as 3D telepresence, 3D virtual education, and entertainment. The MAR model suggests all guidelines of hardware and software systems for embedding LAEs into 3D virtual spaces and interaction with each other. The system's functionalities are categorized into several main components to track the motions of LAEs in the real-world, map tracked spaces and events into the virtual world. They depict integrated scenes of LAEs and the virtual world which are briefly reviewed in Sect. 2.1.

Based on the MAR workbench [15] under development by the authors, this paper presents the design and implementation of natural embedding and interaction of LAEs in a 360° virtual reality (VR) scene that is one of the most useful types of MARs. The 360° VR scene is a virtual or substituted environment surrounding a user and allows the user to look around from any angle. Users can experience 360° VR scenes in many environments, as the user can easily capture the 360° photographs and videos with a 360° camera or smartphone. Specifically, in this paper, we focus on the *natural*

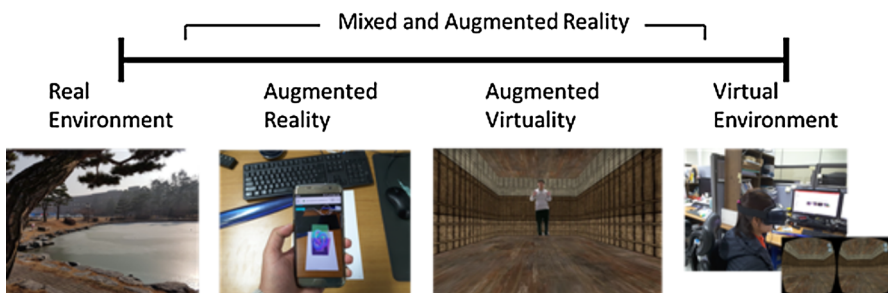


Fig. 1 Mixed and augmented reality (MAR) [14]

embedding of LAEs into the 360° VR scenes, which maps the results of LAE tracking from the real world into the virtual world.

Most systems for embedding of LAEs into the MAR world, which are surveyed in Sect. 2.2, have limitations on the natural integration of the real world and the virtual world. It is because they have to control navigations [16] with devices and icons in order to move from one scene to another scene, represent LAEs with virtual characters of 3D models, or use unnatural mapping methods for directly embedding LAEs into the VR scene. In this paper, we propose another technique called *cylindrical embedding method* that can more naturally embed real LAEs into 360° VR scenes by defining positions of the LAEs in an intermediate cylindrical surface. Our cylindrical method can be applied to other cases of cubical and spherical environments, as well as the cylindrical environment of the virtual world, which are, respectively, described in the sections of Chapter 3.

The implementation of our LAE representation system can be applied into LAEs *with chromakeying or without chromakeying* in the real world according to their specific characteristics. Chapter 4 presents the experimental results in a case with and without chromakeying, which displays natural movement and event handling of the LAEs in the 360° VR scene. Chapter 4 also describes the performance and discussion of our results. The summary and future improvements of our works are included in Chapter 5.

2 Related work

2.1 A conceptual model for LAEs in MAR

Conceptually, the system for implementing a MAR world with LAEs includes five components necessary for processing the representations and interactions of the LAEs to be integrated into the 3D virtual world. The system architecture of the proposed standard model [14] for LAEs in the MAR world is illustrated in Fig. 2. The functionalities of each component can be outlined with the types of input and output as shown in Table 1. As mentioned before in Chapter 1, this paper mainly describes the design and implementation of one *LAE spatial mapper* that naturally embeds LAEs into the 360° VR scenes. Refer to Ref. [14] for details.

2.2 Previous research and our enhancements

Enabling a user to walk inside the virtual environments is realized by the real walking of the user in the real world [17]. Cao et al. [18] proposed a web-based application which allows a user to navigate and interact with objects in a virtual tourist area by using mouse control, virtual reality scene in the system is not rendered to be adapted with head-mounted display (HMD) devices, and there are no issues on embedding of LAE into the virtual environment. Razzaque et al. [19] proposed a redirect walking which is a technique to guide a user to walk on a different path in physical room rather than what he/she is walking in the virtual environment. Chheang et al. [20] proposed a navigation control for virtual reality, which allows the user to navigate

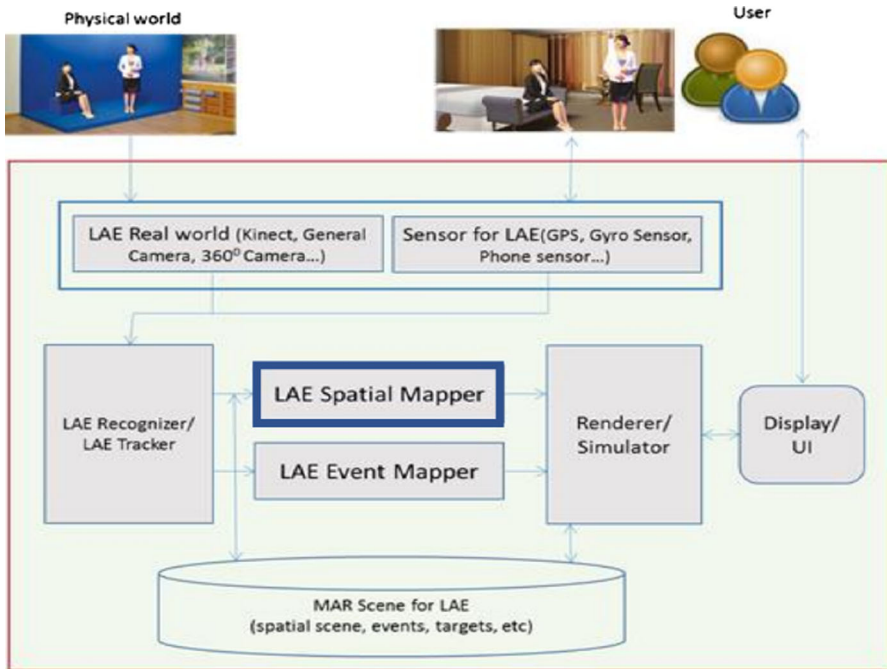


Fig. 2 System architecture for representing LAEs in the MAR world [14]

and move from one scene to another scene in virtual reality. Even though the system can be rendered as stereo vision is well adapted to an HMD device, the user needs to use the controlling device or focusing on navigation icon [21] to navigate from one scene to another scene. Sun et al. [22] proposed a system to match a given pair of the virtual and the real world for immersive VR navigation. This system enables a user to walk freely in the real world while experiencing virtual reality by using an HMD device. Floor plan of the virtual world is much larger than the real world, and globally subjective maps are designed for the proper folding of the large virtual scene into the smaller real scene. Besides that, Hashemian et al. [23] proposed a leaning-based 360° locomotion interfaces, which provide a user with full rotational motion cues and leaning-based translational motion cues for forward/backward and sideways translation. Facebook spaces [24] create a virtual avatar which represents a user in the virtual environment. Also, it allows user to interact with virtual objects by using Oculus controlling devices. The virtual avatar can be embedded into 360° VR environment by searching and downloading Facebook 360° contents, including 360° videos and photographs.

In addition, there are many studies to represent LAEs of the real worlds into the virtual world [25]. In most systems, LAEs are indirectly embedded into the virtual world by being represented with virtual characters of 3D models. Jeong et al. [26] proposed a direct embedding method for a LAE captured by a camera in the real world into 3D virtual scene by mapping with a bounding box. Chheang et al. [27] also presented similar embedding of a LAE by utilizing X3DOM [28] on a web-based

Table 1 Types of input and output for each component in Fig. 2 [14]

Dimension	Types	
	Input	Output
LAE capturer/sensor	Physical world signal	Sensor data related to representation of a LAE
LAE recognizer	Raw or processed signals representing the LAE (provided by sensors) and target object specification data (reference target to be recognized)	A: least one event acknowledging the recognition
LAE tracker	Sensing data related to the representation of a LAE	Instantaneous values of the characteristics (pose, orientation, volume, etc.) of the recognized target signals
LAE Spatial mapper	Sensor identifier and sensed spatial information	Calibrated spatial information for a LAE in a given MAR scene
LAE Event mapper	MAR event identifier and event information	Translated event identifier for a LAE in a given MAR scene
Renderer	MAR scene data	Synchronized rendering output (e.g., visual frame, stereo sound signals, motor commands, etc.)
Display/UI	Render scene data/user actions	Display output/response of user actions

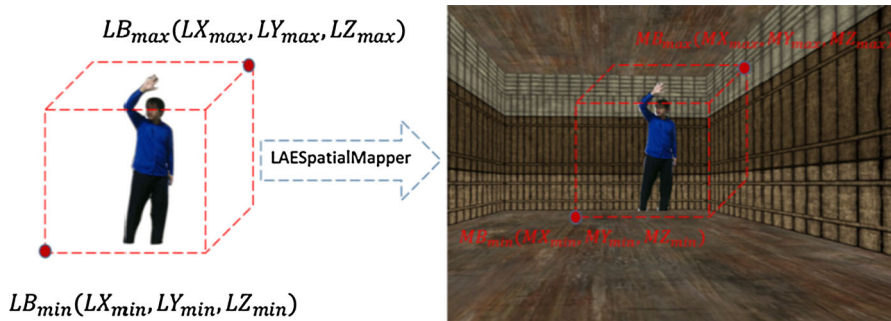


Fig. 3 Bounding box method for embedding a LAE into a virtual scene [26]

system [29] as illustrated in Fig. 3. This embedding method can apply for a LAE 2D video frame in the 3D VR scene, but the mapping bounding box can be jumped up and down during the dynamic movement of the LAE.

The virtual world of MARs treated in this paper is a 360° VR scene which is composed of a set of 360° photographs. In order to construct the virtual world, the set of 360° photographs wrap around the boundary of a solid shape such as a cube, a sphere, or a cylinder, where a viewer located at the center to see the scene from any angle including above, below, behind, and around. We developed a new technique called a *cylindrical embedding method that can be embedded the real LAEs into 360° VR scenes naturally* by defining the positions of the LAEs in an intermediate cylindrical surface. These can be applied into all 360° environments whether solid shaped cubic, sphere, and cylinder. In addition, the proposed system is designed to allow a LAE to *navigate by capturing the LAE’s position in the real world and calibrating its position*

to the virtual world [30], which can provide natural movements of the LAE while wearing a HMD device.

3 Cylindrical embedding method

In our methodology, a LAE can be mapped with two kinds of embedding. The first one refers to static embedding of a LAE into 360° VR scene. The second one is dynamic embedding with the LAE movement. The static embedding can be done by applying the cylindrical embedding method [31, 32], while the dynamic embedding can be done by utilizing a depth camera to get the depth value. Dynamic positions of a LAE in the real world are used to set the positions of a virtual camera in the virtual world. The virtual camera system in this context refers to a coordinate system where users see on the screen and is related to a viewer which acts as a camera to display a view of a 360° VR scene. Virtual camera's orientation and position will be changed to what the viewer sees. We assume that a virtual camera is like the LAE's eyes to see the environment around him/her. The mapped bounding box can be shown by multiplying the given weight to show and move in front of the virtual camera. When a user wears an HMD device, to experience with a 360° VR scene, the mapped bounding box does not need to be shown. We tracked the positions of a LAE in the real world to replace the virtual camera's positions in the virtual world. These positions also can be changed and met the boundaries limited by a given cylinder.

Let LB and MB denote the bounding box of LAE that captured by the general or depth cameras in real world and the mapped bounding box of LAE in the virtual scene, respectively. The LB can be defined with two corner points as LB_{\min} (LX_{\min} , LY_{\min} , LZ_{\min}) and LB_{\max} (LX_{\max} , LY_{\max} , LZ_{\max}). Width and height (I_w , I_h) represent the size of an image captured by general cameras or the depth cameras. Since the captured image determines bounding box of LAE, $LX_{\min} = 0$, $LY_{\min} = 0$, $LX_{\max} = I_w$, and $LY_{\max} = I_h$. The depth values LZ_{\min} and LZ_{\max} can be determined according to the camera types of the general cameras and depth cameras. A 360° VR scene can be constructed by a cylindrical, spherical, or cubical panorama [33]. In various systems, a LAE can be embedded and moved freely in an open-space such as 3D virtual scene or MAR scene where the boundary is unlimited. Besides those scenes, the 360° VR scene which is constructed by a spherical, cylindrical, or cubical environment, the embedding and movement of a LAE may be out of the boundary. Therefore, we propose the cylindrical embedding method, to determine the position of a LAE which is embedded into a 360° VR scene as shown in Fig. 4. Moreover, the cylindrical embedding method is defined based on the cylindrical projection [34, 35].

Figure 5 displays the method of cylindrical embedding for a LAE into a 360° VR scene that is constructed by cylindrical, spherical, or cubical environments. We assume that the movement space of the LAE is represented as a bounding box and the obtained bounding box will be mapped into the 360° VR scene. The cylindrical embedding can define the position of a mapped bounding box to be wrapped and limited around a given cylinder. Figure 5a describes a LAE captured in the real world, which determines the bounding box of LAE (LB). Figure 5b describes the mapped bounding box (MB) of the LAE to be mapped in a 360° VR scene by applying a cylindrical embedding



Fig. 4 360° VR scenes constructed by cylindrical, spherical, and cubical environments

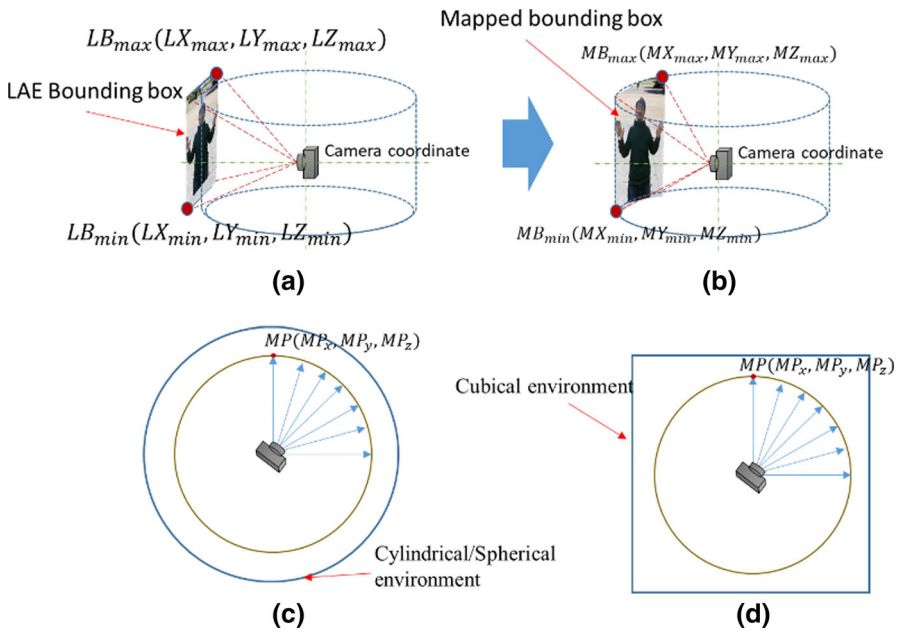


Fig. 5 Cylindrical embedding for a LAE to be embedded into a 360° VR scene

method. Figure 5c, d shows the horizontal projection views of cylindrical embedding inside the cylindrical and spherical environment, and the embedding inside the cubical environment of a 360° VR scene, respectively. The embedding of LAE bounding box is defined with two points $LB_{min}(LX_{min}, LY_{min}, LZ_{min})$ and $LB_{max}(LX_{max}, LY_{max}, LZ_{max})$. Similarly, the mapped bounding box is defined with two points $MB_{min}(MX_{min}, MY_{min}, MZ_{min})$ and $MB_{max}(MX_{max}, MY_{max}, MZ_{max})$. The mapped position in the cylindrical area is defined as $MP(MPx, MPy, MPz)$.

3.1 Cylindrical environment

Since the cylindrical environment offers only a limited rotational freedom, the user can turn around and see views in a full circle. This is the rotation around the vertical axis, while looking up and down is restricted.

As shown in Fig. 6, the cylindrical environment is constructed by creating a cylinder and texture with the cylindrical panorama image. The position of mapped bounding box MP_x can be clarified by static and dynamic mapping. In the case of static mapping, cylindrical embedding method is proposed. In the case of dynamic mapping of LAE in cylindrical environment, the mapped bounding box position is defined by the tracked head position of LAE in real world. This is multiplied to a given weight to set the bounding box in front of the virtual camera.

Figure 7a, b shows the horizontal projections of embedding the LAE bounding box with the cylindrical environment. Since the bounding box can be wrapped in a cylinder with an appropriate radius, we can define the mapping position MP_x with the following equation.

$$MP_x = R\alpha = R \cdot \arctan(LX_{max}/R) \quad (1)$$

In Eq. (1), $MP_x \leq LX_{max}$ and the radius R of the cylinder $R = I_w/w_b * \pi$. Let I_w and w_b be the image plane width and the given weight of the bounding box, respectively. A point C refers to the camera coordinate which is the center of the cylinder. α is defined as the angle in radians with $\alpha = \arctan(LX_{max}/R)$. Figure 7c, d displays the vertical projections of cylindrical embedding, where MP_x is the horizontal ray that hits the cylinder and LX_{max} is the position of ray that hits the flat screen, respectively. We define l as the length of the vector between C and LX_{max} , which can be calculated

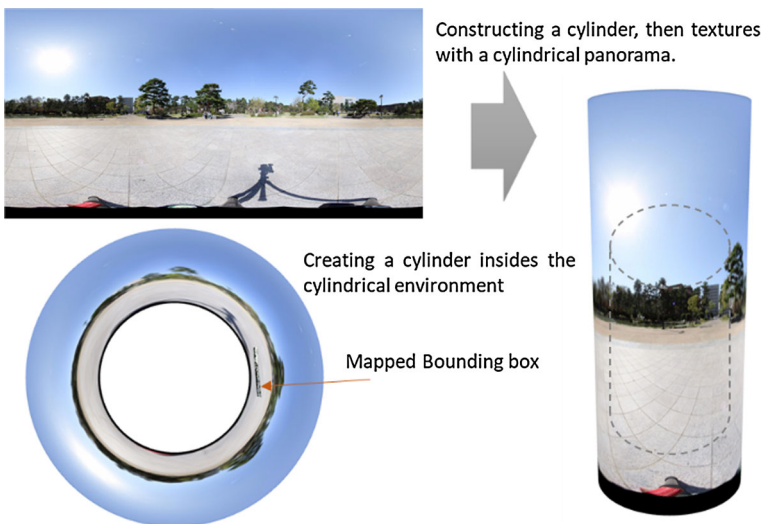


Fig. 6 Cylindrical embedding of a LAE into cylindrical environment

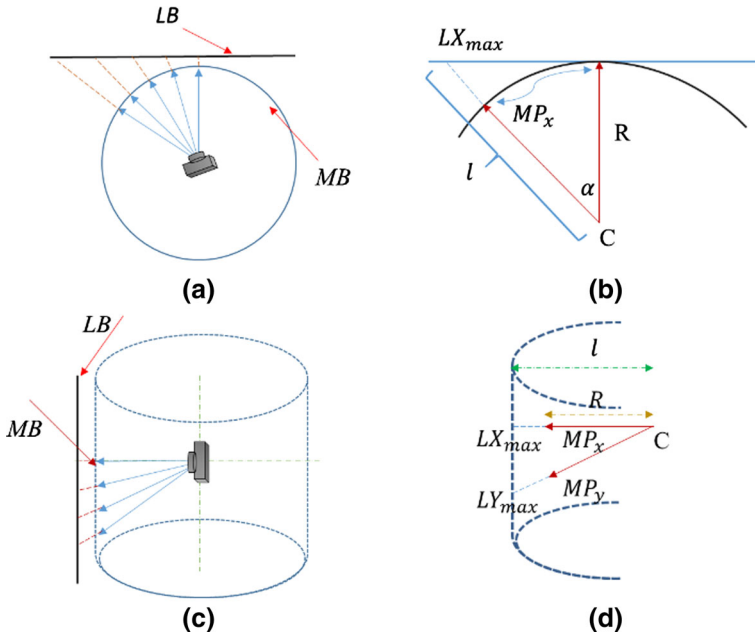


Fig. 7 Horizontal and vertical projections for cylindrical embedding of a LAE

as $l = \sqrt{LX_{max}^2 + R^2}$. The distance from center point C to MP_x is the radius R , and the distance from C to LX_{max} is l . Since two triangles $\triangle CLX_{max}LY_{max}$ and $\triangle CMP_xMP_y$ are isomorphic, they hold true that $MP_y = LY_{max} \cdot R / l$ is derived from $LY_{max} / l = MP_y / R$. As a result, MP_y can be calculated as follows:

$$MP_y = LY_{max} \cdot R / \sqrt{LX_{max}^2 + R^2} \tag{2}$$

In Eq. (2), $MP_y \leq LY_{max}$; $MP_y = LY_{max}$ if $LX_{max} = 0$. The depth value of MP_{z_t} at a specific time t can be recalculated as the same as the depth value of LZ_t according to camera types. After defining the position of the LAE in the given cylinder, a LAE will be embedded and moved naturally within the 360° VR scene. We can set the size of the cylinder by utilizing the weight of bounding box w_b . If the LAE movements meet the specific positions in the cylinder, an event will occur such as moving to the next scene of the 360° image sequences.

3.2 Cubical environment

The cubical environment of a 360° VR scene is constructed by six images for every face of a cube. As shown in Fig. 8, the cubic panorama is used to construct the cubical environment and place the user at its center. We can extend the cylindrical embedding method to apply the mapping LAE bounding box for a cubical environment. Moreover, the mapping position MP of the LAE bounding box can be defined according to the size of a given cylinder. Since the dynamic positions of a LAE in the real world are

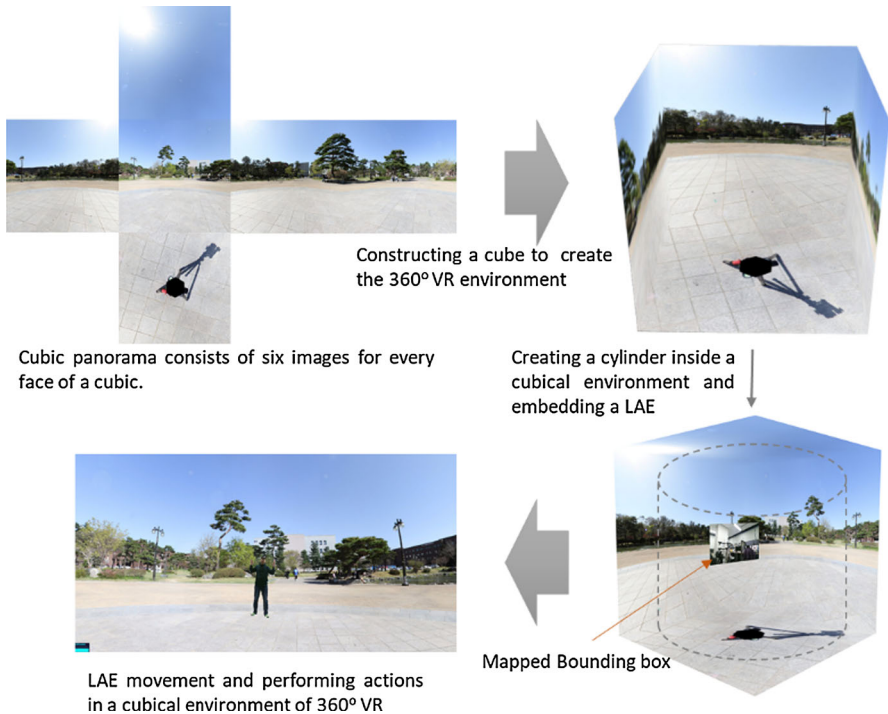


Fig. 8 Cylindrical embedding of a LAE into cubical environment

tracked by the depth sensor, the dynamic mapping position can be defined according to the tracked positions multiplied with a given weight. The tracked positions are also used to set the virtual camera's positions for feeling like natural movement of LAE in a 360° VR scene.

3.3 Spherical environment

Figure 9 shows the process of cylindrical embedding LAE into a spherical environment. As the 360° VR scene can be constructed by the spherical environment, this kind of scene can provide the experience of more immersive virtual reality with every angle of views including top and bottom. The spherical environment is constructed by creating a sphere and placing the user at its center. Moreover, an equirectangular image is used for adding texture to the sphere. Not much different from cubic and cylindrical environments, the cylindrical embedding method can also apply to the spherical environment. This is done by defining the position of mapped bounding box along with a given cylinder.

The cylindrical embedding is widely used for a 360° VR environment. Different from other open-area environments, a 360° VR environment is limited by the boundaries of a sphere, cubic, or cylinder. To solve that, the cylindrical embedding is proposed. Compared to other embedding methods [26, 27], the cylindrical embedding

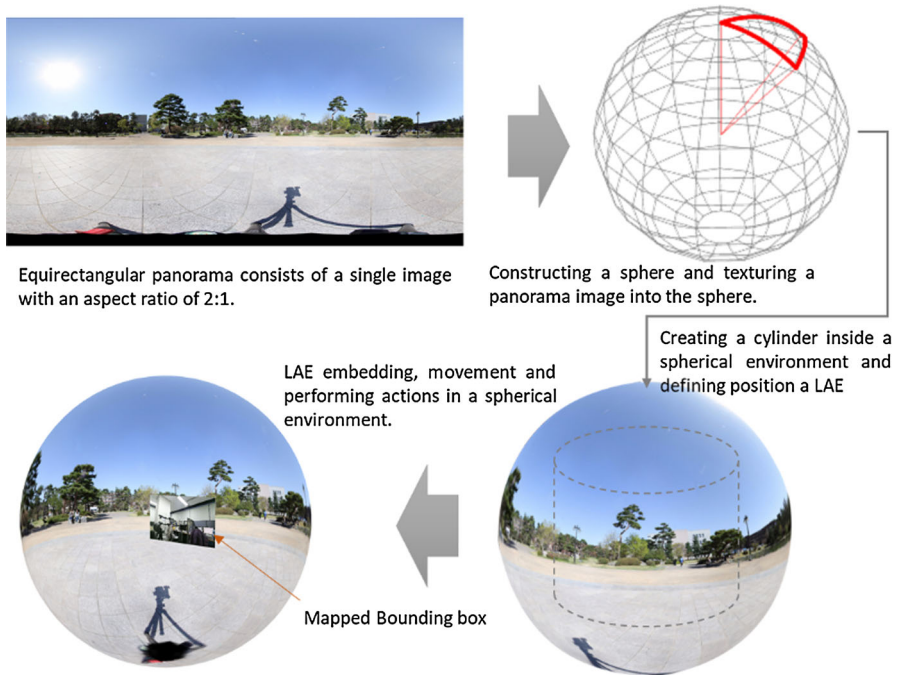


Fig. 9 Cylindrical embedding of a LAE into spherical environment

limits a user in a given cylindrical area to avoid walking out the boundaries. Moreover, it is robust, usable, and effective methods to apply with any kind of 360° VR environments, as we defined a cylindrical area inside those environments. Besides the cylindrical environment, this method can also apply to spherical and cubical environments for the reason that a LAE representation and movements are on X- and Z-axis (left/right and forward/backward), and the positions of mapped bounding box are easily defined by a cylinder as shown in Fig. 5a, b.

3.4 Reachable area

Figure 10 shows the reachable area of LAE movement in front of a capturing device. Kinect which is one of the capturing devices is used for obtaining the depth information of a LAE. The limitation of the LAE's movement is based on field of view (FOV) of the Kinect camera that is used in our system. The minimum depth distance of this camera is around 0.4 m, and the maximum depth distance is around 4.5 m from the center of camera. Moreover, the limitation of LAE's movements in a 360° VR scene is also determined by a cylindrical embedding method which is described in Sect. 3. The virtual camera is presented in the reason that allowing a LAE to see the 360° VR environment. Moving positions of a LAE in the real world also change the positions of virtual camera in the virtual world.

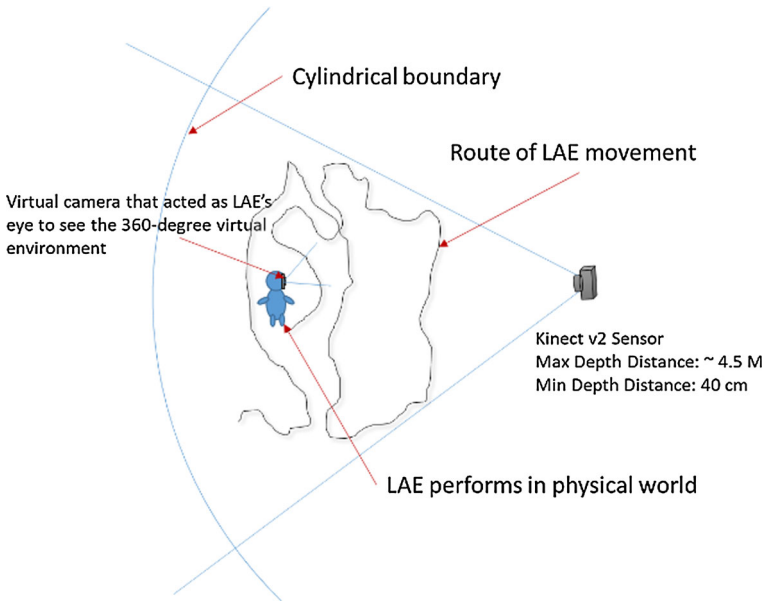


Fig. 10 Reachable area of LAE's movement

4 Experimental results

Our experiment is configured with a Microsoft Kinect V2 sensor and a computer with CPU 3.40 GHz core i7, RAM 16 GB, and Nvidia GeForce GTX 1060 graphic card. JavaScript, NodeJS, Kinect2, Electron, and ThreeJS are used for developing the proposed system.

Figure 11 shows the environment setup where we use the blue board to stick on the wall behind the LAE for the quality of tracking and chromakeying. We prepared an experimental room by 5 m square and assumed there are no physical obstacles inside it. Webcam camera is used to record the demonstration video of the LAE in the real world while experiencing 360° VR scene as shown in Fig. 17b, d. Moreover, the Kinect V2 is used to capture the LAE's positions and calibrate the captured LAE for representing in 360° VR scenes. At the initial step, we tried to set the position of the Kinect camera to a center position of the 360° VR environment (sphere/cube/cylinder) with texture mapping of the captured 360° photograph as shown in Fig. 10. The limitations of LAE's movements are based on the field of view of the Kinect camera and the proposed method which is described in Sect. 3.

Since the Kinect camera is a depth sensor that can be used for tracking positions of a LAE in the real world, in this paper, it is used to track the positions of the LAE's head. The Kinect camera is stood in fixed position; however, it is capable of tracking the movement of LAE's head positions in the real-world coordinates. In addition, the tracked positions are utilized to set the virtual camera's positions for movement. Therefore, we need to prepare the calibration of tracking the LAE's positions in the real world to the virtual world, in the reason of avoiding the jumping [36] of positions and

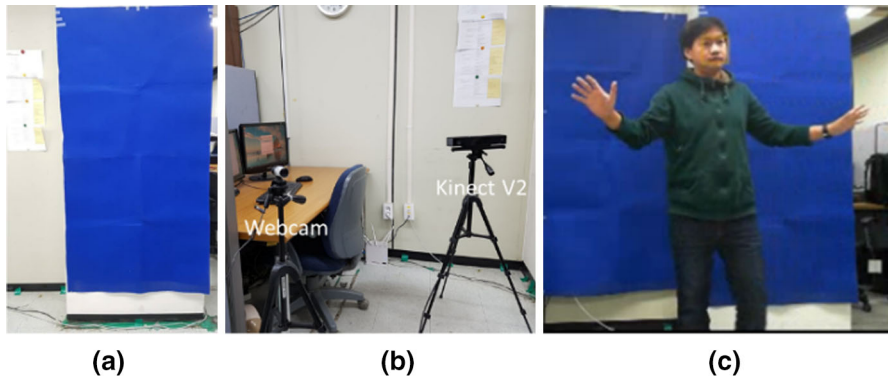


Fig. 11 Environment setup for LAE representation

orientations of a LAE when moving. The calibration of the camera has two parameters (intrinsic and extrinsic parameters); nevertheless, our experimental issues are related to the extrinsic parameters of the camera since the intrinsic parameters are described by the camera sensor itself and its lens. The extrinsic parameters represent the positions and orientations of the camera in the world coordinate. When the position of a LAE is moved from one to another position, the jumping of positions can occur in the virtual world. Therefore, the errors in calibration may result in precision errors. Raposo [37] proposed an algorithm for a fast and accurate calibration of a Kinect sensor. Hence, this algorithm can be used to calculate and determine the consistency of the calibration for LAE.

Moreover, the depth measurements can be measured by the Kinect sensor that consists of an infrared (IR) camera. Since Kinect V2 is a RGB-depth sensor, the distance of LAE within the range of camera recording is calculated from time-of-flight (ToF) analysis with the reflected light beams. The IR camera detects a constant pattern that is emitted by a projector. Thus, it can provide the depth information in disparity units that can be converted to meters. In addition, the software development kit (SDK) of the Kinect V2 is used to detect the LAE shapes based on machine learning techniques. It provides the artificial anatomical landmarks ('Kinect Joints') that we can define the head's positions of the LAE. The studies on movement accuracy of Kinect V2 [38] showed the differences in signal accuracy with the landmark location and the direction of movements. Moreover, the depth accuracy of Kinect camera is relatively constant within a specific captured volume depending on the vertical and horizontal displacement from the center of the camera. To provide the accuracy of movements of a LAE and virtual camera, results of Yang et al. [39] are applied to our system so that the average of depth accuracy is under 2 mm in the central viewing cone and it will be increased to 2–4 mm in the range up to 3.5 m.

Representation of LAE in a 360° VR scene without chromakeying is shown in Fig. 12. The LAE bounding box is mapped into the 360° VR scene by applying the cylindrical embedding method. The positions of LAE movement in the real world are tracked by a Kinect sensor and used for setting the virtual camera's position. That is, it gives a feeling to move naturally in a virtual environment. The mapped bounding box

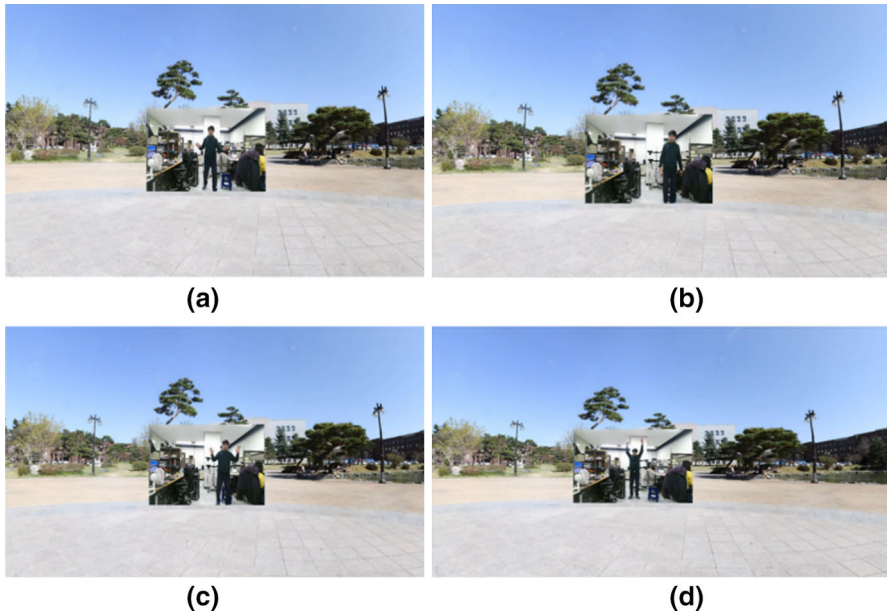


Fig. 12 LAE representation without chromakeying

is set to be mapped in front of the real tracked position by being multiplied with the given weight.

Figure 13 shows the LAE movement in a 360° VR scene with the movement of the virtual camera. The LAE movements in the real world are forward, backward, left, and right, which are shown in Fig. 13a–d, respectively. The positions of a LAE in the real world are captured by a Kinect device and the tracked positions of LAE's head are utilized to set the position of the virtual camera for natural movement. The tracked position can also be used for setting the position of mapping the bounding box by being multiplied with a given weight in order to display the mapped bounding box in front of the virtual camera. Figure 14 illustrates the tracked points of LAE skeleton which is drawn into LAE bounding box and mapped as mapped bounding box into the 360° VR scene. Moreover, the LAE skeleton can walk and interact with objects using a hand gesture. That is, we draw two circles on the hand position of the LAE. The green color means the hand is being released, while the red color means the hand is being gripped. The LAE skeleton representation also can be represented as a 3D skeleton as shown in Fig. 15. That is, the skeleton data are used to draw the 3D bones of the actor. We create the bone hierarchy of the LAE including root, spine mid, neck, head, shoulder, elbow, wrist, hand, hip, and knee. The bone hierarchy is used to represent the captured skeleton data of LAE in the real world.

A LAE in the real world can make some kinds of gestures according to the defined gestures types. The goals of making gestures are referred to provide interaction, controlling, and moving of virtual objects in the virtual world. If a person is represented as a LAE, he/she can also make the gestures for interacting with navigation objects

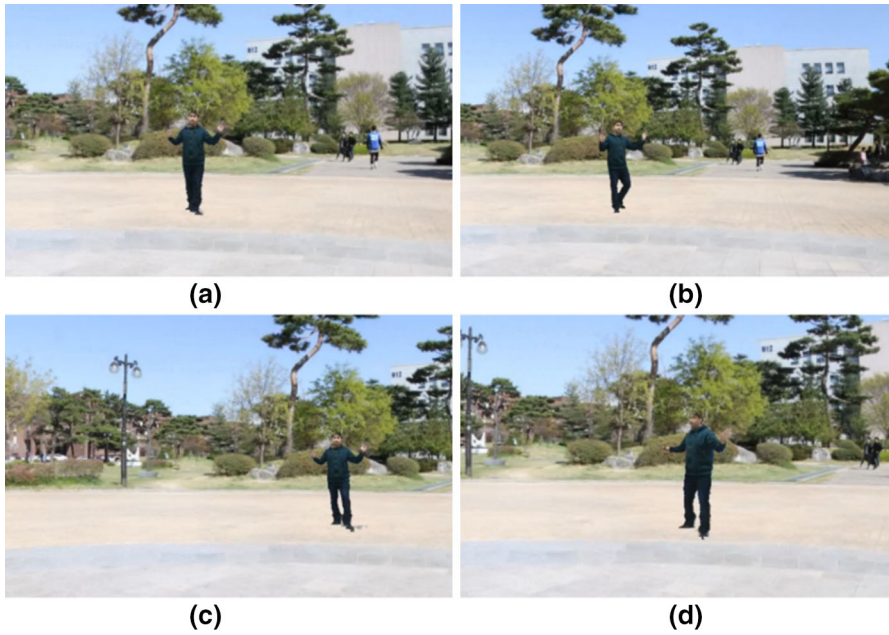


Fig. 13 LAE movement with chromakeying in a 360° VR scene

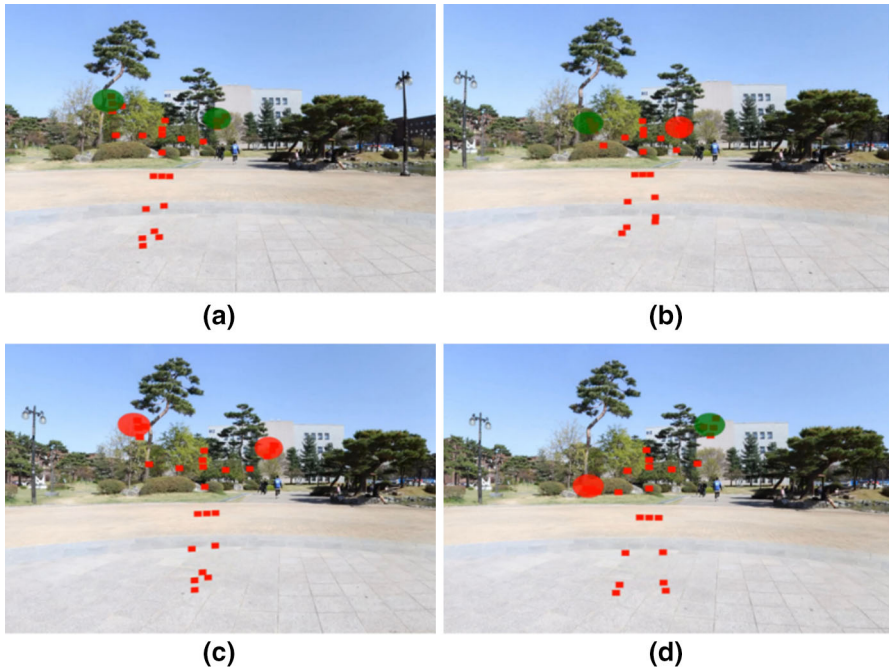


Fig. 14 LAE skeleton representation and movement in a 360° VR scene

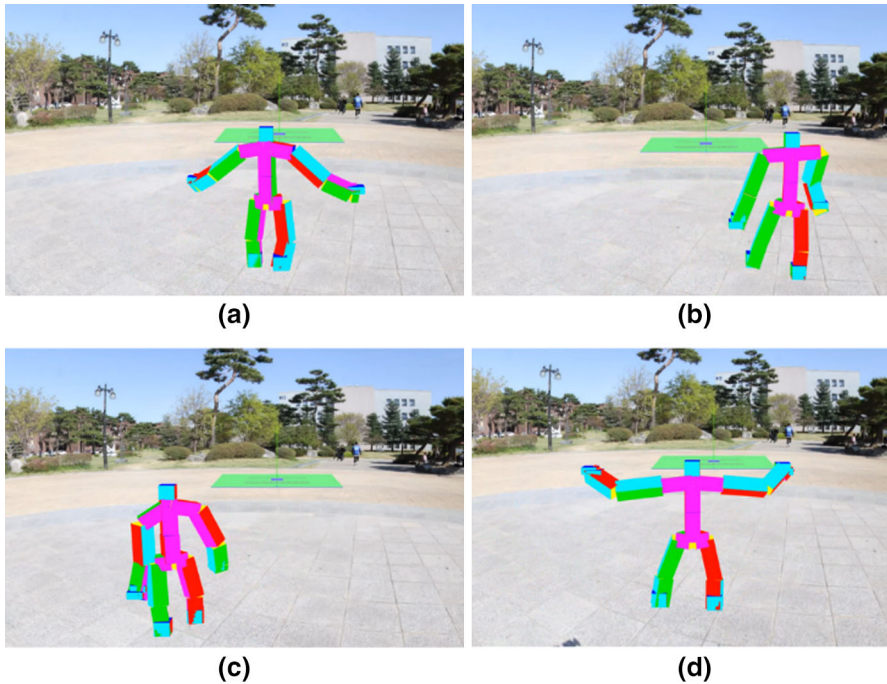


Fig. 15 LAE representation and movement as 3D skeleton

in order to move to another scene. Figure 16 shows the event gesture of the LAE portrayed in the real world, and these gestures can be tracked and recognized to create an event in the virtual world. In Fig. 16a, a LAE is making a gesture with his/her left hand that is being released, and his/her right hand which is being gripped. When the hand is released, virtual fire will be created and set to its position on the LAE hand. In Fig. 16b, the two hands of a LAE are released, and then, virtual fire is created in both two hands. Moreover, the right hand is released, and left hand is gripped as shown in Fig. 16c, and d shows that both hands are gripped.

A person who becomes a LAE can experience 360° VR scenes by wearing a HMD device, while positions of his/her movement are tracking by a depth sensor. Figure 17a, c illustrates the stereoscopic rendering on HMD device for a LAE to view the 360° VR scenes. Figure 17b, d shows the rendering of viewing the 360° VR scenes with the tracked video including a LAE in real world. We can use another webcam camera to capture the LAE in the real world as shown in bottom area of Fig. 17b, d. Furthermore, Fig. 18 shows the other viewpoints of a LAE wearing a HMD device and embedding into the 360° VR scenes. Generally, when wearing a HMD device, the LAE cannot see the outside environment. However, the LAE can see other LAEs that are embedded to the 360° VR scenes as shown in Fig. 18d.

The performance of LAE representation in a 360° VR scene is shown in Table 2, where the response time for the scene and the types of LAE representation are calculated in milliseconds. The cylindrical environment is loaded more sluggishly than

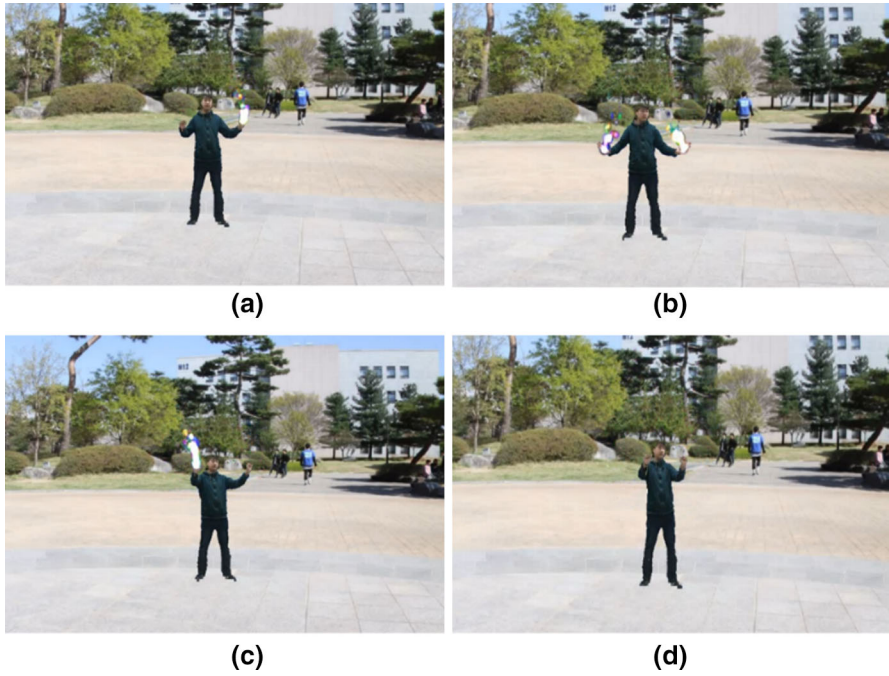


Fig. 16 Event gesture of a LAE in a 360° VR scene

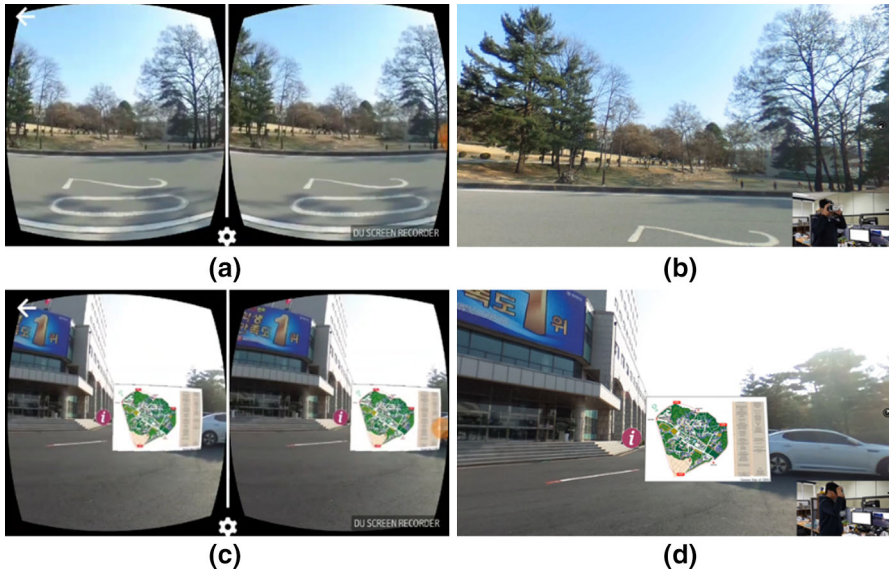


Fig. 17 Stereo rendering on HMD device and the viewing of 360° VR scenes

other environments. The type of LAE representation such as LAE image without or with chromakeying, LAE skeleton, and LAE depth can be measured the performance

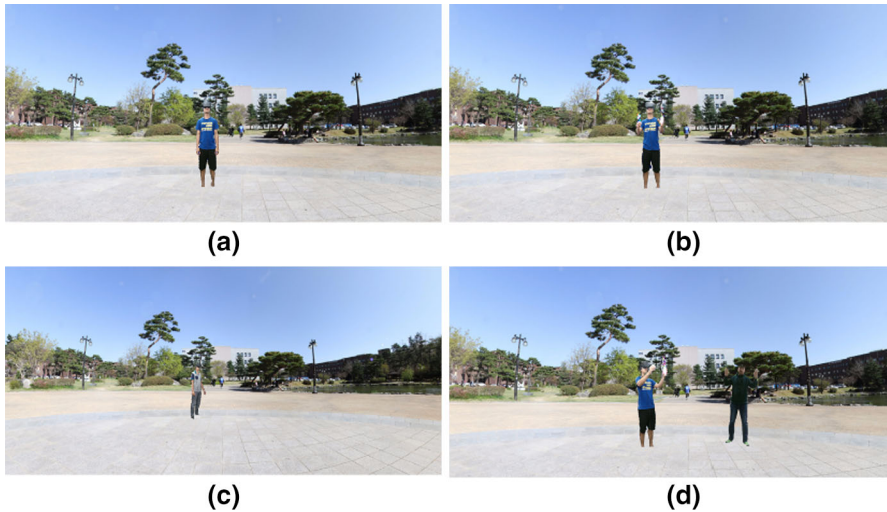


Fig. 18 The representation of a LAE while experiencing in 360° VR scenes

Table 2 Performance of LAE representation in a 360° VR scene

Type/environment	Spherical environment	Cubical environment	Cylindrical environment
LAE image without chromakeying (ms)	0.10	0.114	0.149
LAE image with chromakeying (ms)	0.164	0.165	0.195
LAE skeleton (ms)	0.130	0.139	0.164
LAE skeleton 3D (ms)	0.115	0.13	0.115
LAE depth (ms)	0.104	0.115	0.09

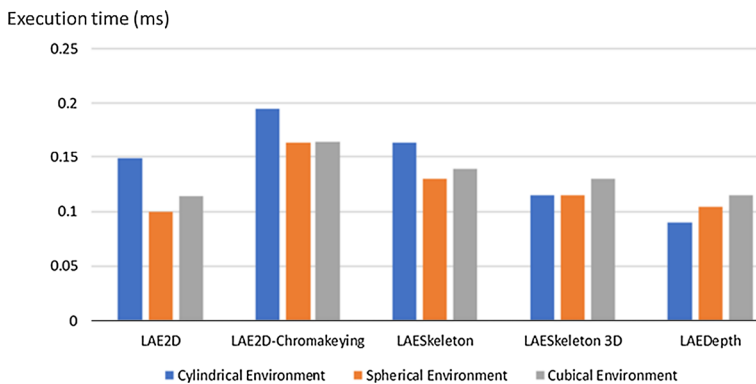
according to the process of capturing and tracking the LAE in the real world. The lower number means that the response time is faster for representing LAE in 360° VR environments. Table 3 shows the performance of a 360° VR environment which is constructed by the cylindrical, cubical, or spherical environments. Frame per second (FPS) is used to measure frames rendered in the last second, where the higher number is the better. Millisecond (MS) is used to measure the response time needed to render a frame, where the lower number is better. Megabyte (MB) is the measurement for the allocated memory in the process of our system.

Figure 19 shows the performance comparison of 360° VR environments for LAE representation such as 2D, 2D chromakeying, skeleton, 3D skeleton, and depth frame. Our experiments have shown that the spherical environment for a 360° VR scene is more reliable than cylindrical and cubical environment. In case of LAE depth, the cylindrical environment is better than the others.

We report experimental results of embedding and interactions of a LAE into a 360° VR scene in Tables 4 and 5, respectively. Table 4 illustrates the embedding of

Table 3 Rendering performance of a 360° VR scene

Type/environment	Spherical environment	Cubical environment	Cylindrical environment
Frame per second (FPS)			
Min	15	23	6
Max	61	61	61
Millisecond (MS)			
Min	1	2	2
Max	351	292	486
Mbytes (MB)	10	10	10

**Fig. 19** A comparison of rendering performances of a 360° VR scene

movement of a captured LAE that is performed from the real world into a 360° VR scene in the real time. The tracked positions of the LAE's head are used to control the virtual camera, if the LAE's head is used to simulate the position. The movement types, such as moving forward, backward, left, and right, can be used to change the virtual camera's positions. Moreover, the event gestures of a LAE in 360° virtual reality scene are shown in Table 5. The hand gestures can be tracked according to the camera types. We tracked the hand releasing and hand gripping for controlling the event mapper. The event control is distinguished according to the left- and right-hand gestures. For an example, while the hand is releasing, the event mapper will create the virtual fire animation along the tracked position of LAE's hand.

We tried to apply the related embedding methods for embedding a LAE into a 360° VR scene. A comparison of embedding results is shown in Fig. 20. Figure 20a shows the embedding result after applying an embedding method which is proposed by Jeong et al. [26] into a 360° VR scene. This method works well in a 3D virtual environment, but a LAE jumped up and down in a 360° environment. Figure 20b illustrates the result of applying an embedding method which is proposed by Chheang et al. [27]. The result looks better than Fig. 20a, but it is still not naturally embedding. Figure 20c

Table 4 Embedding of LAE’s natural movements from real world to a 360° VR scene



















Initial state	LAE in real world	LAE in 360° VR	Movement types	Description
			Moving Forward ↓	Movement of the virtual camera’s position with the increased z-value for moving forward
			Moving Backward ↑	Movement of the virtual camera’s position with the decreased z-value for moving backward
			Moving Left ←	Movement of the virtual camera in positive direction of X axis according to the tracked position of LAE’s head.
			Moving Right →	Movement of the virtual camera in negative direction of X axis according to the tracked position of LAE’s head.

Table 5 Gestures for giving LAE’s interactions in a 360° VR scene

Initial state	LAE in real world	LAE in 360° VR	Hand Gesture types	Description
			LEFT & RIGHT releasing	Tracking left and right hands which are releasing, then creating events along both two hands.
			LEFT releasing & RIGHT gripping	Tracking left hand which is releasing and right hand which is gripping, then create event along only left hand.
			LEFT gripping & RIGHT releasing	Tracking right hand which is releasing and left hand which is gripping, then create event along only right hand.
			LEFT & RIGHT gripping	Tracking left and right hands which are gripping, no event to be occurred.

shows the result of our proposed method, and the LAE is embedded more naturally into a 360° VR scene than others.

A study of calibration of Kinect-type RGB-D sensors [40] has shown the results of sensor accuracy depending on distance from sensor and measurement error. The results are illustrating the errors obtained with nominal parameters enlarged by 50% when the depth sensor with its associated RGB camera is used. The errors were on the order of 35 mm (standard deviation 10 mm) at the distance of 2 m and on the order of 75 mm (standard deviation 15 mm) at the distance of 3 m. And also, the accuracy of

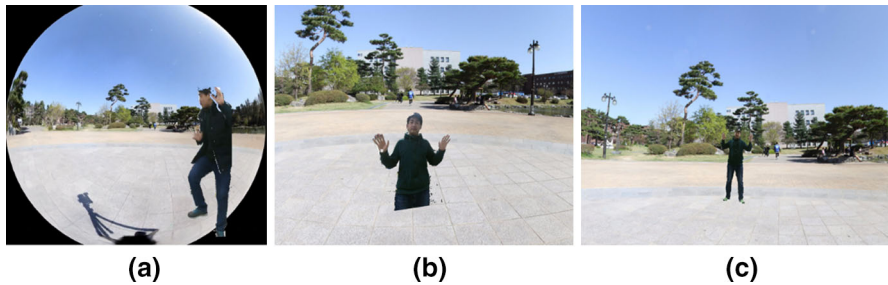


Fig. 20 Comparison of the embedding results of Jeong et al. [26], Chheang et al. [27], and the proposed method

gesture recognition of a LAE can be evaluated similar to a method proposed by Cho and Jeong [31]. The problems of LAE's gesture recognition encompasses with two areas: segmentation and recognition. The segmentation refers to find the starting and ending points of a valid gesture, and recognition involves the matching segmentation gestures with a defined event database. The best performance results were obtained at distance 2.5 m and lowest results at 3.5 m. With experiments, we showed that moving forward gestures were more accurate than moving backward one, and front-pose gesture was recognized very well.

5 Concluding remarks

We have designed and implemented natural embedding and interaction of LAEs in a 360° VR scene by utilizing the cylindrical mapping method. Our efforts to integrate the real world to the virtual world have been adopted or developed for enhancing the research as follows.

- Positions of LAE in real world can be captured by Kinect sensor and are used for controlling the mapped bounding box and virtual camera in the virtual world.
- LAEs can be more naturally embedded into 360° VR scenes with cylindrical embedding method without jumping up and down during the dynamic movements of LAEs, unlike the well-known bounding box embedding which are proposed by Jeong [26] and Chheang [27].
- Our cylindrical embedding can be adapted into all 360° virtual environments with any solid shape such as cubic, sphere, and cylinder.

Our experimental results in a case without chromakeying were described step by step. The proposed method can be used to track the movements of a LAE, to construct the skeleton of the LAE, to recognize the gesture of the LAE, and to integrate naturally the LAE into the 360° VR scene. The performances for all these steps and total construction of the 360° VR scene were also measured, and the movements and event gestures were also evaluated.

Future works on improving this system performances and reconstructing 3D models with captured information are needed. It is a challenge to define and implement all types of methods for embedding physical objects into 360° VR scenes. Duo et al. [41]

proposed a technique that was a new method for real-time high-quality 4D performance capture allowing incremental nonrigid reconstruction from multiple RGB-D cameras. Our future research is applying 3D reconstruction of a LAE rather than 2D bounding box to be embedded into 360° VR scene and the moving a LAE from one scene to another scene by matching the real walking of a LAE in real world into a 360° VR scene. Specially, we are studying on detecting and avoiding a LAE from physical obstacles in the future.

Acknowledgements This research was financially supported by the ICT R&D program of MSIP/IITP [20160001210021001, Standard Development of HMD Based VR Service Framework] and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2013-0-00881) supervised by the IITP (Institute for Information & Communications Technology Promotion).

References

1. Pope VC, Dawes R, Sheikh A (2017) The geometry of storytelling: theatrical use of space for 360-degree videos and virtual reality. In: Conference on Human Factors in Computing Systems (CHI 2017), pp 4468–4478
2. Tan J, Cheung G, Ma R (2018) 360-degree virtual-reality cameras for the masses. *IEEE Multimed* 25(1):87–94
3. Kokkonis G, Psannis KE, Roumeliotis M, Schonfeld D (2017) Real-time wireless multisensory smart surveillance with 3D-HEVC streams for internet-of-things (IoT). *J Supercomput* 73(3):1044–1062
4. Dong W, Zhang L, Shi G, Li X (2012) Nonlocally centralized sparse representation for image restoration. *IEEE Trans Image Process* 22(4):1620–1630
5. Jiang B, Yang J, Jiang N, Lv Z, Meng Q (2018) Quality assessment for virtual reality technology based on real scene. *Neural Comput Appl* 29(5):1199–1208
6. Azuma RT (1997) A survey of augmented reality. *Presence Teleop Virtual Environ* 4(2):355–385
7. Paul M, Takemura H, Utsumi A, Kishino F (1994) Augmented reality: a class of displays on the reality-virtuality continuum. In: Proceedings of Telemanipulator and Telepresence Technologies 2351:34
8. Jeon S, Choi S (2011) Real stiffness augmentation for haptic augmented reality. *Presence: Teleop Virtual Environ* 20(4):337–370
9. Lindeman R, Noma H, Barros P (2008) An empirical study of hear-through augmented reality using bone conduction to deliver spatialized audio. In: Proceedings of IEEE VR, pp 35–42
10. Lee G, Dunser A, Kim S, Billinghurst M (2012) CityViewAR: a mobile outdoor AR application for city visualization. In: Proceedings of ISMAR, pp 57–64
11. Perrine P, Fleig O, Jannin P (2005) Augmented virtuality based on stereoscopic reconstruction in multimodal image-guided neurosurgery: methods and performance evaluation. *IEEE Trans Med Imag* 24(11):1500–1511
12. Billinghurst M, Kato H, Poupyrev I (2001) The MagicBook: a transitional AR interface. *Comput Gr* 25(5):745–753
13. Fuchs H, Livingston M, Raskar R, Colucci D, Keller K, State A, Crawford J, Rademacher P, Drake S, Meyer A (1998) Augmented reality visualization for laparoscopic surgery. In: Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention, pp 934–943
14. ISO/IEC DIS 18039 (2018) Information technology—computer graphics and image processing—mixed and augmented reality reference model
15. ISO/IEC CD 18040 (2018) Information technology—computer graphics, image processing and environmental data representation—Live actor and entity representation in mixed and augmented reality
16. Hong L, Muraki S, Kaufman A, Bartz D, He T (1997) Virtual voyage: interactive navigation in the human colon. In: SIGGRAPH'97, pp 27–34
17. Kunz A, Zank M, Fjeld M, Nescher T (2016) Real walking in virtual environments for factory planning and evaluation. In: 6th CIRP Conference on Assembly Technologies and Systems (CATS), Procedia CIRP, 44:257–262

18. Cao Z, Hu Z (2012) Design virtual reality scene roam for tour animations base on VRML and Java. In: 2012 International Conference on Solid State Devices and Materials Science, Physics Procedia 25:693–699
19. Razzaque S, Kohn Z, Whitton MC (2001) Redirected walking. In: Proceedings of Eurographics, pp 289–294
20. Chheang V, Chhaya P, Bunrong L, Ryu G, Jeong S, Lee G, Yoo KH (2017) Navigation control in virtual reality walking tour. In: The 4th International Conference on Bigdata and Application, 4(1):210–212
21. Schild J, Laviola J, Masuch M (2012) Understanding user experience in stereoscopic 3d games. In: CHI'12, pp 89–98
22. Sun Q, Wei LY, Kaufman A (2016) Mapping virtual and physical reality. In: Proceedings of ACM SIGGRAPH 2016, 35(64):1–12
23. Hashemian AM, Riecke BE (2017) Leaning-based 360 interfaces: investigating virtual reality navigation interfaces with leaning-based-translation and full-rotation. HCI 10:15–32
24. Facebook Spaces. <https://www.facebook.com/spaces>. 06 June 2018
25. Kim IK, Kwon SA, Kim M, Yoo KH (2015) X3D nodes for representing and rendering real characters in 3D virtual environments. Int J Smart Home 09(2):193–206
26. Jeong JS, Kwon SA, Park JA, Park C, Baek N, Jang RH, Yoo KH (2013) An embedding method for real characters into the 3D virtual space. Int J Softw Eng Appl 7(1):69–78
27. Chheang V, Ryu G, Jeong S, Lee G, Yoo KH (2016) A web-based system for embedding a live actor and entity using X3DOM. In: 2016 Korea Broadcasting and Media Engineering Conference, pp 1–3, Nov 2016
28. X3DOM (2018) <http://www.x3dom.org>, 19 June 2018
29. Kontakis K, Malamos AG, Steiakaki M, Panagiotakis S (2017) Spatial indexing of complex virtual reality scenes in the web. Int J Image Gr 17(2):1–18
30. Cheng L-P, Roumen T, Rantzsch H, Kohler S, Schmidt P, Kovacs R, Jasper J, Kemper J, Baudisch P (2015) Turkdeck: Physical virtual reality based on people. In: UIST'15, pp 417–426
31. Cho MY, Jeong YS (2017) Human gesture recognition performance evaluation for service robots. In: International Conference on Advanced Communications Technology(ICACT), pp 847–851
32. Chheang V, Jeong S, Lee G, Yoo KH (2017) Cylindrical embedding a live actor and entity into 360° virtual reality scene. In: The 5th International Conference on Bigdata and Application, pp 43–45
33. Jolesz FA (1997) Image-guided procedures and the operating room of the future. Radiology 204(3):601–612
34. CompuPhase.com (2017) Panorama image projection. Technical report (ITB CompuPhase), August 2000, retrieved 15 October 2017
35. Wikipedia.org (2018) Cylindrical equal-area projection. https://en.wikipedia.org/wiki/Cylindrical_equal-area_projection. 15 June 2018
36. VRTracker.xyz (2018) Solving your calibration problems. <https://vrtracker.xyz/solving-calibration-problems/>. 24 July 2018
37. Raposo C, Barreto JP, Nunes U (2013) Fast and accurate calibration of a kinect sensor. In: International Conference on 3D Vision, pp 342–349
38. Wang Q, Kurillo G, Ofii F, Bajcsy R (2015) Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect. In: International Conference on IEEE Healthcare Informatic (ICHI), 2015, pp 380–389
39. Yang L, Zhang L, Dong H (2015) Evaluating and improving the depth accuracy of kinect for windows v2. IEEE Sens J 5:4275–4285
40. Karan Branko (2015) Calibration of kinect-type RGB-D sensors for robotic applications. FME Trans 43(1):47–54
41. Dou M, Khamis S, Degtyarev Y, Davidson P, Fenello SR, Kowdle A, Escolano SO, Rhemann C, Kim D, Taylor J, Kohli P, Tankovich V, Izadi S (2016) Fusion4D: real-time performance capture of challenging scenes. ACM Trans Graphics 35(4):114