

**A Study on Effective View-Dependent Projection Mapping by Projecting
Virtual Object and Real Background with Perspectively-Correct Appearance**

2021年3月

岩手大学大学院工学研究科

デザインメディア工学専攻

Khuslen Battulga

©2021 Khuslen Battulga

All Rights Reserved

ABSTRACT

“Augmented reality (AR)” is a computer-based technology to augment the real world by adding virtual objects. Augmenting reality adds extra layers of digital information projected into the reality we see every day around us. Combining the physical world with computer-generated virtual elements, objects, and effects can and will enhance the physical world’s appearance around us as we know it. It could be a fascinating experience when the virtual and real-world could coexist seamlessly, and digital information must be recognized and emerged into the real world. A viewer feels like a virtual object existed in the real world if its appearance is correctly seen from an arbitrary position. Augmented reality (AR) is one of the best representations of what is stored for us in the future of computer vision and machine learning and virtual reality (VR) entertainment. The display of an image of a virtual object with its correct appearance according to the viewer’s position is called a “view-dependent” display. In AR, a viewer usually uses his/her display device, such as a tablet, a smartphone, and special glasses, to display a view-dependent image for his/her position.

On the other hand, in “spatial augmented reality (SAR)” including projection mapping, multiple viewers at different positions see a typical image projected on the surface of a real object, such as a wall, a building, and even a human, by the naked eyes without their own display devices. Then, if the shape of the virtual object is different from that of the real object, a viewer at an arbitrary position sees an incorrect appearance of the virtual object in general. Besides, the difference between the shapes causes the actual object’s surface to have some empty areas onto which the virtual object is not projected.

Such open spaces make the viewer not feel the virtual object merging into the real world. While lots of methods to solve the “incorrect appearance” problem have been proposed, as far as we know, there is no existing method to treat the “empty area” problem. As we mentioned before it could be easier for the viewer to spot the line between what is the real-world objects are and what are virtual reality projected lights. To build a bridge between these two opposite ends, we are proposing a method of projecting view-dependent images of the real-world surroundings with the virtual reality 3D model onto the real-world object. We propose a view-dependent projection mapping method to solve the “empty area” problem. By employing our method, on the main projection object will have less gap between the virtual object that has been projected on and the physical objects that are existing in real-life. Our method eliminates undesired empty areas by projecting the real background behind the real object in a view-dependent matter. To treat a far background in a large space, even outdoors, the actual background image is captured by an RGB camera, not an RGB-D camera with a depth range limit and converted to an appropriate image for a viewer’s position by using a background plane to approximate the background shape based on homography. The background plane is adjusted for an arbitrary viewer’s position by experimental parameters given by a grid-based interpolation mechanism. This way, we have a flexible background plane that we can control as desired depending on the viewer’s position for a successful, geometrically correct view. Consequently, the viewer does not feel the presence of the real object and feels the virtual object merging into the real world merging them into one.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Tadahiro Fujimoto and Prof. Norishige Chiba, for the continuous support of my PhD study and research.

Prof. Tadahiro Fujimoto, for his patience, motivation, enthusiasm, vision, and immense knowledge that inspired me and my study. Always to push forward and elevate our study. His sincere, encouraging and understanding characteristics helped me in all the time of intense researching and writing of this thesis. With his guidance, I was able to carry out our research and to have a work that we are to be proud of. It was a great privilege and honour to learn and work under his guidance. I am extending my heartfelt thanks to his wife and kids, for their acceptance and patience during all this time working on completion of our research work. I also thank my fellow lab mates of Fujimoto Kenkyushitsu for their support and advice.

Certainly, I would like to thank my family. They are my heroes; there is nothing in this world that can represent my love and admiration for my parents. All the support and love, being the best parents that someone could ever ask for I thank you from the bottom of my heart.

TABLE OF CONTENTS

1. Introduction	
1.1. Background	1
1.2. Objective	5
2. Related Work and Argument	
2.1. Related Work	8
2.2. Argument	11
3. View-Dependent Projection Mapping Enhanced by Real Background	
3.1. Outline	15
3.2. Practice Environment	21
3.3. Generation of View Composite Image	22
3.3.1. Viewer Virtual Object Image	22
3.3.2. Viewer Background Image	24
3.3.3. Viewer Composite Image	29
3.4. Generation of the Projection Image	29
3.5. Projection of the Projection Images	33
3.6. Adjustment of Background Parameters	33
3.6.1. Practical background Parameters	33
3.6.2. Interpolation of Background Parameters	36
3.7. Practice	38
3.7.1. Tracking of the Viewer	38
3.7.2. Generation of the Viewer Virtual Object Image	39

4. Experiment	42
4.1. Experiment in Virtual Environment	42
4.1.1. View-Dependent Projection Mapping	46
4.1.2. Adjustment of Projected Real-Background	47
4.1.3. Projection of Virtual Object and Real-Background	49
4.1.4. Interpolation of Background Parameters	50
4.2. Experiment in Real-Environment	55
4.3. Experiment in Real-Environment with a Specific 3D Model	62
Conclusion	60
References	62
Appendix	66

CHAPTER 1

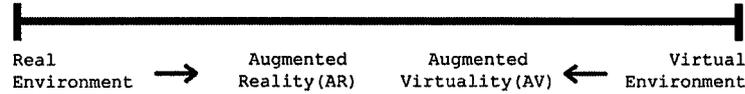
INTRODUCTION

1.1 Background

Augmented Reality (AR) [1][2] is a computer-based technology to generate a virtual world by adding a variety of virtual things, such as virtual objects, to the real world to augment the physical environment. AR was defined in 1997 [37], the researcher indicated that AR is not only restricted by the specialized hardware, whereas also it brings the real and virtual objects together in a physical environment. There is a fascination with AR because the technology records the real and virtual world together, and they run jointly in real-time in three dimensions. Augmented Reality and Virtual Reality are much different from each other, but they also have much in common. Our minds and bodies exist in a mix of space and time continuum that we all perceive as reality. Nowadays, if we want, we can even exist inside a computer-generated mix of space that we know as Virtual Reality (VR). However, there are other types of reality in between these extreme ends where the real physical world and virtual reality intertwine to varying degrees, one of which being augmented Reality (AR)[]. VR immerses the users in a synthetic virtual world entirely. The only tie to the physical world would be the space used in the VR experience.

To fully convince the brain, the VR uses calibration technology to create a safe zone for VR activities. On the other hand, AR technology augments reality by superimposing virtual objects and cues upon the real-world in real-time.

Mixed Reality (MR)



Milgram's Reality-Virtuality Continuum [38]

The virtual world is usually displayed in an image, including a video, for viewers to see using various kinds of display devices, such as a monitor, a tablet, a smartphone, a head-mounted display, and special glasses. In this case, generally, the viewer has his/her display device to see an image generated for only him/her according to his/her viewing position. This means that multiple viewers see different images, respectively. On the other hand, spatial augmented reality (SAR) [1][2][3][4] augments the real world without such display devices. SAR usually uses a projector to display an image by projecting it onto a real object's surface, such as a screen, a wall, a building, a house, a car, and even a human in the real world. A viewer sees the image projected on the actual object's surface by the naked eyes; that is, multiple viewers simultaneously see the same image. In SAR, the display of an image does not depend on a viewer [2]. Therefore, SAR enables multiple workers to collaborate by seeing a shared virtual world simultaneously without their own display devices [2]. Projection mapping is a well-known technology to achieve SAR in many fields, such as entertainment, advertisement, and education. Many research works of SAR using projection mapping have been done by a variety of researchers so far. In the following, we use "SAR" and "projection mapping" has the same meaning if the usage causes no problem. In AR, a viewer feels as if a virtual object existed in the real world if the object's appearance is observed correctly from an arbitrary position around the object. The display to show a virtual object's correct appearance according to a viewer's position is usually called view-dependent display, which is exceedingly important in AR.

As one typical way to determine view-dependent display, AR systems use a marker to indicate the exact position of a virtual object in the 3D space; by obtaining the object's relative positions and a viewer by using the marker. the viewer's display device is given a correct image of

the object seen from his/her position. According to a viewer's position, some systems use a screen made from special material to reflect a different correct image. "Parallax" is a visual effect caused by the difference in left and right eyes. In a typical stereoscope system, a viewer wears special glasses that give the left and right eyes different images generated using the parallax effect. "Parallax scrolling" is a well-known computer graphics technique to give the viewer a sense of immersion in a 3D virtual space using 2D images. This technique yields a feeling of the depth difference between foreground and background objects by moving foreground images faster than background images according to the viewer's movement. In the methods described above, an image seen by a viewer depends on the position of the viewer's eyes. On the other hand, in SAR, multiple viewers usually see a virtual object's common image by the naked eyes without their own display devices. A SAR system sometimes, or often, does not need the view-dependent display of a virtual object. If needed, one easy way is to project an image of a virtual object onto the surface of a real object whose shape is the same as that of the virtual object. In this case, the virtual object's appearance projected on the real object's surface is observed correctly from an arbitrary viewer's position. However, it is practically difficult to prepare such a real object for an arbitrary virtual object. Then, the difference between their shapes causes a problem of incorrect appearance on a projected image. For example, in Figure 1, the correct appearances of a virtual object, teapot, in (b1) and (b3) become the incorrect appearances in (d1) and (d3) by projecting a correct image (b2) on the surface of a real object, cube. The detail is explained in Section 3. The difference between the shapes of a virtual object and a real object causes another problem. The real object's surface has "empty areas" on which has no virtual object projected on. Such empty areas make a viewer not feel the virtual object merging into the real world. In (d1), (d2), and (d3) of Figure 1, empty areas on the cube's surface are coloured in grey. In the real world, the cube is seen with a real background behind it as shown in (g1), (g2), and (g3), in which the coloured boxes simulate the real background. The empty areas degrade the viewer's feeling that the virtual teapot merges into the real background.

1.2 Objective

The first problem described above, that is, the incorrect appearance of a virtual object has been one of the main problems, probably the most important problem, in SAR. Thus, many methods for view-dependent projection to solve the problem have

been proposed. On the other hand, as far as we know, there is no existing method to treat the second problem caused by the empty areas. In this paper, we propose a method to eliminate undesired empty areas by projecting the real background behind the real object in a view-dependent way. To treat a far background in a large space, even outdoors, the real background shape is not measured by an RGB-D camera with a depth range limit but approximated by a plane based on homography appropriately and given colours captured by a standard RGB camera. Although the plane is determined manually, a practical way is proposed to easily adjust the plane, intuitively, and efficiently by devising effective background parameters. A simple grid-based interpolation mechanism for the parameters provides an appropriate plane to an arbitrary viewer's position. Consequently, the projected real background shows its appearance as correctly as possible to match the real background directly seen by the viewer seamlessly along the real object's contour according to the viewer's position. Our method makes the viewer not feel the real object's presence but feel the virtual object merging into the real world. In Figure 1, ideal views seen by the viewer are shown in (h1), (h2), and (h3). This study aims to study and clarify how deeply the merging effect affects the viewer's feeling by our method using the simple background approximation and interpolation mechanism. In general, to realize view-dependent projection, an existing method using a single/multiple off-the-shelf projector/projectors and an ordinary real object with no specialized material sacrifices significant merit of SAR; only a single viewer is allowed to see a virtual object's correct appearance. The same applies to our method. However, our method's simplicity enables the expansion to the view-dependent projection for multiple viewers simultaneously by utilizing a special device and a real object with specialized material used in an existing method developed for multiple viewers.

1.2 Objective

The first problem described above, that is, the incorrect appearance of a virtual object has been one of the main problems, probably the most important problem, in SAR. Thus, many methods for view-dependent projection to solve the problem have been proposed. On the other hand, as far as we know, there is no existing method to treat the second problem caused by the

empty areas. In this paper, we propose a method to eliminate undesired empty areas by projecting the real background behind the real object in a view-dependent way. To treat a far background in a large space, even outdoors, the real background shape is not measured by an RGB-D camera with a depth range limit but approximated by a plane based on homography appropriately and given colours captured by a usual RGB camera. Although the plane is determined manually, a practical way is proposed to adjust the plane easily, intuitively, and efficiently by devising effective background parameters. A simple grid-based interpolation mechanism for the parameters provides an appropriate plane to an arbitrary viewer's position. Consequently, the projected real background shows its appearance as correctly as possible to match the real background directly seen by the viewer seamlessly along the real object's contour according to the viewer's position. Our method makes the viewer not feel the presence of the real object but feel the virtual object merging into the real world. In Figure 1, ideal views seen by the viewer are shown in (h1), (h2), and (h3). This study aims to study and clarify how deeply the merging effect affects the viewer's feeling by our method using the simple background approximation and interpolation mechanism. In general, in order to realize view-dependent projection, an existing method using a single/multiple off-the-shelf projector/projectors and an ordinary real object with no specialized material sacrifices significant merit of SAR; only a single viewer is allowed to see a virtual object's correct appearance, and multiple viewers are not allowed to see it simultaneously. The same applies to our method. However, the simplicity of our method enables the expansion to the view-dependent projection for multiple viewers simultaneously by utilizing a special device and a real object with specialized material used in an existing method developed for multiple viewers.

The first problem described above, that is, the geometrically incorrect appearance of a virtual object has been one of the main problems, probably the most important problem, in SAR. Consequently, many methods for view-dependent projection have been proposed to solve the problem have been mentioned. There are a lot of methods and solutions to achieve a precise application of image-based modelling and rendering techniques. Acquiring geometric models of environments has been the subject of research in interactive image-based modelling techniques. To address problems, it is essential to make judicious use of all the available views, especially

when a particular surface is seen from different directions in multiple images. In our case, it would be not so much of a problem because we created a 3D Model of our desired projection object from the physical world.

On the other hand, there is no existing method to treat the second problem caused by the empty areas. In this study, we propose a method to eliminate undesired empty areas by projecting the real object's real background in a view-dependent way. To treat a far background in a large space, even outdoors, the real-background shape is not measured by an RGB-D camera with a depth range limit but approximated by a plane based on homography appropriately and given colours captured by a standard RGB camera. Although the plane is determined manually, a practical way is proposed to easily adjust the plane, intuitively, and efficiently by devising effective background parameters. A simple grid-based interpolation mechanism for the parameters provides an appropriate plane to an arbitrary viewer's position.

Consequently, the projected real background shows its appearance as correctly as possible to match the real background directly seen by the viewer seamlessly along the real object's contour according to the viewer's position. Our method makes the viewer not feel the real object's presence but feel the virtual object merging into the real world. In Figure 1, ideal views seen by the viewer are shown in (h1), (h2), and (h3). This study aims to study and clarify how deeply the merging effect affects the viewer's feeling by our method using the simple background approximation and interpolation mechanism.

An existing method using a single/multiple off-the-shelf projector and an ordinary real object with no specialized material sacrifices significant merit of SAR to realize view-dependent projection. Only a single viewer can see a virtual object's correct appearance, and multiple viewers are not allowed to see it simultaneously. The same applies to our method. However, our method's simplicity enables the expansion to the view-dependent projection for multiple viewers simultaneously by utilizing a special-device and a real object with specialized material used in an existing method developed for multiple viewers.

CHAPTER 2

RELATED WORK AND ARGUMENT

2.1 Related Work

One of the pioneering technologies concerning SAR is “the office of the future” [5] proposed in 1998. The representative works proposed in recent 20 years are surveyed and categorized intelligibly in [6][7]. As described in Section 1, the most critical problem in SAR is the view-dependent projection to show a virtual object’s correct appearance, which is often called *perspectively-correct* appearance, according to a viewer’s position. To achieve complete *perspectively-correct* appearance, an appropriate *projection image* that a projector projects on a real object surface need to be obtained. Such a projection image depends on a projector’s position, pose, optical properties, and a real object surface’s geometry and material properties such as texture and reflectance. Thus, first, the projector and the real object surface need to be calibrated. The calibration in SAR consists of *geometric calibration* and *photometric calibration* [6][7]. The purpose of geometric calibration is to make a *projected image* on a real object surface geometrically correct without distortion. When multiple projectors are used, projected images by the respective projectors are partially overlapped on a real object surface to reconstruct a virtual object by their correct arrangement. The purpose of photometric calibration is to make a projected image photometrically correct such that the projected image’s colours become as similar as possible to its original image’s colours according to a real object surface’s material property. When

multiple projectors are used, the respective projected images' colour intensities are adjusted on the overlap areas. A typical calibration method uses a camera to capture a projected image for automatic geometric and photometric calibration. Such calibration is usually called *projector-camera calibration* [6][7].

Our method involves geometric calibration and does not treat photometric calibration. Therefore, we mention geometric calibration in the following. In geometric calibration, the geometry of a real object surface and the projector's intrinsic and extrinsic parameters are estimated. In projector-camera calibration, the intrinsic and extrinsic parameters of a camera are also estimated. Typically, the correspondence between projector pixels and camera pixels, which is often called *geometric registration*, is estimated, and used. Some methods estimate unknown parameters of a projector by using known parameters of a camera while others estimate all unknown parameters of a projector and a camera. Besides, a real object surface's geometry is unknown and estimated in some methods while the geometry is known and used to estimate unknown parameters of a projector and a camera in others. "Shader Lamps" [8] is an early well-known SAR work. In this work, the 3D shape of a real object is measured by a 3D touch probe scanner to obtain a 3D model with the same shape as a virtual object. Multiple projectors are then calibrated by projecting markers from each projector onto the real object's surface and matching the projector's marker pixels with 3D points on the surface. In general, geometric projector-camera calibration is categorized into *semi-automatic calibration* and *self-calibration* [7]. A semi-automatic calibration method uses a specialized apparatus for calibration [9][10][11][12][13]. Typically, the projector-camera pixel correspondence is estimated by projecting structured light patterns, such as a chessboard pattern, onto a planar surface, such as a board or a screen, by a projector and capturing it by a camera. A homography matrix to relate the planar surface with the projector's image planes and the camera is often used. On the other hand, a self-calibration method does not use a specialized apparatus for calibration [14][15][16][17][18]. Typically, structured light patterns are projected onto a non-planar surface whose geometry is unknown. In addition, recently, various *dynamic projection mapping* methods have been proposed to calibrate a real object's movement, such as human face

and clothes, and give an appropriate projection in real-time. For example, an object silhouette [19], a rigid surface [20], and a non-rigid surface [21][22][23] are treated.

As described in Section 1, an existing SAR method using an ordinary configuration basically enables view-dependent projection for only a single viewer. For example, in “HeatSpace” [24] and “OptiSpace” [25] systems, an image of a virtual object is projected so as to show its correct appearance on the environment in a room. These systems use projectors and Kinect sensors. Each system measures not only the room environment but also the movement and viewing behaviour of a single viewer during a certain time, analyzes the measured data, and automatically determines the optimal surface, such as a wall and a desk, on which the virtual object image should be projected in the environment. The movement of the viewer is restricted within an analyzed small area, such as sitting in a chair, and the virtual object is given its correct appearance according to the small area and the optimal surface’s geometry. In the “dyadic SAR system” of [26], two viewers stand face-to-face with each other near the opposite walls in a room and collaborate by interacting with a common virtual object. This system uses three projector-camera pairs mounted on the ceiling; each camera is a Kinect sensor. The two pairs face the two viewers, respectively. Each pair measures the surfaces of one viewer and the back environment behind him/her by the Kinect sensor and projects a virtual object image on the surfaces by the projector. Another viewer sees the projected image. The virtual object is displayed in the respective projected images such that the two viewers perceive existing in the same 3D position. The remaining pair measures and projects the projection image according to the environment between the physical above two pairs’ environments. Even in this system, each projected image is given the virtual object’s correct appearance for only a single viewer. The above systems can work in only a non-large room-scale environment because the real-time capturing of the viewer and the environment by a Kinect sensor, generally an RGB-D camera with a depth range limit, is necessary.

On the other hand, many SAR systems to allow multiple viewers to see the correct appearance of a typical virtual object simultaneously have been proposed [27][28][29][30][31][32][33]. Such SAR is often called *light field projection*. These systems use specialized configurations consisting of special devices and real object surfaces; typically, multiple projectors and an anisotropic or

lenticular surface made from special material to reflect different images to different directions are used. In general, the development of view-dependent projection technologies for multiple viewers using ordinary configurations is an important future work in SAR.

2.2 Argument

In general, AR's main purpose is to add a virtual object in the real world to augment the real environment by making a viewer feel the virtual object merging into the real world; multiple viewers can experience the augmentation simultaneously and each viewer usually uses his/her display device. On the other hand, one purpose of SAR is to decorate the surface of a real object with various images by projecting them; this is also enjoyed by multiple viewers simultaneously and each viewer sees the decoration by the naked eyes without a display device. This purpose has been achieved for not only non-dynamic objects [8][34] but also dynamic objects [19][20][21][22] Error! Reference source not found. In this case, a 3D model with the real object's shape needs to be obtained as a virtual object by measuring or estimating the shape. Meanwhile, many SAR methods have been proposed to achieve another purpose, that is, to project an image of a virtual object on the surface of a real object, such as a screen and a wall, from which the virtual object does not originate. This means that the shape of the real object has no relation with that of the virtual object. A typical method with this purpose aims to provide the view-dependent projection of the virtual object with its correct appearance to only a single viewer or multiple viewers simultaneously, without any consideration for the relation between the shapes. From the viewpoint of merging the virtual object into the real world, there is no problem if the real object should be shown as an element in the real world [26]. On the other hand, there are practically many cases in which the real object is used only for the projection and its presence should not be shown. Although one solution is to prepare a real object whose shape is the same as that of the virtual object, it is difficult in general. Consequently, as described in Section 1, the problem of empty areas that have no virtual object's projection on the real object's surface happens. In view-dependent projection to provide a virtual object's correct appearance to a viewer, or to multiple viewers, moving freely

within a wide range around a real object, the empty area problem becomes more serious. This results in degrading the merging effect of the virtual object into the real world.

As far as we know, there is no method to solve the empty area problem. The system of [29] uses a specialized mirror rotating at high speed to reflect a virtual object image to a viewer. The real background behind the rotating mirror is seen through it intermittently, which results in giving the viewer a visual effect by which the virtual object is floating in front of the real background without feeling empty areas. However, the viewer unavoidably perceives the presence of the mirror. Our method solves the empty area problem by projecting a real background as well as a virtual object in a view-dependent way. This results in achieving AR's main purpose by making a viewer feel the virtual object merging into the real world by the naked eyes.

Our method uses a projector and a camera, which is used not for calibration but for capturing a real background. We assume that the projector's intrinsic and extrinsic parameters and the camera and the geometry of a real object are already known by geometric calibration. An arbitrary photometric calibration method can be used for our method. However, in this paper's experiments, photometric calibration was not applied and a white object with a diffuse reflection surface was used as a real object. Besides, our method uses a tracking sensor to track a moving viewer in real time. The real background shape is appropriately approximated by a plane based on homography and given colors captured by the camera. This enables our method to treat a far background in a large space, even outdoors. As described in Section 2.1, homography is often used to estimate a projector-camera pixel correspondence in a geometric calibration method using a planar surface. In this case, a camera is used as a viewer's eye; in this sense, we can also call it *projector-viewer pixel correspondence*. This is used to obtain a projection image to project by a projector from a given image that the viewer wants to see; the projection image is "appropriately distorted" to show its correct appearance without distortion to the viewer. On the other hand, our method uses homography for a different purpose. Homography is used to estimate a *viewer-camera pixel correspondence*. This is used to obtain an approximated image that a viewer should see from a true image captured by a camera. The plane to approximate the real background shape is adjusted by effective background parameters, which are interpolated on a 3D grid space to provide an

appropriate plane to an arbitrary viewer's position. Our method uses the "two-pass algorithm" [5][35][36] to obtain a projection image containing a virtual object and a real background for view-dependent projection. This image's projection results in displaying the virtual object's correct appearance on a real object's surface and its empty area, we projected real background appropriately eliminates areas according to a viewer's position. Our method uses an ordinary configuration consisting of a projector, a camera, and an arbitrary real object, all of which are not specialized. Thus, in the same way as an existing method using such an ordinary configuration, only a single viewer can see the virtual object's correct appearance. However, our method can be expanded to view-dependent projection for multiple viewers by utilizing a specialized configuration used in an existing method developed for multiple viewers.

CHAPTER 3

VIEW-DEPENDENT PROJECTION MAPPING ENHANCED BY REAL BACKGROUND

3.1 Outline

If a real object's surface is given the projection of a virtual object and their shapes are different from each other, the virtual object's correct appearance cannot be seen from an arbitrary viewer's position. This is explained in Figure 1, in which projection mapping is simulated virtually by a graphic library OpenGL. In the following, the notation "(a1,2)" means "(a1) and (a2)", and "(a,b1)" means "(a1) and (b1)". In this simulation, a virtual object is a teapot shown in (b1,2,3), and a real object is a cube shown in (c1,2,3). The images (a1,2) are top views to show the positions of the teapot, the cube, a projector, and a viewer on the horizontal plane. The images (b,c2) are seen from the projector's position while the images (b,c1) and (b,c3) are seen from the left and right viewer's positions. The images (b1,2,3) are obtained by rendering the teapot from the respective positions; they have the teapot's correct appearances. The grey colours in (c1,2,3) mean depth values from the respective positions. If the image (b2) is used as a projection image and projected onto the cube's surface by the projector, the surface is seen from the respective positions as shown in (d1,2,3); the red part on the cube's surface in (a2) is given the projection of the teapot. The teapot's appearance is correct in (d2), while it is incorrect in (d1,3). The images (e1,2,3) are obtained by using a sphere instead of the cube. The comparison between (d1,3) and (e1,3) shows

that the incorrect appearance depends on the real object's shape as well as the virtual object's shape. Our method solves the "incorrect appearance" problem by the "two-pass algorithm" [5][35][36], which "correctly distorts" an image with correct appearance to obtain an appropriate projection image to project by a projector according to the positional relation between the projector and a viewer. In the images (d1,2,3) and (e1,2,3), the grey-coloured empty areas on the surface of the cube/sphere are not given the projection of the teapot. The coloured boxes in (f1,2,3) simulate the real background behind the cube/sphere seen from the respective positions in the real world. Then, for the cube, the actual views seen from the positions become (g1,2,3). In addition to the incorrect appearance, the empty areas greatly degrade the viewer's feeling that the teapot merges into the real world. To avoid this problem, our method captures an image of the real background behind a real object by a camera and projects the image onto such empty areas. This results in making a viewer not feel the presence of the real object. The capture and projection of the real background are made in real-time. Thus, especially, objects moving in real-time, such as humans, in the real background enhance the merging effect of a virtual object into the real world. The view-dependent projection of the teapot with its correct appearance and the real background behind it provides the ideal views in (h1,2,3), in which the teapot looks as if it were floating in the air in front of the real background. In order to avoid a viewer's strange feeling, the real background projected on a real object's surface should match the real background directly seen by the viewer seamlessly along the real object's contour. One solution is that the viewer carries a camera at the eye position to use a captured background image to project directly. However, this imposes a burden on the viewer. From a practical viewpoint, the camera should be fixed at a certain position while the viewer moves freely. In this case, a background image captured from the camera's position needs to be converted into the image seen from the viewer's position. An easy solution is to use an RGB-D camera to obtain not only a colour image but also a depth image of objects existing in the background. The depth image is used to obtain the 3D shapes of the background objects, and the 3D shapes are rendered from the viewer's position to obtain the objective image. However, practically, the quality of such an image obtained by using a reasonable and popular RGB-D camera, such as a Kinect sensor, is not so high because of the measurement errors of depth values and the pixel

correspondence errors between a colour image and a depth image. Besides, as a crucial problem, the depth values that the RGB-D camera can measure is limited to a certain range; for example, the practical depth range of a Kinect sensor is limited from 0.5 to 4.0 meters. This means that a background far from the camera cannot be measured. A traditional solution in computer vision to estimate depth values from colour images captured by multiple RGB cameras can treat a far background. However, it is troublesome and difficult to set up and calibrate the cameras accurately. Besides, the estimation is not always exact and stable. Therefore, our method uses only a colour image captured by a single RGB camera to treat a far background in a large space, even outdoors, easily and stably. A background image captured by the camera is converted into the image seen from a viewer's position based on homography by using a background plane to approximate the 3D shape of the background, that is, the 3D shapes of the background objects. The converted background image is "correctly distorted" in the same way as the virtual object image to obtain an appropriate projection image to be projected by the projector. The converted background image by the approximation needs to be like the real background directly seen by the viewer as exactly as possible. Particularly, to make the viewer not feel the presence of the real object but feel the virtual object merging into the real world, it is the most important that the converted background projected on the real object's surface matches the directly seen real background seamlessly along the real object's contour according to the viewer's position. The algorithm of our method executes the following steps, as shown in Figure 2. Steps 1 and 2 are done in the virtual space of a computer while Step 3 is done in the real space; that is, the real world. Step 2 is known as the two-pass algorithm [5][35][36]. Step 1: A composite viewer image seen from a viewer's position is generated by the next sub-steps, as shown in Figure 2 (a1). Step 1-1: A virtual object is rendered from the viewer's position to generate a viewer virtual object image (b1). Step 1-2: A background image captured by the camera, called camera background image (c1*), is converted into a background image seen from the viewer's position, called viewer background image (c1). Step 1-3: The viewer virtual object image (b1) and viewer background image (c1) are combined into a composite viewer image (d1). Step 2: The composite viewer image (d1) is used as a texture and projectively-mapped on the real object's surface. The textured surface is rendered from the projector's position to

generate a projection image (d2), as shown in (a2). The image (d2) consists of (b2) and (c2), which are generated from (b1) and (c1), respectively. Step 3: The projection image (d2) is projected on the real object's surface, which results in showing the view of the virtual object's correct appearance (d3), as shown in (a3). The view (d3) consists of (b3) and (c3). The view (e3) shows the background directly seen by the viewer in the real world. Then, the actual view seen by the viewer is (d3'). The view (c3') contains only the background. The details of these steps are explained below. The main novelty of our method is in Step 1-2. The viewer background image is generated from the camera background image by a background plane defined by appropriate background parameters interpolated on a 3D grid space. In the following, we use "background" instead of "real background" if it does not mislead a reader.

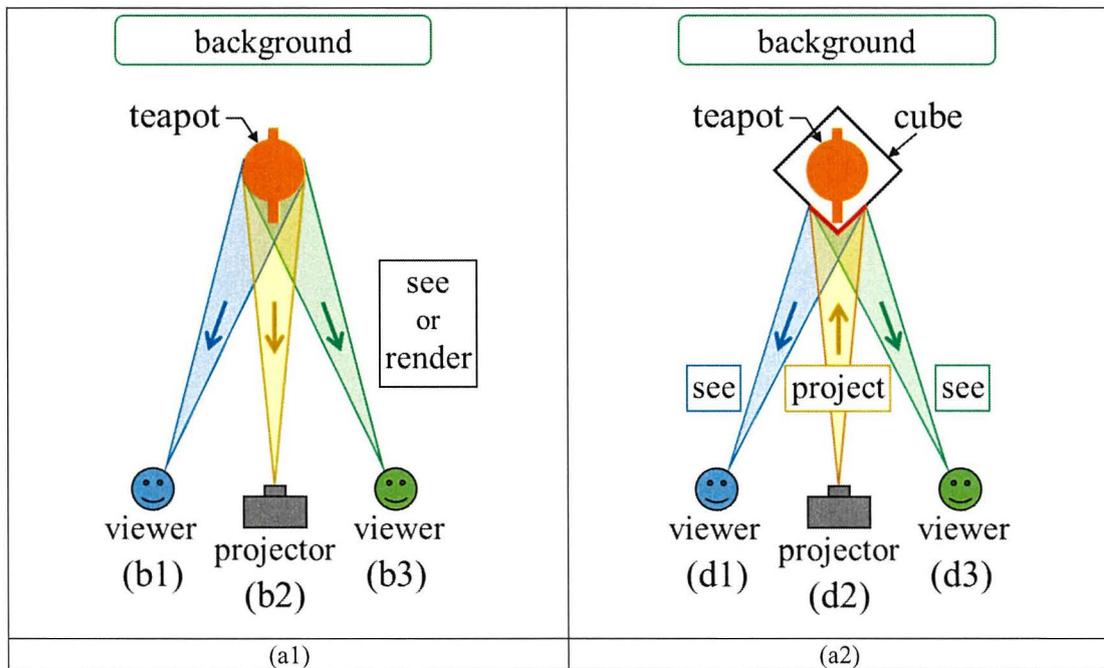


Figure 1. Correct/incorrect appearances of projected images and enhancement by real background.

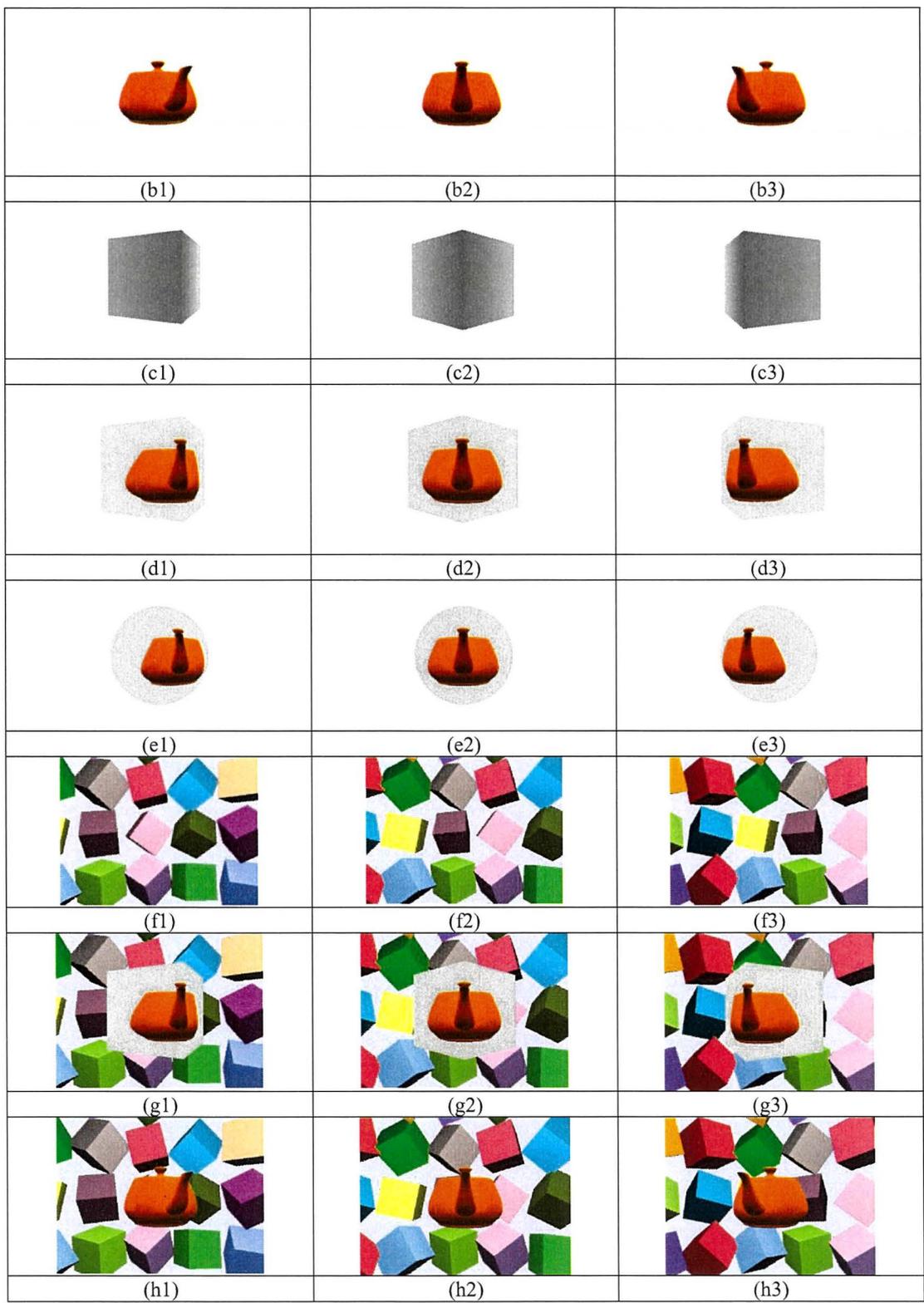


Figure 1. Continued.

3.2 Practice environment

We need to prepare the same practice environment in both real space and virtual space. In real space, a real object, a projector, and a camera are prepared. In the virtual space, an accurate 3D geometric model of the real object is made. Then, the registration between the spaces is done by placing the real object, the projector, and the camera in the same positions and orientations in both spaces. Their positions and orientations are fixed during the viewing practice of a viewer.

The world coordinate system $[x, y, z]$ by the rectangular coordinate system is given in the practice environment. The coordinates x and y define the horizontal plane, and the coordinate z defines the vertical direction. A viewer usually moves around the real object. Therefore, the real object is put on the origin O of the world coordinate system, and the polar coordinate system $[r, \theta, \varphi]$, which is defined by

$$x = r \cos \theta \sin \varphi, \quad (1)$$

$$y = r \sin \theta \sin \varphi, \quad (2)$$

$$z = r \cos \varphi, \quad (3)$$

is used for the viewer's position.

3.3 Generation of viewer composite image

Using the above environment, a viewer composite image is generated for an arbitrary viewer's position by Step 1 of our algorithm. This step consists of the three sub-steps described below.

3.3.1 Viewer virtual object image

A viewer virtual object image is generated in Step 1-1. In the virtual space, a 3D geometric model of a virtual object is made. It is scaled and positioned to fit inside the real object. Then, it is rendered from the viewer's position to generate a viewer virtual object image.

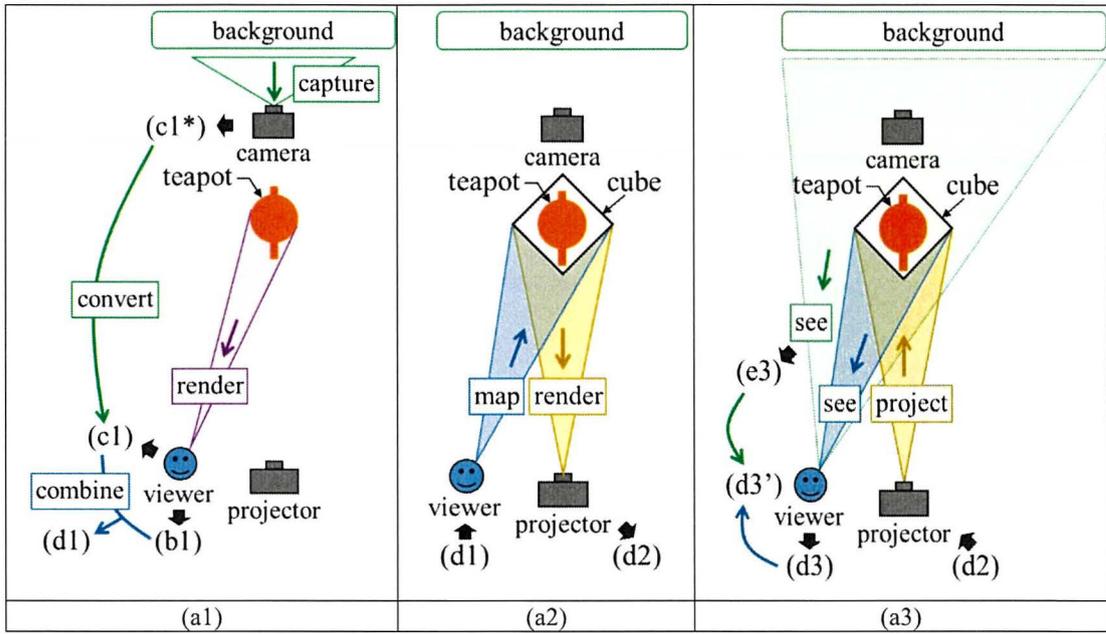


Figure 2. Our Algorithm.

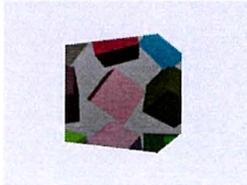
	Step 1	Step 2		Step 3	
		Virtual Space		Real Space	
Virtual Object					
		(b1)	(b2)	(b3)	(b4)
Background					
	(c1*)	(c1)	(c2)	(c3)	(c3')
Composite					
		(d1)	(d2)	(d3)	(d3')
	Camera's View	Viewer's View	Projector's View	Viewer's View	

Figure 2. Our algorithm. Continued.

3.3.2 Viewer background image

A viewer background image is generated in Step 1-2. It is obtained from a camera background image by using a background plane to approximate the actual shape of a background. In the following, first, we present a method using a homography matrix. Then, we mention another method using projective texture mapping.

(1) Homography-based method

(1-1) Explanation of homography-based method

A homography is a projection from a plane to another plane by a projective transformation in the 3D space. When an image on a plane P is seen from two cameras, the images of these cameras are directly-related to each other by a 3×3 homography matrix. The 2D pixel coordinates $\mathbf{U}_1 = [u_1, v_1]^T$ of a camera C_1 and $\mathbf{U}_2 = [u_2, v_2]^T$ of a camera C_2 have the relationship

$$\lambda \tilde{\mathbf{U}}_1 = \lambda \begin{bmatrix} \mathbf{U}_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} \mathbf{U}_2 \\ 1 \end{bmatrix} = H \tilde{\mathbf{U}}_2 \quad (4)$$

by a homography matrix

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \quad (5)$$

where $\lambda \neq 0$ is a constant. The symbol T means a transposed matrix.

With respect to the camera C_m , $m = 1, 2$, the intrinsic parameter matrix A_m and the extrinsic parameter matrix $[R_m \ \mathbf{t}_m]$ defined by the rotation matrix R_m and the translation vector \mathbf{t}_m are given by

$$A_m = \begin{bmatrix} f_{xm} & s_m & c_{xm} \\ 0 & f_{ym} & c_{ym} \\ 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$R_m = \begin{bmatrix} r_{11}^m & r_{12}^m & r_{13}^m \\ r_{21}^m & r_{22}^m & r_{23}^m \\ r_{31}^m & r_{32}^m & r_{33}^m \end{bmatrix}, \quad (7)$$

$$\mathbf{t}_m = \begin{bmatrix} t_{xm} \\ t_{ym} \\ t_{zm} \end{bmatrix}. \quad (8)$$

In the matrix A_m , the parameters f_{xm} and f_{ym} are the focal lengths in terms of pixels, c_{xm} and c_{ym} are the coordinates of the principal point, and s_m is the skew coefficient. The 2D pixel coordinates \mathbf{U}_m and the 3D world coordinates $\mathbf{X} = [x, y, z]^T$ have the relationship

$$\lambda_m \tilde{\mathbf{U}}_m = \lambda_m \begin{bmatrix} \mathbf{U}_m \\ 1 \end{bmatrix} = A_m [R_m \mathbf{t}_m] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = A_m [R_m \mathbf{t}_m] \tilde{\mathbf{X}}, \quad (9)$$

where $\lambda_m \neq 0$ is a constant. Besides, the plane P is defined by

$$\mathbf{N}^T \mathbf{X} + d = n_x x + n_y y + n_z z + d = 0, \quad (10)$$

where $\mathbf{N} = [n_x, n_y, n_z]^T$, $|\mathbf{N}| = 1$, is a unit normal vector, and d is a constant. Then, the homography matrix H is represented by

$$H = A_1 \{ (\mathbf{N}^T R_2^{-1} \mathbf{t}_2 - d) R_1 + (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2) \mathbf{N}^T \} R_2^{-1} A_2^{-1}. \quad (11)$$

The derivation of Equation (11) is described in Section 3.3.2 (1) (1-2).

If the angles of view α_{xm} and α_{ym} , and the numbers of pixels M_{xm} and M_{ym} of the camera C_m are known, then the parameters f_{xm} , f_{ym} , c_{xm} , and c_{ym} are obtained by

$$f_{xm} = M_{xm} / \{2 \tan(\alpha_{xm}/2)\}, \quad (12)$$

$$f_{ym} = M_{ym} / \{2 \tan(\alpha_{ym}/2)\}, \quad (13)$$

$$c_{xm} = M_{xm}/2, \quad (14)$$

$$c_{ym} = M_{ym}/2. \quad (15)$$

Besides, it is often assumed that

$$s_m = 0. \quad (16)$$

We consider the camera C_1 as a camera to capture a background, the camera C_2 as a viewer, and the plane P as a background plane to approximate the shape of the background. Then, a viewer background image is obtained from a camera background image by determining the homography matrix H in Equation (11) and using the correspondence between the 2D pixel coordinates \mathbf{U}_1 and \mathbf{U}_2

in Equation (4). This 2D computation using the 3×3 matrix H is so efficient compared to the usual 3D computation using a 4×4 matrix to convert an image by using the relation among the image planes of the two cameras and the plane P in the 3D space.

The homography matrix H in Equation (11) consists of the intrinsic and extrinsic matrices of the cameras C_m , $m = 1, 2$, and the parameters of the plane P . These are obtained as follows. The camera C_1 is considered as a camera to capture a background. The intrinsic parameter matrix A_1 can be obtained by camera calibration. Instead, it can be also obtained by Equations from (12) to (16) although it is in some level influenced by the errors of the angles α_{xm} and α_{ym} given in the specification and the assumption for the coefficient s_m by Equation (16). The rotation matrix R_1 and the translation vector t_1 are obtained from its position and viewing direction fixed in the practice environment.

The camera C_2 is considered as a viewer. The matrix A_2 can be given arbitrarily because a consistent result can be obtained in Step2 by using the same matrix in the texture mapping and rendering process. The matrix R_2 and the vector t_2 are obtained from the viewer's position and viewing direction, which are obtained by tracking the viewer in real time as described in Section 3.7.1. Finally, with respect to the plane P , the constant d is obtained from the vector N , the world coordinates X_0 of an arbitrary reference point Q_0 , and the distance D_P from the point Q_0 to the plane P as follows:

$$d = -N^T X_0 - D_P, \quad (17)$$

where the vector N has the same direction as the direction from the point Q_0 to the plane P . We simply use the origin O of the world coordinate system as the point Q_0 ; its coordinates are $O =$

$[0, 0, 0]^T$. Then, we obtain

$$d = -D_P. \quad (18)$$

The world coordinates of the vector N can be defined by using two coordinates θ_P and φ_P and setting $r = |N| = 1$ in Equations (1), (2), and (3) as follows:

$$n_x = \cos \theta_P \sin \varphi_P, \quad (19)$$

$$n_y = \sin \theta_P \sin \varphi_P, \quad (20)$$

$$n_z = \cos \varphi_P. \quad (21)$$

Among the above-described elements to define the homography matrix, only the three parameters, and to define the plane cannot be determined automatically when there is no information about the shape of a background. Thus, we call them *background parameters* and determine them manually. We adjust the parameters to make the plane approximate the background's shape optimally such that a resulting viewer background image becomes as similar as possible to the actual background view seen from the viewer's position. However, it is not practically easy to obtain the optimal plane by using the above three parameters. We improve them into other parameters in Section 3.6.1.

(1-2) Derivation of homography matrix.

The homography matrix H of Equation (11) is obtained as follows.

The homography matrix H of Equation (11) is obtained as follows. In the following, for convenience, some equations used in Section 3.3.2 (1-1) are written again.

Each of two cameras C_m , $m = 1, 2$, has an intrinsic parameter matrix A_m and an extrinsic parameter matrix $M_m = [R_m \ \mathbf{t}_m]$ defined by a rotation matrix R_m and a translation vector \mathbf{t}_m as follows:

$$A_m = \begin{bmatrix} f_{xm} & s_m & c_{xm} \\ 0 & f_{ym} & c_{ym} \\ 0 & 0 & 1 \end{bmatrix}, \quad (a.1)$$

$$R_m = \begin{bmatrix} r_{11}^m & r_{12}^m & r_{13}^m \\ r_{21}^m & r_{22}^m & r_{23}^m \\ r_{31}^m & r_{32}^m & r_{33}^m \end{bmatrix}, \quad (a.2)$$

$$\mathbf{t}_m = \begin{bmatrix} t_{xm} \\ t_{ym} \\ t_{zm} \end{bmatrix}. \quad (a.3)$$

For a point Q in the 3D space, its 2D pixel coordinates $\mathbf{U}_m = [u_m, v_m]^T$ and 3D camera coordinates $\mathbf{X}_m = [x_m, y_m, z_m]^T$ of the camera C_m and its 3D world coordinates $\mathbf{X} = [x, y, z]^T$ have the relationships

$$\mathbf{X}_m = [R_m \ \mathbf{t}_m] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = M_m \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = M_m \tilde{\mathbf{X}}, \quad (\text{a.4})$$

$$\lambda_m \tilde{\mathbf{U}}_m = \lambda_m \begin{bmatrix} \mathbf{U}_m \\ 1 \end{bmatrix} = A_m \mathbf{X}_m, \quad (\text{a.5})$$

where $\lambda_m \neq 0$ is a constant. The symbol T means a transposed matrix. These are unified as follows:

$$\lambda_m \tilde{\mathbf{U}}_m = A_m M_m \tilde{\mathbf{X}}. \quad (\text{a.6})$$

Equations (a.4), (a.5) and (a.6) are given other representations using 4×4 matrices \tilde{M}_m and \tilde{A}_m as follows:

$$\tilde{\mathbf{X}}_m = \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} = \begin{bmatrix} R_m & \mathbf{t}_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} M_m \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \tilde{M}_m \tilde{\mathbf{X}}, \quad (\text{a.7})$$

$$\tilde{\mathbf{U}}_m^* = \begin{bmatrix} \lambda_m \tilde{\mathbf{U}}_m \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda_m \begin{bmatrix} \mathbf{U}_m \\ 1 \end{bmatrix} \\ 1 \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} = \tilde{A}_m \tilde{\mathbf{X}}_m, \quad (\text{a.8})$$

$$\tilde{\mathbf{U}}_m^* = \tilde{A}_m \tilde{M}_m \tilde{\mathbf{X}}. \quad (\text{a.9})$$

The following is obtained from Equations (a.7) and (a.8):

$$\tilde{A}_m \tilde{M}_m = \begin{bmatrix} A_m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_m & \mathbf{t}_m \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A_m R_m & A_m \mathbf{t}_m \\ 0 & 1 \end{bmatrix}. \quad (\text{a.10})$$

The next equations are obtained from Equation (a.7):

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \tilde{M}_m^{-1} \tilde{\mathbf{X}}_m = \begin{bmatrix} R_m^{-1} & \mathbf{t}'_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix}, \quad (\text{a.11})$$

$$\mathbf{t}'_m = -R_m^{-1} \mathbf{t}_m. \quad (\text{a.12})$$

The next equation is obtained from Equation (a.8):

$$\tilde{\mathbf{X}}_m = \begin{bmatrix} \mathbf{X}_m \\ 1 \end{bmatrix} = \tilde{A}_m^{-1} \tilde{\mathbf{U}}_m^* = \begin{bmatrix} A_m^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_m \begin{bmatrix} \mathbf{U}_m \\ 1 \end{bmatrix} \\ 1 \end{bmatrix}. \quad (\text{a.13})$$

The next equation is obtained from Equation (a.9):

$$\tilde{\mathbf{X}} = \tilde{\mathbf{M}}_m^{-1} \tilde{\mathbf{A}}_m^{-1} \tilde{\mathbf{U}}_m^*. \quad (\text{a.14})$$

The following is obtained from Equations (a.11) and (a.13):

$$\begin{aligned} \tilde{\mathbf{M}}_m^{-1} \tilde{\mathbf{A}}_m^{-1} &= \begin{bmatrix} R_m^{-1} & \mathbf{t}'_m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A_m^{-1} & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_m^{-1} A_m^{-1} & \mathbf{t}'_m \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (\text{a.15})$$

Equation (a.9) gives the camera C_1

$$\tilde{\mathbf{U}}_1^* = \tilde{\mathbf{A}}_1 \tilde{\mathbf{M}}_1 \tilde{\mathbf{X}}. \quad (\text{a.16})$$

Equation (a.14) gives the camera C_2

$$\tilde{\mathbf{X}} = \tilde{\mathbf{M}}_2^{-1} \tilde{\mathbf{A}}_2^{-1} \tilde{\mathbf{U}}_2^*. \quad (\text{a.17})$$

Then, the next equation is obtained from Equations (a.16) and (a.17):

$$\tilde{\mathbf{U}}_1^* = \tilde{\mathbf{A}}_1 \tilde{\mathbf{M}}_1 \tilde{\mathbf{M}}_2^{-1} \tilde{\mathbf{A}}_2^{-1} \tilde{\mathbf{U}}_2^*. \quad (\text{a.18})$$

The following is obtained from Equations (a.10) and (a.15):

$$\begin{aligned} \tilde{\mathbf{A}}_1 \tilde{\mathbf{M}}_1 \tilde{\mathbf{M}}_2^{-1} \tilde{\mathbf{A}}_2^{-1} &= \begin{bmatrix} A_1 R_1 & A_1 \mathbf{t}_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_2^{-1} A_2^{-1} & \mathbf{t}'_2 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} A_1 R_1 R_2^{-1} A_2^{-1} & A_1 R_1 \mathbf{t}'_2 + A_1 \mathbf{t}_1 \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (\text{a.19})$$

Then, the following is obtained from Equations (a.12), (a.18), and (a.19):

$$\begin{aligned} \lambda_1 \tilde{\mathbf{U}}_1 &= \begin{bmatrix} A_1 R_1 R_2^{-1} A_2^{-1} & A_1 R_1 \mathbf{t}'_2 + A_1 \mathbf{t}_1 \end{bmatrix} \begin{bmatrix} \lambda_2 \tilde{\mathbf{U}}_2 \\ 1 \end{bmatrix} \\ &= \lambda_2 A_1 R_1 R_2^{-1} A_2^{-1} \tilde{\mathbf{U}}_2 + A_1 R_1 \mathbf{t}'_2 + A_1 \mathbf{t}_1 \\ &= \lambda_2 A_1 R_1 R_2^{-1} A_2^{-1} \tilde{\mathbf{U}}_2 + A_1 (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2). \end{aligned} \quad (\text{a.20})$$

The world coordinates \mathbf{X} of a point Q on a plane P seen from the camera C_m satisfy

$$\mathbf{N}^T \mathbf{X} + d = n_x x + n_y y + n_z z + d = 0, \quad (\text{a.21})$$

where $N = [n_x, n_y, n_z]^T$, $|N| = 1$, is a unit normal vector and d is a constant. From Equation (a.11), the world coordinates X and the camera coordinates X_m of the point Q have the relationship

$$X = R_m^{-1}X_m + \mathbf{t}'_m. \quad (\text{a.22})$$

Then, from Equation (a.21), the camera coordinates X_m satisfy

$$N^T(R_m^{-1}X_m + \mathbf{t}'_m) + d = 0. \quad (\text{a.23})$$

From Equation (a.12), Equation (a.23) is arranged into

$$N^T R_m^{-1} X_m = N^T R_m^{-1} \mathbf{t}'_m - d. \quad (\text{a.24})$$

Then, the following is obtained from Equations (a.5) and (a.24):

$$\lambda_m N^T R_m^{-1} A_m^{-1} \tilde{U}_m = N^T R_m^{-1} \mathbf{t}'_m - d. \quad (\text{a.25})$$

From Equation (a.22), the origin O_m of the camera coordinates X_m has world coordinates

$$O_m = R_m^{-1}[0 \ 0 \ 0]^T + \mathbf{t}'_m = \mathbf{t}'_m. \quad (\text{a.26})$$

Then, from Equations (a.22) and (a.26), the viewing direction vector V_m from the origin O_m to the point Q has world coordinates

$$V_m = X - O_m = R_m^{-1}X_m. \quad (\text{a.27})$$

If $N^T V_m = N^T R_m^{-1} X_m = 0$, the vector V_m is parallel to the plane P . In this case, the plane P cannot be seen from the camera C_m . Therefore, it is assumed that

$$N^T R_m^{-1} X_m \neq 0. \quad (\text{a.28})$$

Thus, the following is obtained from Equations (a.24) and (a.28):

$$N^T R_m^{-1} \mathbf{t}'_m - d \neq 0. \quad (\text{a.29})$$

Then, the following is obtained from Equation (a.25):

$$\frac{\lambda_m}{N^T R_m^{-1} \mathbf{t}'_m - d} N^T R_m^{-1} A_m^{-1} \tilde{U}_m = 1. \quad (\text{a.30})$$

Equation (a.30) is given $m = 2$ and substituted in Equation (a.20) to obtain

$$\begin{aligned} \lambda_1 \tilde{U}_1 &= \lambda_2 A_1 R_1 R_2^{-1} A_2^{-1} \tilde{U}_2 \\ &+ A_1 (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2) \frac{\lambda_2}{N^T R_2^{-1} \mathbf{t}_2 - d} N^T R_2^{-1} A_2^{-1} \tilde{U}_2. \end{aligned} \quad (\text{a.31})$$

Equation (a.31) is arranged as follows:

$$\begin{aligned} \lambda_1(N^T R_2^{-1} \mathbf{t}_2 - d) \tilde{\mathbf{U}}_1 &= \lambda_2 A_1 \{ (N^T R_2^{-1} \mathbf{t}_2 - d) R_1 \\ &+ (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2) N^T \} R_2^{-1} A_2^{-1} \tilde{\mathbf{U}}_2. \end{aligned} \quad (\text{a.32})$$

Then, the following is obtained from Equation (a.32):

$$\begin{aligned} \lambda \tilde{\mathbf{U}}_1 &= A_1 \{ (N^T R_2^{-1} \mathbf{t}_2 - d) R_1 \\ &+ (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2) N^T \} R_2^{-1} A_2^{-1} \tilde{\mathbf{U}}_2, \end{aligned} \quad (\text{a.33})$$

where

$$\lambda = \lambda_1 (N^T R_2^{-1} \mathbf{t}_2 - d) / \lambda_2 \neq 0 \quad (\text{a.34})$$

is a new constant. Equation (a.33) is arranged as follows:

$$\lambda \tilde{\mathbf{U}}_1 = H \tilde{\mathbf{U}}_2, \quad (\text{a.35})$$

$$H = A_1 \{ (N^T R_2^{-1} \mathbf{t}_2 - d) R_1 + (\mathbf{t}_1 - R_1 R_2^{-1} \mathbf{t}_2) N^T \} R_2^{-1} A_2^{-1}.$$

(a.36)

Equations (a.35) and (a.36) are the same as Equations (4) and (11).

(2) Texture mapping-based method

If a graphic library or tool to *projectively map* a texture on a plane from one viewpoint and render the textured plane from another viewpoint is available, a viewer background image can be obtained easily by using the two-pass algorithm [5][35][36], which is used in Step 2, instead of using the homography matrix. For example, OpenGL is a typical library to have a function of *projective texture mapping*, which works by giving α_{xm} , α_{ym} , M_{xm} , and M_{ym} , and assuming $s_m = 0$. By determining a background plane P in the same way as the above-described way, this texture mapping-based method executes the next two steps: 1) A camera background image is used as a texture and projectively mapped on the plane P from the camera's position, 2) The textured plane P is rendered from a viewer's position to generate a viewer background image. The resulting viewer background image is theoretically the same as the image generated by the homography-based method.

The texture mapping-based method can use not only a plane but also any other shapes, which are available in the graphic library, to approximate the shape of a background. Even in such a situation, the shape used for the approximation needs to be easy to adjust for various background shapes. Therefore, we tested a sphere, a cylinder, and a cube as well as a plane in a preliminary experiment by using OpenGL. As a result, we found that a plane was the most useful to approximate various types of background shapes flexibly by adjusting only three parameters intuitively. The texture mapping-based method can be influenced by some restrictions of the graphic library. For example, in OpenGL, the depth ranges from the camera, or the viewer, to the plane P needs to be given as near and far z values to the system in advance. Besides, each of the projective texture mapping and the rendering needs the 3D computation using a 4×4 matrix. Consequently, we use the homography-based method using a background plane, which has no restriction and enables the efficient 2D computation using the 3×3 matrix H .

3.3.3 Viewer composite image

In Step 1-3, a viewer composite image is obtained by overlaying the viewer virtual object image on the viewer background image. The virtual object can be made semitransparent by alpha blending so that the background can be seen through the virtual object.

3.4 Generation of the projection image

In Step 2, a projection image has been generated from the viewer composite image by using the two-pass algorithm [5][35][36]. A composite is an image made from the combination of a variety of pictures (two or more). After the perspective rendering of the desired 3D model, using layers, combined images of the 3D models that are being called projection images are now complete. In the same way as the texture mapping-based method, first, the viewer composite image is used as a texture and *projectively mapped* on the surface of the geometric model of the real object from the viewer's

position. Then, the textured surface is rendered again from the projector's position to generate a view-dependent projection image. The resulting projection image is "correctly distorted" so that the virtual object and the background in the original viewer composite image can be seen with their "correct appearances" by the viewer in Step3.

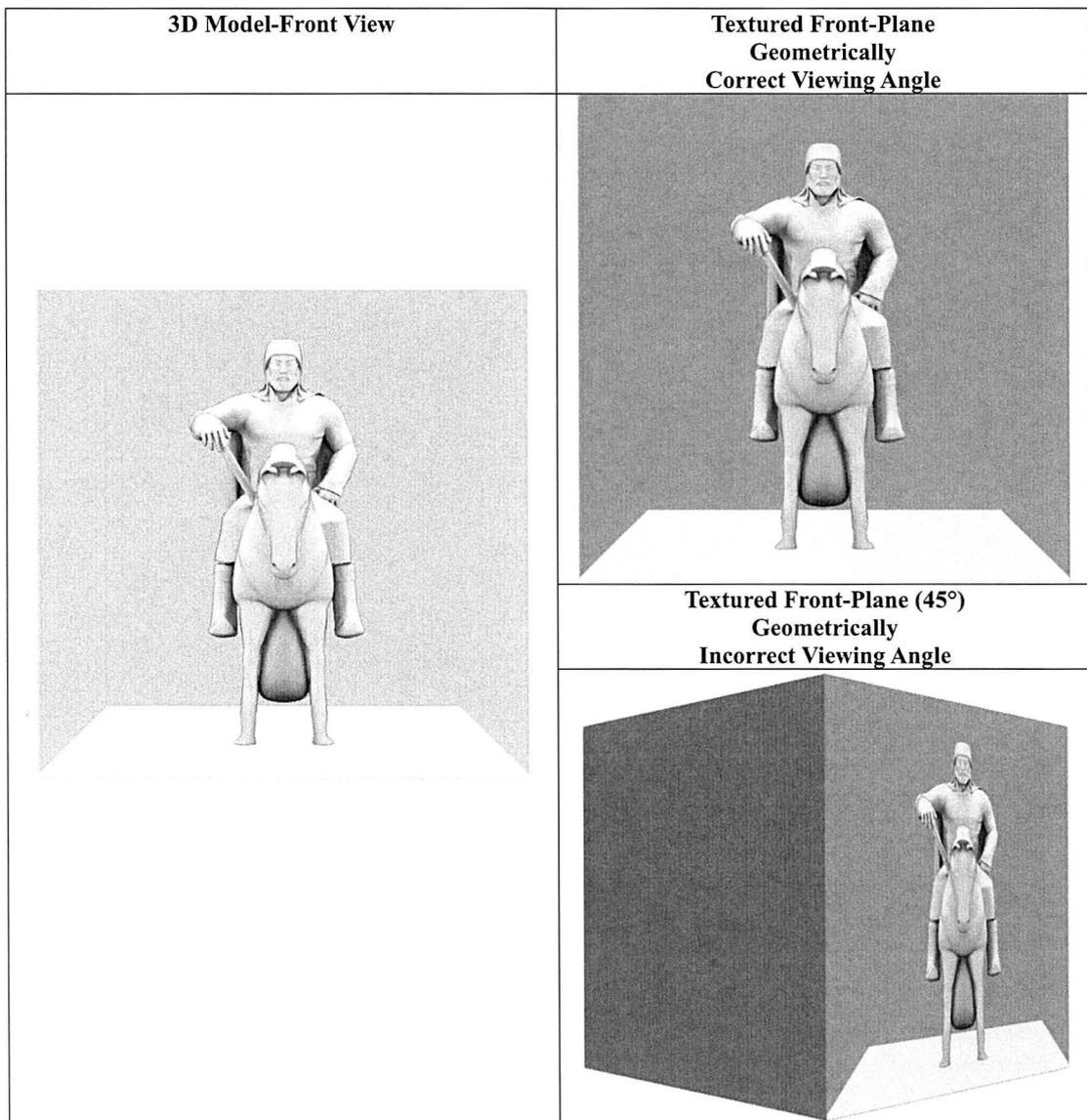


Figure 3. 3D Model versus Projection Image.

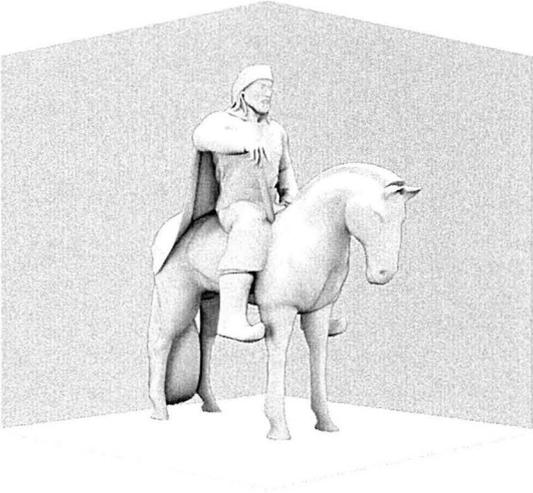
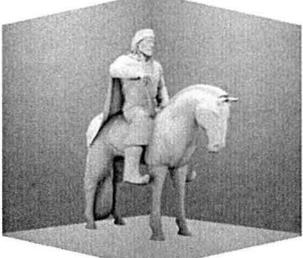
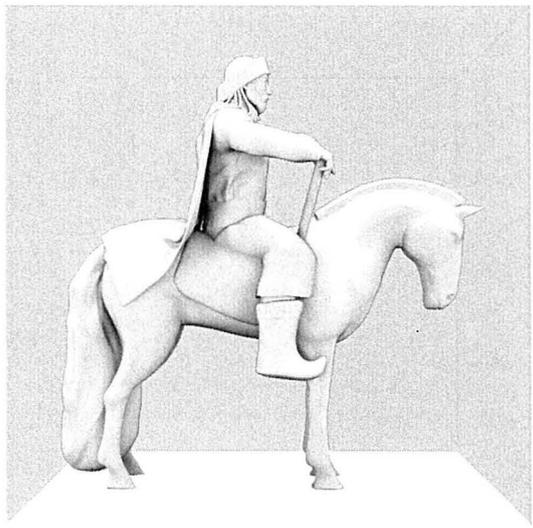
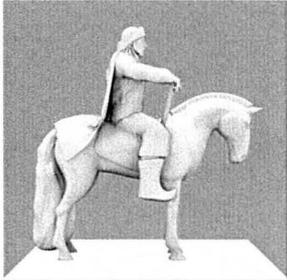
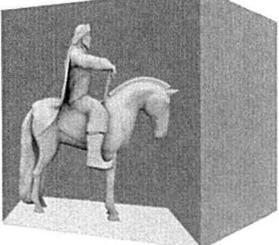
<p>3D Model (45°)</p>	<p>Textured 45°-Plane Geometrically Correct Viewing Angle</p>
	
	<p>Textured 25°-Plane Geometrically Incorrect Viewing Angle</p>
<p>3D Model (90°)</p>	<p>Textured Right-Plane Geometrically Correct Viewing Angle</p>
	
	<p>Textured 60°-Plane Geometrically Incorrect Viewing Angle</p>
	

Figure 3. Continued.

3.5 Projection of projection image

The projection of the projection image in Step3 is made in the real space while the generation of the viewer composite image and the projection image in Steps 1 and 2 is done in the virtual space. In Step 3, the projection image obtained in Step 2 is projected on the surface of the real object by the projector. The viewer sees the virtual object and the background correctly.

3.6 Adjustment of background parameters

The background parameters to define a background plane need to be adjusted as easily as possible to approximate a real background's shape optimally. We present a practical adjustment method as follows.

3.6.1 Practical background parameters

The three background parameters D_P , θ_P , and φ_P to define a background plane P to approximate a background's shape are presented in Section 3.3.2 (1). The parameter D_P is the distance from the origin O to the plane P . The parameters θ_P and φ_P define the unit normal vector N of the plane P . The real object is put on the origin O , and a viewer sees it. In a practical use, the plane P needs to be adjusted to obtain an optimal viewer background image by changing the parameters and observing the projected image on the real object's surface. This adjustment is not done efficiently if the parameters D_P , θ_P , and φ_P are used directly. In the following, the "viewer" means a person to prepare a projection mapping event and adjust the plane P in advance. In Figure 4 (a), the viewer adjusts the plane P by seeing the real object and the real background toward the origin O from the viewer's eye position Q_v ; the viewer's viewing direction is represented by the long red arrow. In the following, we mean "viewer's eye position" by "viewer's position". The plane P_1 and P_2 have the unit normal vectors N_1 and N_2 defined by the parameter sets $[\theta_{P_1}, \varphi_{P_1}]$ and $[\theta_{P_2}, \varphi_{P_2}]$. The two planes have the same distance D_P from the origin O in the directions of their normal vectors. However, the

distances D_1 and D_2 in the viewer's viewing direction are different. That is, when the viewer changes θ_P and φ_P to adjust the normal vector, the distance in the viewing direction also changes against the viewer's intention. This causes a serious difficulty for the viewer to obtain an optimal plane P .

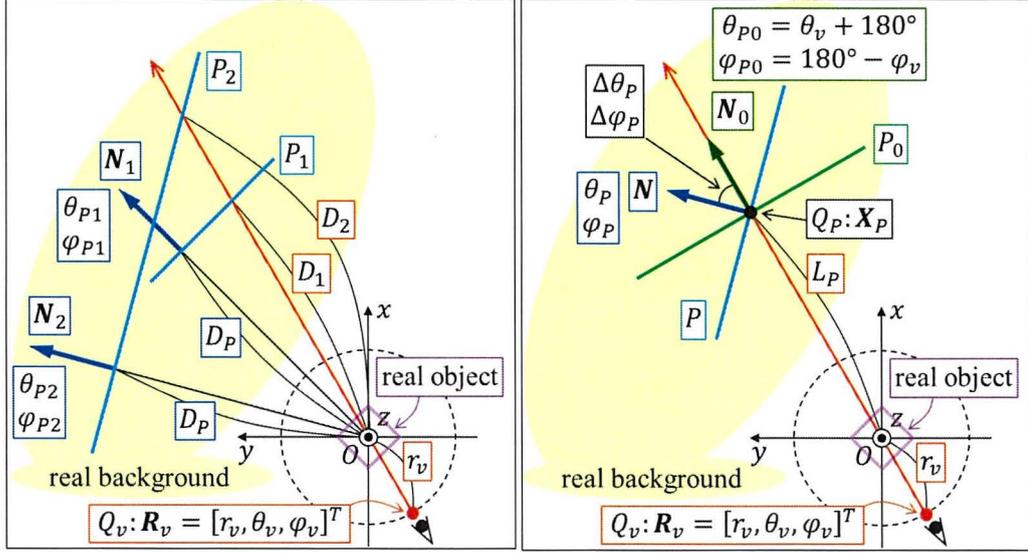


Figure 4. Background parameters

Therefore, we present another way that depends on the viewer's position for practical usefulness, as shown in Figure 4 (b). First, the base normal vector $\mathbf{N}_0 = [n_{x0}, n_{y0}, n_{z0}]^T$, $|\mathbf{N}_0| = 1$, is defined to have the same direction as the viewer's viewing direction. If the viewer's position Q_v has polar coordinates $\mathbf{R}_v = [r_v, \theta_v, \varphi_v]^T$, the vector \mathbf{N}_0 is determined as follows:

$$\theta_{P0} = \theta_v + 180^\circ, \quad (22)$$

$$\varphi_{P0} = 180^\circ - \varphi_v, \quad (23)$$

$$n_{x0} = \cos \theta_{P0} \sin \varphi_{P0}, \quad (24)$$

$$n_{y0} = \sin \theta_{P0} \sin \varphi_{P0}, \quad (25)$$

$$n_{z0} = \cos \varphi_{P0}. \quad (26)$$

Next, the base plane P_0 is defined by giving a distance L_P such that it has the normal vector \mathbf{N}_0 and the distance L_P from the origin O . An arbitrary unit normal vector \mathbf{N} defined by parameters θ_P , φ_P , and Equations (19), (20), and (21) is obtained based on the vector \mathbf{N}_0 using difference parameters

$\Delta\theta_P$ and $\Delta\varphi_P$ by

$$\theta_P = \theta_{P_0} + \Delta\theta_P = \theta_v + 180^\circ + \Delta\theta_P, \quad (27)$$

$$\varphi_P = \varphi_{P_0} + \Delta\varphi_P = 180^\circ - \varphi_v + \Delta\varphi_P. \quad (28)$$

Then, the viewer can obtain an objective plane P based on the plane P_0 by changing the three parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$, as shown in Figure 4 (b). The base point Q_P is defined as the intersection of the plane P_0 and the viewing direction and has the world coordinates $\mathbf{X}_P = L_P \mathbf{N}_0$. The point Q_P is positioned on an arbitrary plane P defined by L_P , $\Delta\theta_P$, and $\Delta\varphi_P$. This means that the coordinates \mathbf{X}_P satisfy Equation (10), which gives

$$d = -\mathbf{N}^T \mathbf{X}_P = -L_P \mathbf{N}^T \mathbf{N}_0. \quad (29)$$

Equations (18) and (29) give the relation between D_P and L_P below:

$$D_P = L_P \mathbf{N}^T \mathbf{N}_0. \quad (30)$$

The two parameter sets $[D_P, \theta_P, \varphi_P]$ and $[L_P, \Delta\theta_P, \Delta\varphi_P]$ are related by Equations (27), (28), and (30). The set $[L_P, \Delta\theta_P, \Delta\varphi_P]$ enables the viewer to adjust the plane P more easily than $[D_P, \theta_P, \varphi_P]$. Therefore, we use the parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$ as background parameters practically. The parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$ determine an arbitrary plane P for a viewer's position Q_v with coordinates θ_v and φ_v . The unit normal vector \mathbf{N} is determined by Equations (19), (20), (21), (27), and (28). The constant d is determined by Equation (29), in which the vector \mathbf{N}_0 is determined by Equations (22), (23), (24), (25), and (26). The plane P always has the base point Q_P , which is seen by the viewer at the center of his/her visual field. The point Q_P is away from the viewer by the distance $L_P + r_v$. For fixed values of $\Delta\theta_P$ and $\Delta\varphi_P$, the change of the distance L_P moves the plane P forward and backward in the viewing direction with the normal vector \mathbf{N} unchanged. For a fixed value of L_P , the change of the angles $\Delta\theta_P$ and $\Delta\varphi_P$ causes the 3D rotation of the plane P around the point Q_P , which is fixed as a rotational center, by moving the normal vector \mathbf{N} apart from the base normal vector \mathbf{N}_0 . The above way enables the viewer to obtain an optimal plane P easily, intuitively, and efficiently.

3.6.2 Interpolation of background parameters

An optimal background view projected on the real object's surface by using a background plane P should be as similar as possible to the actual background view seen by a viewer. The optimal plane P for such a projected background view depends on the viewer's position. Our method provides optimal background parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$ according to the viewer's position. They are separated as follows:

$$L_P = L_P^G + L_P^L, \quad (31)$$

$$\Delta\theta_P = \Delta\theta_P^G + \Delta\theta_P^L, \quad (32)$$

$$\Delta\varphi_P = \Delta\varphi_P^G + \Delta\varphi_P^L. \quad (33)$$

The *global parameters* L_P^G , $\Delta\theta_P^G$, and $\Delta\varphi_P^G$ are common for every viewer's position to adjust the plane P roughly. The *local parameters* L_P^L , $\Delta\theta_P^L$, and $\Delta\varphi_P^L$ change according to the viewer's position to adjust the plane P in detail to optimize the projected background view. For an arbitrary viewer's position with coordinates $\mathbf{R}_v = [r_v, \theta_v, \varphi_v]^T$, the local parameters are given by functions of \mathbf{R}_v as follows:

$$L_P^L = F_{L_P^L}(\mathbf{R}_v) = F_{L_P^L}(r_v, \theta_v, \varphi_v), \quad (34)$$

$$\Delta\theta_P^L = F_{\Delta\theta_P^L}(\mathbf{R}_v) = F_{\Delta\theta_P^L}(r_v, \theta_v, \varphi_v), \quad (35)$$

$$\Delta\varphi_P^L = F_{\Delta\varphi_P^L}(\mathbf{R}_v) = F_{\Delta\varphi_P^L}(r_v, \theta_v, \varphi_v). \quad (36)$$

To define the functions, for coordinates \mathbf{R}_v in the ranges

$$r_{min} \leq r_v \leq r_{max}, \quad (37)$$

$$\theta_{min} \leq \theta_v < \theta_{max}, \theta_{min} = 0, \theta_{max} = 360, \quad (38)$$

$$\varphi_{min} \leq \varphi_v \leq \varphi_{max}, \quad (39)$$

$N_r \times N_\theta \times N_\varphi$ grid points $G_{[i,j,k]}$, $0 \leq i < N_r$, $0 \leq j < N_\theta$, $0 \leq k < N_\varphi$, are given. They are positioned at $[r_i, \theta_j, \varphi_k]$ given by

$$r_i = \begin{cases} r_{min} (= r_{max}) & (N_r = 1), \\ r_{min} + \Delta r \cdot i, \quad \Delta r = \frac{r_{max} - r_{min}}{N_r - 1} & (N_r \geq 2), \end{cases} \quad (40)$$

$$\theta_j = \Delta\theta \cdot j, \quad \Delta\theta = 360/N_\theta, \quad (41)$$

$$\varphi_k = \begin{cases} \varphi_{min} (= \varphi_{max}) & (N_\varphi = 1), \\ \varphi_{min} + \Delta\varphi \cdot k, \quad \Delta\varphi = \frac{\varphi_{max} - \varphi_{min}}{N_\varphi - 1} & (N_\varphi \geq 2). \end{cases} \quad (42)$$

Each grid point is given local parameters $L_P^L[i, j, k]$, $\Delta\theta_P^L[i, j, k]$, and $\Delta\varphi_P^L[i, j, k]$, which are interpolated linearly to define the functions $F_{L_P^L}$, $F_{\Delta\theta_P^L}$, and $F_{\Delta\varphi_P^L}$. From Equations (27), (28), (31), (32), (33), (34), (35), and (36), the original parameters L_P , θ_P , and φ_P to define a plane P are obtained by

$$L_P = L_P^G + F_{L_P^L}(\mathbf{R}_v), \quad (43)$$

$$\theta_P = \theta_v + 180^\circ + \Delta\theta_P^G + F_{\Delta\theta_P^L}(\mathbf{R}_v), \quad (44)$$

$$\varphi_P = 180^\circ - \varphi_v + \Delta\varphi_P^G + F_{\Delta\varphi_P^L}(\mathbf{R}_v). \quad (45)$$

In order to obtain these parameters for an arbitrary viewer's position, optimal global and local parameters L_P^G , $\Delta\theta_P^G$, $\Delta\varphi_P^G$, $L_P^L[i, j, k]$, $\Delta\theta_P^L[i, j, k]$, and $\Delta\varphi_P^L[i, j, k]$ need to be given in advance.

Our current system provides an interface to give the global and local parameters by trial and error manually. In the real space, the interface enables us to stand at a position near each grid point, check a background view projected on the real object's surface, and adjust the background plane by changing the parameters interactively to obtain an optimal background view in real time. It is an important future work to develop an efficient way to determine the optimal parameters automatically or semi-automatically.

3.7 Practice

The following are necessary for the practice of our method.

3.7.1 Tracking of viewer

In real space, the position of a moving viewer needs to be known and tracked in real-time. We use a sensor to track the viewer to obtain the exact position to correlate the physical world element and the virtual world. The obtained position is being used in Steps 1 and 2 in the virtual space. The tracking

sensor is fixed near the real object and directed toward the area in which the viewer can move.

Consequently, the tracking sensor and the camera has been placed directly near the physical object, and they are in a fixed position directed towards the sides opposite to each other. Tracking the viewer is the key to achieving geometrically correct looking view-dependent projection mapping. If there is an error on the way, the whole projection mapping, including the background view-dependent projection, will be disturbed. In particular, there are necessary steps that had to be achieved orderly to complete the experience. Detailed examination of the full capability of the sensors that are being used in the study the experiment's accuracy was elevated to the point that it almost matched with the real-life background with line by line.

3.7.2 Generation of viewer virtual object image

Our system has two options to obtain a viewer virtual object image in Step 1-1. The first option renders the 3D geometric model of a virtual object in real-time. This option can generate an image for an arbitrary position of a moving viewer with less memory than the second one. Besides, a desired rendering and shading algorithm can be used. However, the algorithm needs to be implemented to work efficiently in real-time in our system, and the quality of the image is restricted by the real-time processing time. The second option renders the model in advance for predetermined discrete viewer's positions and saves the rendered images to a storage device such as a hard disk; the discrete positions are defined in the range of Equations (37), (38), and (39) in the same way as the grid points. In a viewer's experience, all the saved images are stored on a memory, and the image for the discrete position nearest to the actual position of a moving viewer is used. To achieve a fast real-time effect on the viewers viewing experience by displaying unloading smooth view-dependent projection mapping as a result. It was a necessity to pre-render the 3D object for a better viewing experience.

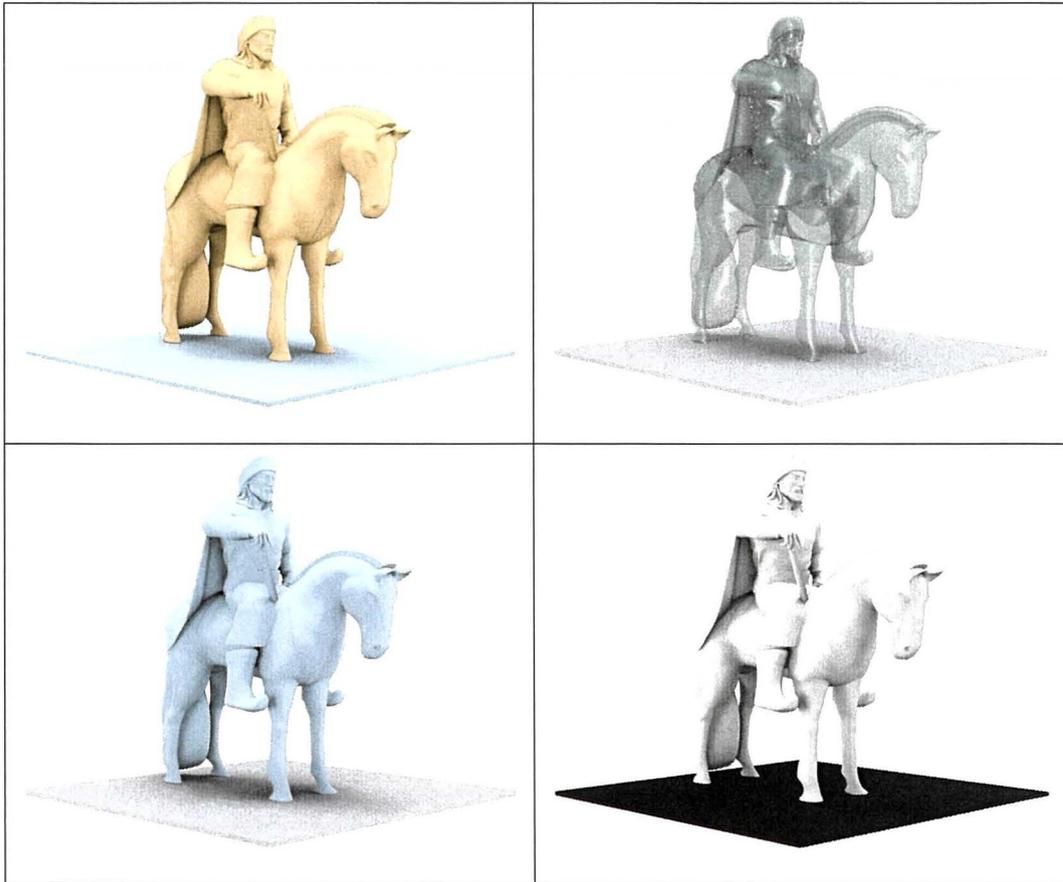


Figure 5. 3D Models rendered with various materials

We used a variety of 3D Models as an object to be displayed, but we settled on the current 3D Model of Genghis Khan (Chingis Khaan) of the ruler of Mongol Empire which is a real-life statue located in the capital city of Mongolia. This option can use high-quality images generated by time-consuming advanced rendering and shading algorithms and graphic tools. Furthermore, the rendering process could include not different three-dimensional objects, also different materials, environment, lighting etc. If desired, it is possible to render an animated object to be displayed and, in a view-dependent manner too. When it comes to the virtual side of the projection mapping, there are no limits that could hold back what the presenter wants to present. However, it needs large storage/memory; in particular, an amination of a virtual object needs huge storage/memory to treat all frame images for all discrete positions.

CHAPTER 4

EXPERIMENT

4.1 Experiment in virtual environment

We firstly made an experiment in a virtual environment constructed by OpenGL. It is shown in Figure 6. The top view (a) shows the horizontal xy plane, and the side view (b) shows the xz plane. A cube is a real object, which is fixed at the origin O ; it is denoted by the violet square. A viewer sees it at a position Q_v with coordinates $\mathbf{R}_v = [r_v, \theta_v, \varphi_v]^T$. The three small round red dots denote typical viewer's positions and the long red arrows denote their viewing directions; in each of (a) and (b), the middle dot denotes the viewer's initial position Q_v^{ini} with $\mathbf{R}_v^{ini} = [r_v^{ini}, \theta_v^{ini}, \varphi_v^{ini}]^T$. A projector is fixed at a position Q_{pr} with $\mathbf{R}_{pr} = [r_{pr}, \theta_{pr}, \varphi_{pr}]^T$; it is directed toward the cube. A camera is fixed at a position Q_c with $\mathbf{R}_c = [r_c, \theta_c, \varphi_c]^T$; it is directed toward the side opposite to the cube. The green and blue dots denote the positions Q_{pr} and Q_c , and the long green and blue arrows denote their viewing directions. The viewer, the projector, and the camera are given the values in Table 1. The height of the floor is $z = -1.2$. In this experiment, the viewer's coordinate φ_v moves from 80 to 100 degrees; the coordinate z relative to the floor's height moves roughly from 1.7 to 0.7 meter for $r_v = 3.0$ meters. This range of z simulates the height of an actual viewer's eyes. The size of the cube is 0.4^3 meters. The cube's top and bottom faces are parallel to the floor, while its four vertical edges are put toward

$\pm x$ and $\pm y$ directions. The angles of view $[\alpha_x, \alpha_y]$ and the numbers of pixels $[M_x, M_y]$ of the projector and the camera are the same as those of our actual projector and camera used in the experiments in Section 4.2, while those of the viewer were selected to appropriately evaluate the performance of our method.

The result of the experiment is shown in Figure 7. The images are “augmented” views seen from twenty-five viewer’s positions given by $\theta_v = 160, 170, 180, 190, 200$ and $\varphi_v = 80, 85, 90, 95, 100$ at a constant distance $r_v = r_v^{ini} = 3.0$. An orange teapot is a virtual object. The colored boxes simulate objects in a real background. The boxes are arranged close to two planes vertical to the xy plane; the brown lines in Figure 6 © represent the planes. The two planes intersect perpendicularly, and they are 10 meters away from the origin O . The boxes are rotated randomly. Their centers are arranged regularly with random displacements within $\pm 10\%$ of the interval between neighboring boxes horizontally and vertically on the planes and within ± 0.1 meter perpendicularly to the planes. The size of each box is 0.55^3 meters. We call the planes “box planes”. The distribution of the boxes close to the two box planes needs to be approximated by a single background plane P . Each image in Figure 7 has two areas. The dark central area is the cube’s surface onto which the teapot and the background are projected by the projector; the cube’s top face is bright, and its bottom face is black due to the position of the projector. The remaining area is the background directly observed. Our method aims to remove the presence of the cube by projecting the background on its surface. Thus, the background plane P should be adjusted such that the projected background matches the directly-observed background along the boundary of the cube’s contour as seamlessly as possible. The background parameter interpolation provides the “boundary match” for an arbitrary viewer’s position.

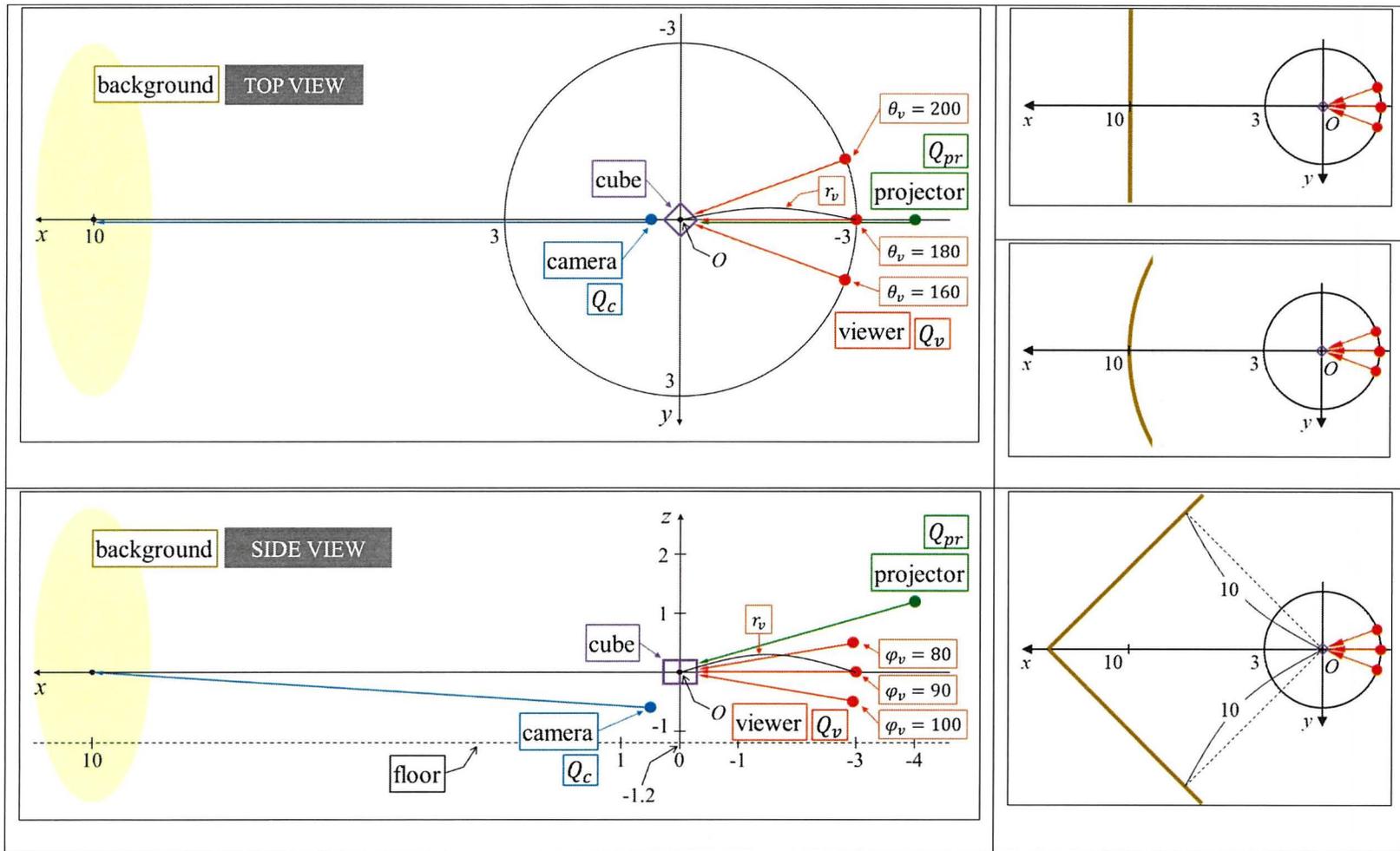


Figure 6. Virtual Environment

Table 1. Values for experiments in virtual environment.

	Position [r, θ, φ] (r : meters) (θ, φ : degrees)	Position [x, y, z] (meters)	Gaze point [x, y, z] (meters)	Angles of view [α_x, α_y] (degrees)	Numbers of pixels [M_x, M_y]
Viewer	3.0, 180.0, 90.0	-3.0, 0.0, 0.0 -3.0, 0.0, 1.2	0.0, 0.0, 0.0 0.0, 0.0, 1.2	20.0, 15.0	640, 480
Projector	4.2, 180.0, 74.0	-4.0, 0.0, 1.2 -4.0, 0.0, 2.4	0.0, 0.0, 0.0 0.0, 0.0, 1.2	38.0, 23.0	640, 400
Camera	0.8, 0.0, 140.0	0.5, 0.0, -0.6 0.5, 0.0, 0.6	10.0, 0.0, 0.0 10.0, 0.0, 1.2	57.0, 43.0	640, 480

4.1.1 View-dependent projection mapping

The result of the first experiment “Exp.1” is shown in Figure 7. This experiment shows the fundamental ability of our view-dependent projection mapping using the two-pass algorithm, that is, how the correct appearance of a virtual object is shown to a viewer moving around a real object. In this experiment, a teapot colored in orange is used as a virtual object, and a background is not treated. The viewer sees the real object, that is, the cube from nine positions given by $\theta_v = 160, 180, 200$ and $\varphi_v = 80, 90, 100$ at a constant distance of $r_v = r_v^{ini} = 3.0$ from the cube. The images in (a) are rendered images, that is, correct views seen from the respective positions. The images in (b) and (c) are projection images obtained for the projector’s position Q_{pr} by the two-pass algorithm. While the images in (b) are obtained by the angles of view of the projector in Table 1, the images in (c) are obtained by halving the angles to magnify the area of the teapot and show the detail of the “correctly distorted” projection image. The images in (d) are simulated real views seen from the viewer; the images are obtained by simulating the projection of the projection images in (b) onto the cube’s surface. The comparison between the images in (a) and (d) shows that the correct appearances of the teapot are shown according to the viewer’s positions.

4.1.2 Adjustment of projected real background

The result of the second experiment “Exp.2” is shown in Figure 6. This experiment shows

how a viewer background image is adjusted by background parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$. The colored boxes simulate objects in a real background. They are arranged close to the vertical plane which is parallel to the yz plane and 10 meters away from the origin O in the x direction; the plane is represented by the brown line in Figure 6 (c). The boxes are rotated randomly; their centers are arranged regularly with random displacements within $\pm 10\%$ of the interval between neighboring boxes horizontally and vertically on the plane and within ± 0.1 meter perpendicularly to the plane. The length of one side of each box is 0.55 meter. In Figure 6, the image (a) is a camera background image captured from the camera's position Q_c . We mean a real view directly seen from a viewer without image conversion and projection by "directly-observed" view. The image (b) is a directly-observed background view seen from the viewer's initial position Q_v^{ini} with $R_v^{ini} = [3.0, 180, 90]$. The images in (c1), (d1), and (e1) are viewer background images for the viewer's position Q_v^{ini} ; the images are obtained from the image (a) by the background planes P defined by different sets of parameters $L_P = 5, 10, 15$, $\Delta\theta_P = -45, 0, +45$, and $\Delta\varphi_P = -45, 0, +45$. Each of the bottom-left and bottom-right images in (c1) has a black area at one top corner; this area is out of the range in which the image (a) is converted on the image in (c1). We mean a real view containing a real object with projection and the directly-observed background behind it by "augmented" view. The images in (c2), (d2), and (e2) are augmented views seen from the viewer's position Q_v^{ini} ; in each image, a dark central area is the cube's surface onto which the viewer background image in (c1), (d1), or (e1) is projected from the projector's position Q_{pr} by using the two-pass algorithm while the remaining area behind the cube is the directly-observed background in (b).

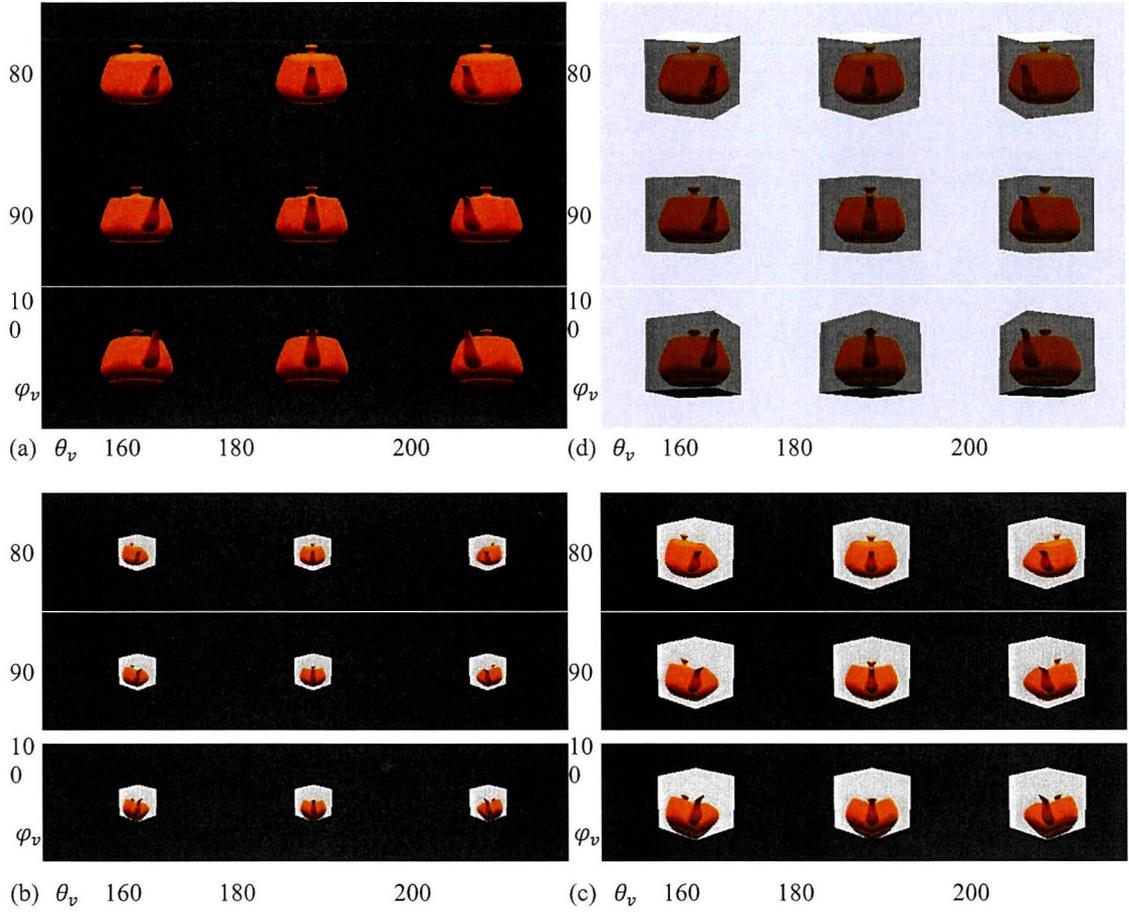


Figure 7. View-dependent projection of a virtual object for different viewer's positions in Exp.1. $r_v = r_v^{ini} = 3.0$.

The images in Figure 8 show the practical usefulness of the parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$. The change of L_P moves the plane P forward/backward in the viewer's viewing direction, which magnifies/demagnifies the viewer background image and adjusts the sizes of all the boxes uniformly. The change of $\Delta\theta_P$ and $\Delta\varphi_P$ changes the incline of the plane P , which distorts the viewer background image and adjusts the sizes of the boxes relatively according to their positions. The main purpose of our method is to make a viewer feel a virtual object merging into the real world by making the viewer not feel the presence of a real object by the background projection. For this purpose, the most important evaluation to the background approximation by the plane P is to evaluate how seamlessly the projected

background on the cube’s surface matches the directly-observed background behind the cube along the boundary of the cube’s contour. The central image in (d2), which is obtained by $L_P = 10$, $\Delta\theta_P = 0$, and $\Delta\varphi_P = 0$, has the best seamless match between the projected background and the directly-observed background along the boundary; in this case, the plane P is the same as the plane which the boxes are arranged close to. The result of this experiment shows that our method can adjust the “boundary match” between a projected background and a directly-observed background easily, intuitively, and efficiently. This ability is important to achieve the merge of a virtual object into the real world.

4.1.3 Projection of virtual object and real background

The result of the third experiment “Exp.3” is shown in Figure 9. This experiment shows how our method achieves the merge of a virtual object into the real world by the view-dependent projection of not only the virtual object but also the real background. The images in Figure 9 are augmented views seen from twenty-five viewer’s positions given by $\theta_v = 160, 170, 180, 190, 200$ and $\varphi_v = 80, 85, 90, 95, 100$ at a constant distance $r_v = r_v^{ini} = 3.0$. Each image has a dark central area of the cube’s surface with the projection of the teapot and the background and the remaining area of the directly-observed background. The projector at the position Q_{pr} cannot project a projection image on the bottom face of the cube, which makes the bottom face black in each image of $\varphi_v = 100$. Conversely, the top face of the cube is bright in each image of $\varphi_v = 80$. Background boxes are arranged close to the surface of a sphere, which is represented by the brown curve in Figure 8 (d). The sphere has its center at the origin O and a radius of 10 meters. The boxes are given the same random rotation and displacement as those of the boxes used in Exp.2. Then, if the distance L_P of the background plane P is the same as the sphere’s radius and the plane’s normal vector N is the same as its base normal vector N_0 , the resulting augmented view for every viewer’s position has almost the best boundary match between the projected background on the cube’s surface and the directly-

observed background behind it. Thus, we select $L_P = 10$, $\Delta\theta_P = 0$, and $\Delta\varphi_P = 0$. Each image in Figure 9 shows the above result although there are some slight gaps between the projected background and the directly-observed background along the cube’s contour. The gaps are caused by the difference of the shapes of the sphere and the plane P as well as the approximation of the 3D shapes of the boxes by the plane P .

The brown checkered plane in the images of Figure 9 is the floor. The boundary match between the projected floor on the cube’s surface and the directly-observed floor is not appropriate. In particular, they do not match at all in the area near the cube, as shown in the images for $\varphi_v = 80$. This is caused by the reason that the floor is far from the plane P .

4.1.4 Interpolation of background parameters

The fourth experiment shows the effectiveness of the interpolation of background parameters L_P , $\Delta\theta_P$, and $\Delta\varphi_P$. This experiment consists of two sub-experiments “Exp.4-1” and “Exp.4-2”. The results of these experiments are shown in Figures 10 and 11. The images in each figure are augmented views seen from the same twenty-five viewer’s positions as those used in Exp.3. In Exp.4-1 and 4-2, grid points $G_{[l,j,k]}$ with coordinates $[r_l, \theta_j, \varphi_k]$ given by Equations (40), (41), and (42) are defined by Table 2. Background boxes are arranged close to one vertical plane in Exp.4-1 and two vertical planes in Exp.4-2, as represented by the brown lines in Figure 8 (c) and (e) respectively. The plane in (c) was also used in Exp.2. The two planes in (e) intersect perpendicularly, and the line of intersection of the planes is parallel to the z axis and intersects with the x axis. Both planes are 10 meters away from the origin O . The boxes are given the same random rotation and displacement as those of the boxes used in Exp.2 and 3. In the following, we mean the above planes for the boxes by “box plane”. The background parameter interpolation provides an appropriate boundary match between a projected background and a directly-observed background for an arbitrary viewer’s position. In each of Exp.4-1 and 4-2, the best result obtained by the interpolation is compared with the best result without the

interpolation. The global parameters $L_P^G, \Delta\theta_P^G, \Delta\varphi_P^G$ and the local parameters $L_P^L, \Delta\theta_P^L, \Delta\varphi_P^L$ are shown in Tables 3, 4, and 5. When the interpolation is not used, only global parameters are used and local parameters are not used. In this case, the global parameters were selected to provide the best result for the viewer's initial position Q_v^{ini} .

In Exp.4-1, the images in Figure 10 (a) are obtained when the interpolation is not used. The global parameters in Table 3 make the background plane P become the same as the box plane for the viewer's initial position Q_v^{ini} . Therefore, the image for $R_v^{ini} = [3.0, 180, 90]$ in the center of Figure 10 (a) has almost a perfect boundary match between the projected background and the directly-observed background. However, the same parameters are also used for other viewer's positions without the interpolation. Then, as the viewer goes away from Q_v^{ini} , the plane P moves away from the box plane. This makes the boundary match worse. Particularly, in the images for $[\theta_v, \varphi_v] = [160, 80], [160, 100], [200, 80],$ and $[200, 100]$ in the four corners of Figure 10 (a), there are noticeable gaps between the projected background and the directly-observed background along the cube's contour.

The images in Figure 10 (b) are obtained when the interpolation is used. In addition to using the global parameters in Table 3, the local parameters in Table 4 are given to the grid points to define the linear interpolation functions $F_{L_P^L}, F_{\Delta\theta_P^L},$ and $F_{\Delta\varphi_P^L}$ in Equations (34), (35), and (36). The local parameters were selected to make the plane P become the same as the box plane for the viewer's positions at the respective grid points. In Figure 10 (b), the nine images with black frames are obtained directly from the local parameters given to the grid points while the other images are obtained from local parameters given by the interpolation functions. The nine images have almost perfect boundary matches between the projected background and the directly-observed background although there are some slight gaps caused by the approximation of the 3D shapes of the boxes by the plane P . The other images also have appropriate boundary matches due to the interpolation although their qualities are slightly lower than the qualities of the above nine images. The comparison between the images in (a) and (b) of Figure 10 shows that the background parameter interpolation works quite well to obtain an

appropriate boundary match for an arbitrary viewer's position.

In Exp.4-2, the distribution of the boxes close to the two box planes needs to be approximated by a single background plane P . The images in Figure 11 (a) are obtained when the interpolation is not used. The global parameters in Table 3 were carefully selected such that the projected background was matched with the directly-observed background as appropriately as possible along the cube's contour for the viewer's initial position Q_v^{ini} . The resulting plane P is parallel to the yz plane and 12.8 meters away from the origin O . The image for $R_v^{ini} = [3.0, 180, 90]$ in the center of Figure 11 (a) has an appropriate boundary match although there are some noticeable gaps caused by the approximation of the two box planes by the single plane P as well as the approximation of the 3D shapes of the boxes by the plane P . However, the other images have worse boundary matches with serious gaps due to the same parameters selected only for Q_v^{ini} . The boundary matches in Figure 11 (a) are much worse than those in Figure 10 (a) because of the difficulty of the approximation by the single plane P .

The images in Figure 11 (b) are obtained when the interpolation is used by the global parameters in Table 3 and the local ones in Table 5. The grid point with coordinates $[r_i, \theta_j, \varphi_k] = [3.0, 180, 90]$, which are the same as R_v^{ini} , is given the same parameters in total as those given in the "without-interpolation" case, that is, $L_P = L_P^G + L_P^L = 12.8$, $\Delta\theta_P = \Delta\theta_P^G + \Delta\theta_P^L = 0$, $\Delta\varphi_P = \Delta\varphi_P^G + \Delta\varphi_P^L = 0$. Consequently, the central images in (a) and (b) of Figure 11 are the same by the same plane P . The two grid points with $[r_i, \theta_j, \varphi_k] = [3.0, 180, 80]$ and $[3.0, 180, 100]$ are given the local parameters to make the same plane P remain. This means that the same plane P remains for a viewer's position with $r_v = 3.0$, $\theta_v = 180$, and $80 \leq \varphi_v \leq 100$ by the interpolation. The three grid points with $\theta_j = 160$ in Table 5 are given the local parameters to make the plane P become the same as the right box plane in $y \leq 0$ seen from the origin O . This is reasonable because a viewer at the position with $\theta_v = 160$ sees mainly the boxes close to the right plane. In the same way, the local parameters given to the three grid points with $\theta_j = 200$ make the plane P become the same as the left box plane in $y \geq 0$. In Figure 11 (b), the nine images with black frames are obtained directly from the local parameters

given to the grid points. These images have appropriate boundary matches although there are some gaps. Among them, the three images for $\theta_v = 180$ have more noticeable gaps than the others have. The projected background in each of the three images has the boxes, some of which are close to the right box plane and others of which are close to the left one. The noticeable gaps are caused by the imperfectness of the approximation for the two box planes by the single plane P . The projected background in each of the remaining six images for $\theta_v = 160$ and 200 has the boxes, all of which are close to only one of the two box planes. The single plane P works quite well to approximate the distribution of the boxes. On the other hand, the images other than the above nine images are obtained from local parameters given by the interpolation functions. These images also have appropriate boundary matches due to the interpolation although there are also some gaps. Among them, the two images for $\theta_v = 180$ have some noticeable gaps and the four images for $\theta_v = 160$ and 200 have less gaps by the same reason as that for the images with black frames. The ten images for $\theta_v = 170$ and 190 have noticeable gaps although the projected background in each image has the boxes, all of which are close to only one of the two box planes; this situation of the boxes is the same as that for $\theta_v = 160$ and 200 . The background parameters for $\theta_v = 170$ are obtained by interpolating the parameters given to the grid points for $\theta_v = \theta_j = 160$ and 180 . This means that the background plane P_{170} is the intermediate between the two planes P_{160} and P_{180} , where P_{θ_v} means a plane for θ_v . Thus, the noticeable gaps for $\theta_v = 170$ are caused by the plane P_{170} that does not coincide with the box plane. The same applies to the case of $\theta_v = 190$. Compared to Figure 11 (a), the boundary matches are greatly improved in Figure 11 (b) by the background parameter interpolation although there are some gaps, part of which are noticeable.

The result of the above experiments shows that the background parameter interpolation has the fundamental ability to make the boundary match between a projected background and a directly observed background as appropriate as possible for an arbitrary viewer's position. This ability achieves the effective view-dependent projection to make the viewer not to feel the presence of a real object but

feel a virtual object merging into the real world.

4.2 Experiment in real environment

We secondly made some experiments by applying our method in a real environment. We used a white cube made from styrofoam as a real object; its size is $0.4 \times 0.4 \times 0.4$ meters. We used two Kinect sensors; one was used as a tracking sensor to track a viewer, and another was used as a camera to capture a real background. They were fixed near the cube and directed toward the sides opposite to each other. We used the 3D geometric model of a virtual object shown in Figure 10. Viewer virtual object images were obtained by the second option described in Section 3.7.2; the model was rendered in advance for $M_r \times M_\theta \times M_\phi = 1 \times 360 \times 1$ discrete viewer's positions, that is, for every one degree around the model. The demo videos were captured by moving a video camera around the cube.

The results of the three experiments are shown below. Figure 11 shows some frame images of the demo video obtained in the experiment "Exp.5". In this experiment, the background is a planar wall with three figures near the cube. The video shows that our method worked well. The correct appearance of the virtual object is always seen from the moving video camera. The appropriate boundary match between the projected background and the directly-observed background was obtained by adjusting the background plane so as to fit the wall. The virtual object looks as if it existed in the real world. It was difficult to eliminate the presence of the cube perfectly due to the difference of the brightness of the projected background and the directly-observed background. As a result, the cube looks like a transparent box containing the virtual object inside. This visual effect of the virtual object and the cube was also yielded in the experiments "Exp.6" and "Exp.7".

Figure 12 shows some frame images of the demo video obtained in the experiment "Exp.6". In this experiment, a rack containing some boxes was put as a background object in front of the wall used in Exp.5. The video also shows that our method worked well to show the correct appearance of the virtual object and the appropriate boundary match. Some gaps between the projected background and the directly-observed background were caused by the difficulty of the approximation for the non-

planar background object by the background plane. The dark parts appearing in the left and right sides of the rack in the projected background are out of the range of the camera background image. Figure 13 shows some frame images of the demo video obtained in the experiment “Exp.7”. Figure 14 shows some projection images used for the demo video. In this experiment, the background is far from the cube and the background objects have complicated shapes. The black jaggy-shaped wall is about 10 meters away from the cube. The white wall behind the black wall is about 15 meters away from the cube. Besides, there are desks and chairs in front of the black wall, black computers and monitors in front of the white wall, and speech tables. Our method worked well to treat a far complicated-shaped background in a large space. In the video, the correct appearance of the virtual object and the appropriate boundary match for the complicated background are seen even if the video camera moves within a wide-angle range around the cube. The boundary match over the wide range was achieved by the interpolation of background parameters to obtain an optimal background plane for each position of the video camera.

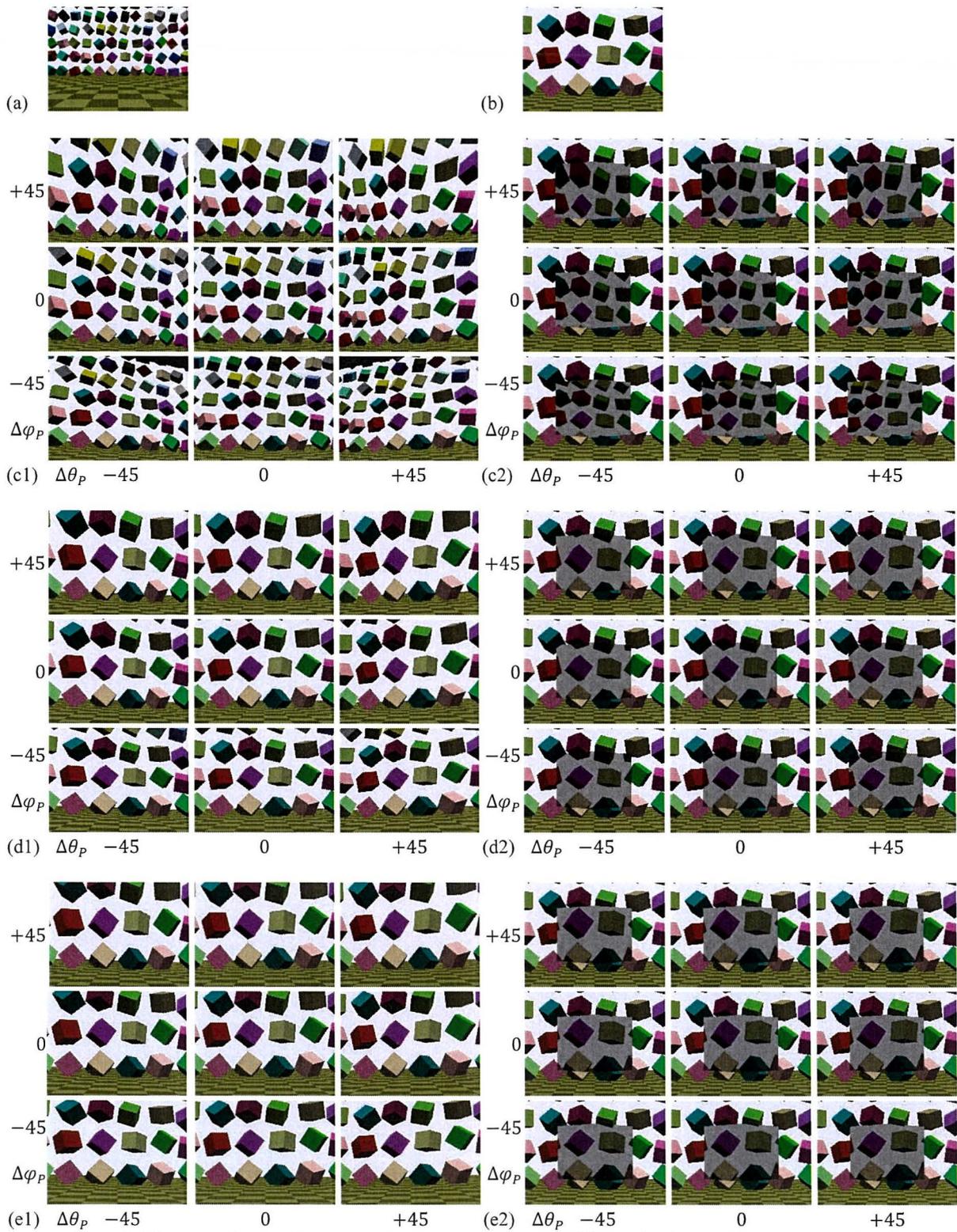


Figure 8. Viewer background images and augmented views for different background parameters in Exp.2.

(c1), (c2) $L_p = 5$. (d1), (d2) $L_p = 10$. (e1), (e2) $L_p = 15$.

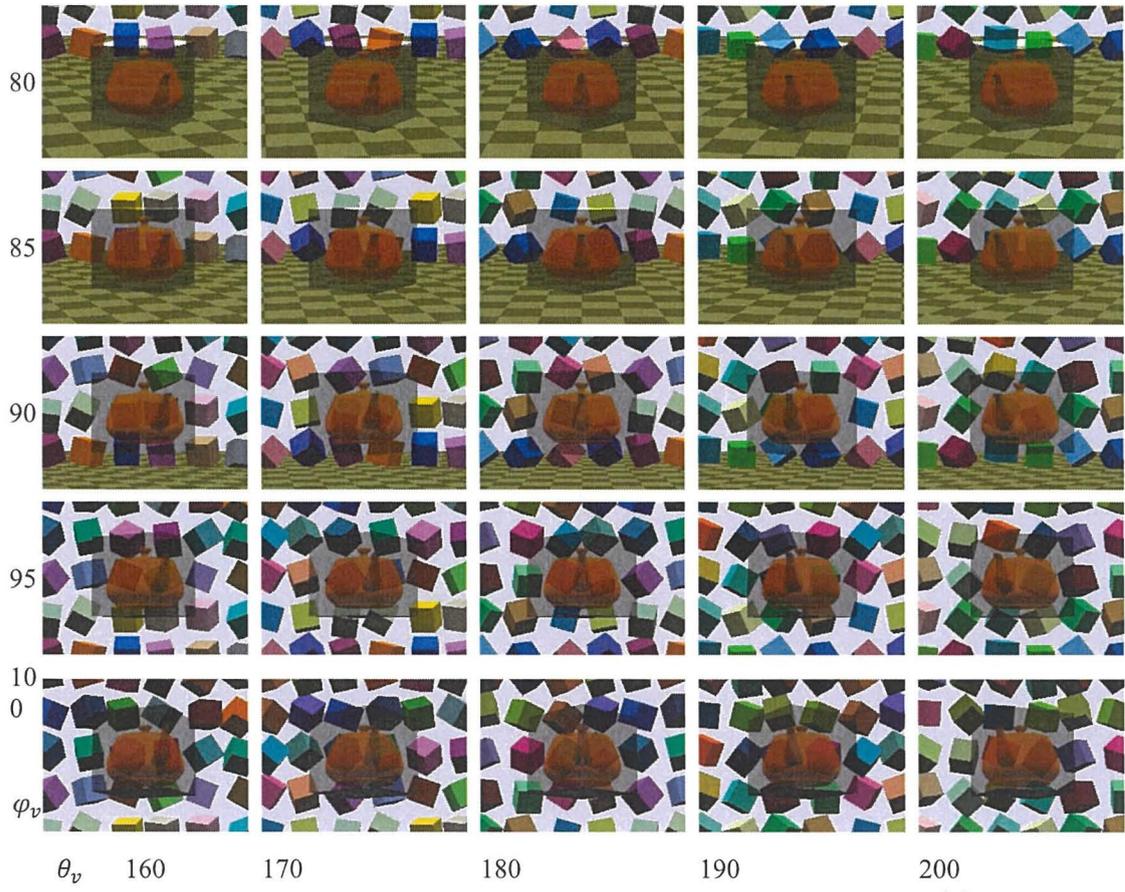


Figure 9. Augmented views for different viewer's positions in Exp.3. $r_v = r_v^{ini} = 3.0$.

Table 2. Definition of grid points in Exp.4-1 and 4-2.

	Minimum coordinate	Maximum coordinate	Number of points	Interval
r_i	$r_{min} = 2.5$	$r_{max} = 3.5$	$N_r = 3$	$\Delta r = 0.5$
θ_j	$\theta_{min} = 0$	$\theta_{max} = 360$	$N_\theta = 18$	$\Delta\theta = 20$
φ_k	$\varphi_{min} = 80$	$\varphi_{max} = 100$	$N_\varphi = 3$	$\Delta\varphi = 10$

Table 3. Global and local background parameters in Exp.4-1 and 4-2.

Exp.	Interpolation	Global parameters			Local parameters
		L_p^G	$\Delta\theta_p^G$	$\Delta\varphi_p^G$	
Exp.4-1	not used	10.0	0.0	0.0	not used
	used	10.0	0.0	0.0	Table.4
Exp.4-2	not used	12.8	0.0	0.0	not used
	used	10.0	0.0	0.0	Table.5

Table 4. Local background parameters given to grid points in Exp.4-1. Other grid points are given $[L_p^L, \Delta\theta_p^L, \Delta\varphi_p^L] = [0.0, 0.0, 0.0]$.

$r_i = 3.0$									
θ_j	160			180			200		
φ_k	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$
80	0.805944	20.0	-10.0	0.154266	0.0	-10.0	0.805944	-20.0	-10.0
90	0.641778	20.0	0.0	0.0	0.0	0.0	0.641778	-20.0	0.0
100	0.805944	20.0	10.0	0.154266	0.0	10.0	0.805944	-20.0	10.0

Table 5. Local background parameters given to grid points in Exp.4-2. Other grid points are given $[L_p^L, \Delta\theta_p^L, \Delta\varphi_p^L] = [0.0, 0.0, 0.0]$.

$r_i = 3.0$									
θ_j	160			180			200		
φ_k	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$	L_p^L	$\Delta\theta_p^L$	$\Delta\varphi_p^L$
80	1.203993	-25.0	-10.0	2.997460	0.0	-10.0	1.203993	25.0	-10.0
90	1.033779	-25.0	0.0	2.8	0.0	0.0	1.033779	25.0	0.0
100	1.203993	-25.0	10.0	2.997460	0.0	10.0	1.203993	25.0	10.0

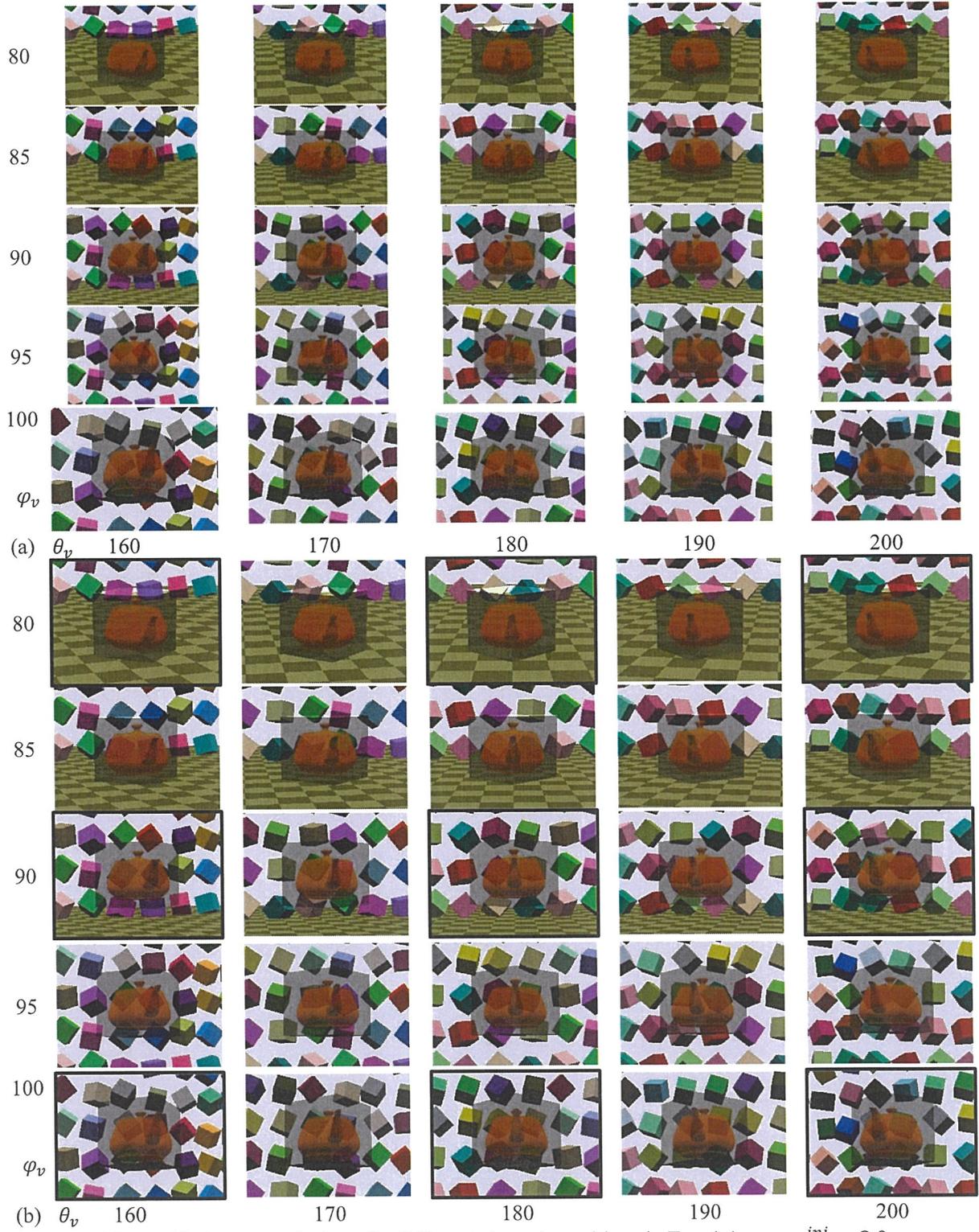


Figure 10. Augmented views for different viewer's positions in Exp.4-1. $r_v = r_v^{ini} = 3.0$.
 (a) Background parameters are not interpolated. (b) Background parameters are interpolated.

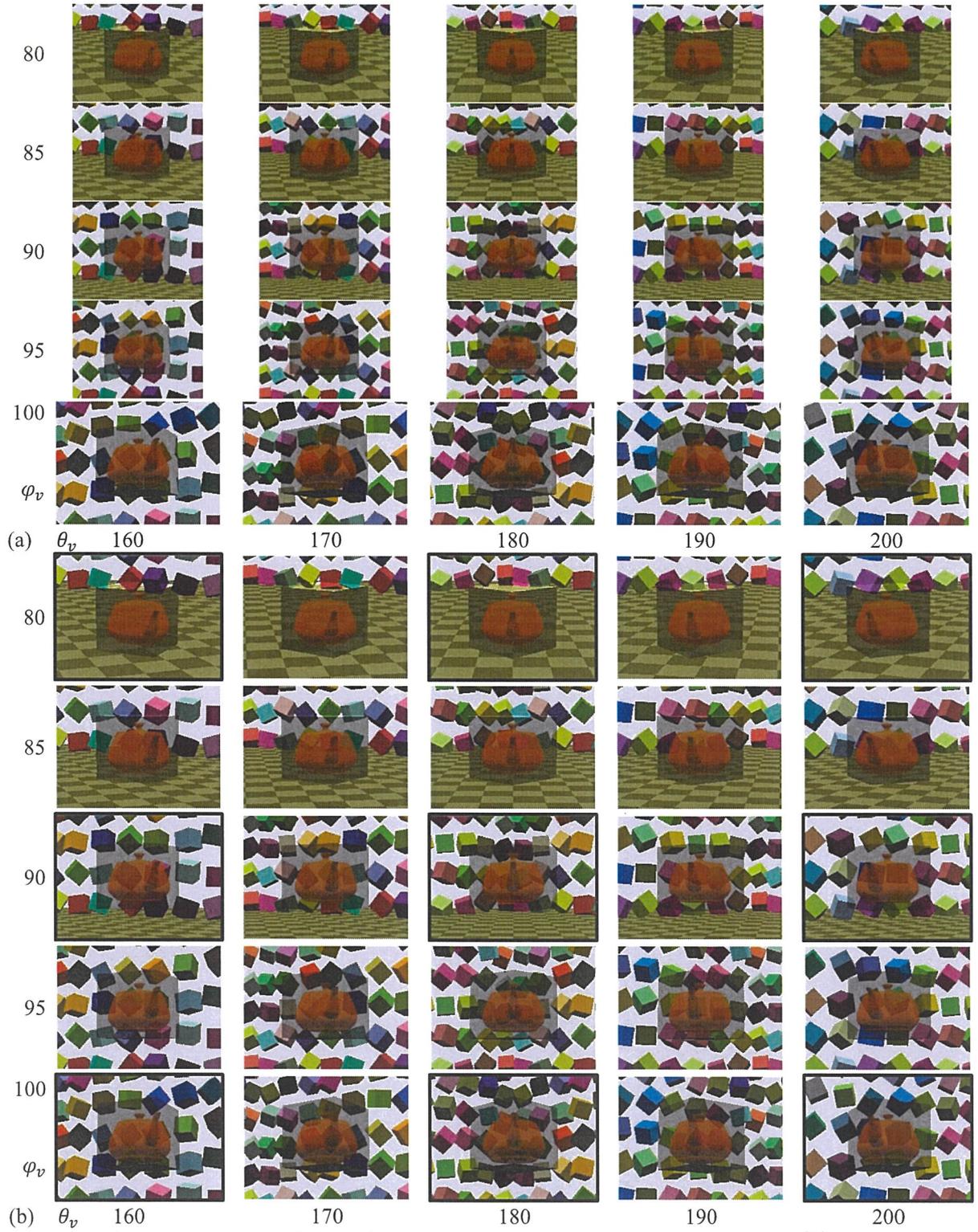


Figure 11. Augmented views for different viewer's positions in Exp.4-2. $r_v = r_v^{ini} = 3.0$.
 (a) Background parameters are not interpolated. (b) Background parameters are interpolated.

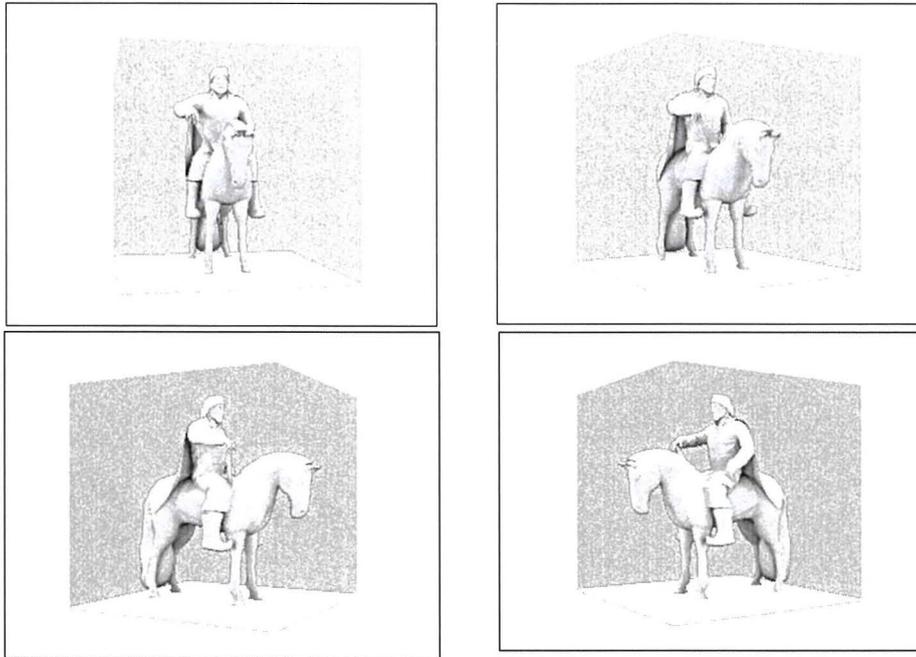


Figure 12. Virtual object of Equestrian Statue (Tsonjin Boldog).

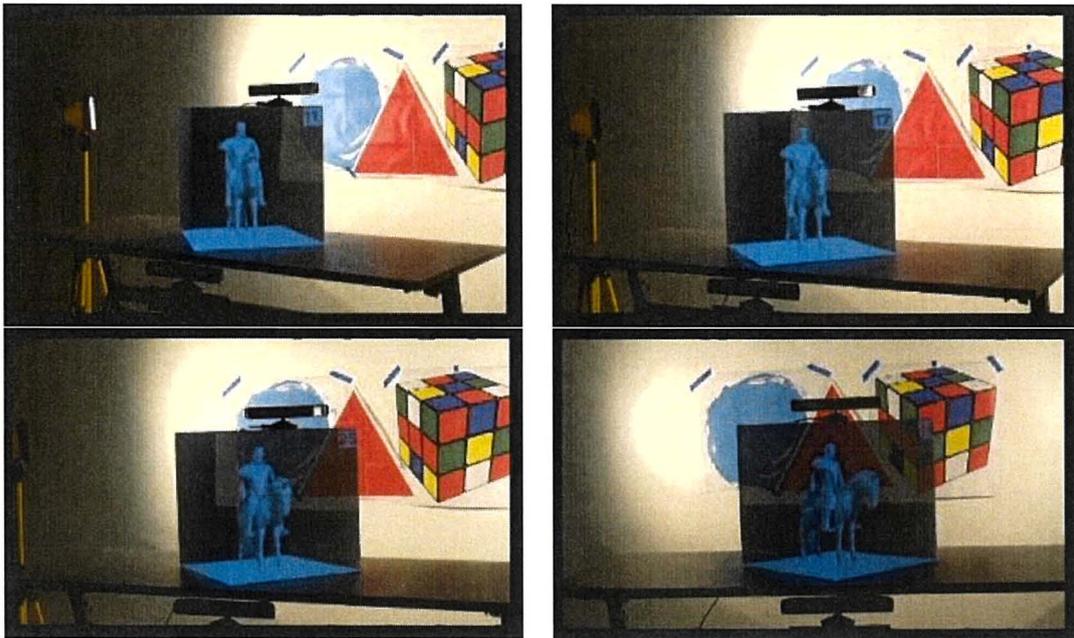


Figure 13. Augmented views in real environment with near planar background.

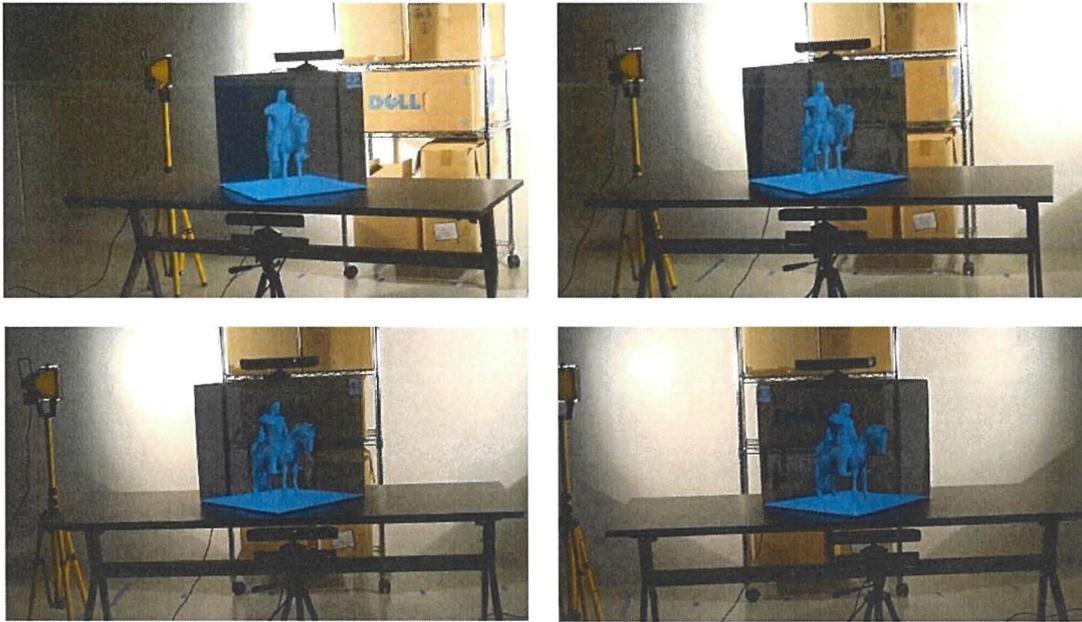


Figure 14. Augmented views in real environment with near non-planar background.

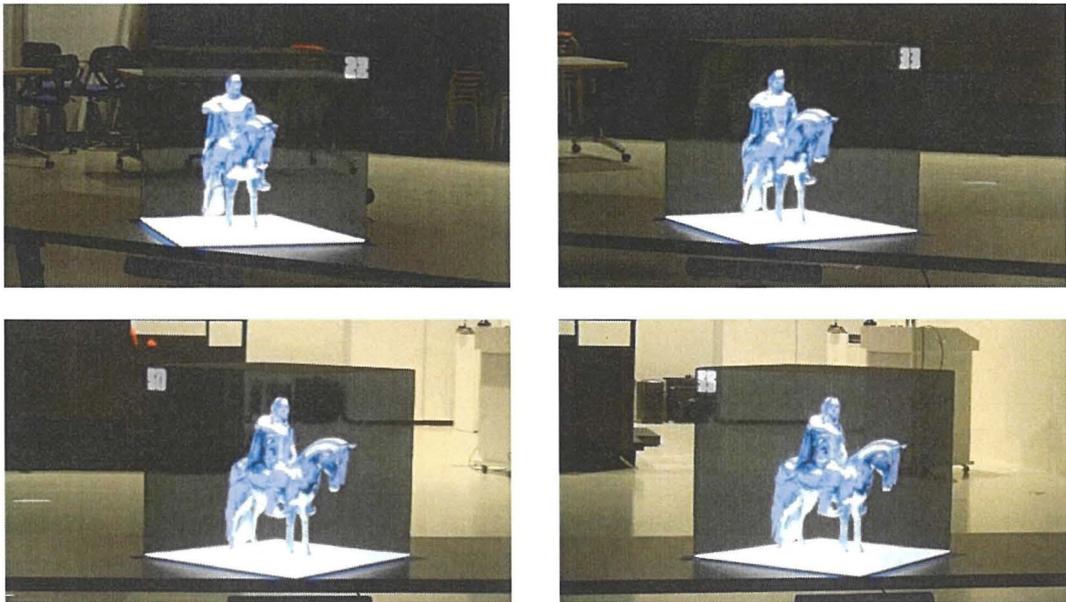


Figure 15. Augmented views in real environment with far complicated-shaped background.

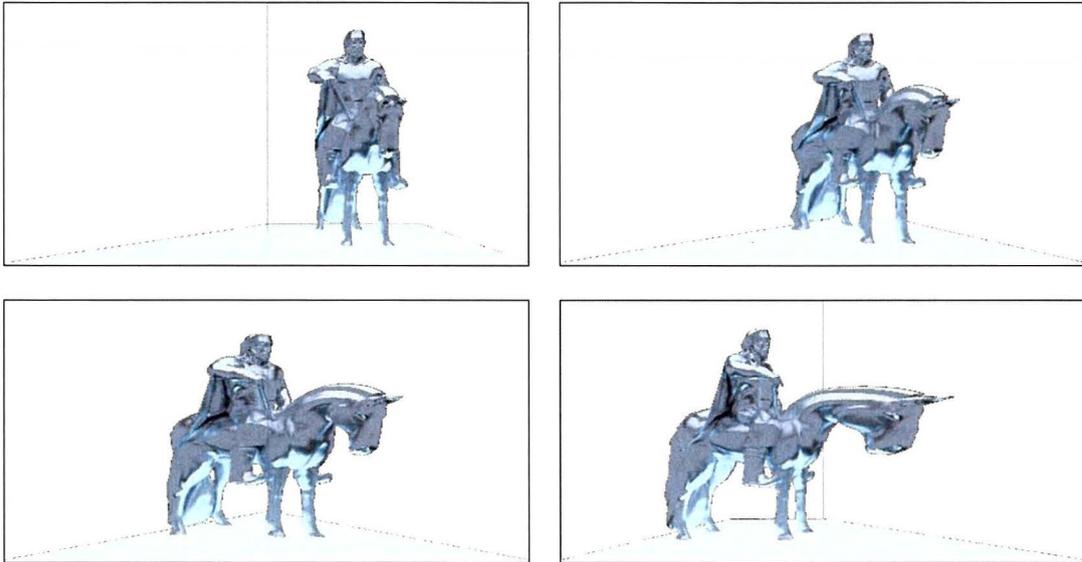


Figure 16. Projection images used in experiment of Figure 15.

5. CONCLUSION

In SAR, when a virtual object's shape is different from a real object's shape, the difference of the shapes causes undesired empty areas onto which the virtual object is not projected on the real object's surface. We proposed a view-dependent method to eliminate such empty areas by projecting the real background behind the real object. To treat a far background in a large space, the real background's image is captured by an RGB camera and converted to an image for the viewer's position based on homography. The image conversion uses the approximation of the background's shape by a background plane, which is defined by practical background parameters interpolated on a 3D grid space. This adjusts the projected background so as to match the directly-observed background according to the viewer's position. Consequently, our method makes the viewer not feel the presence of the real object but feel the virtual object merging into the real world.

Currently, the presence of a real object is not eliminated perfectly due to the difference of the brightness of a projected background and a directly-observed background. This should be appropriately improved by photometric calibration. Our method has a serious problem; the view-dependent display is available to only a single viewer and not available to multiple viewers simultaneously. However, the simplicity of our method

enables the integration with an existing method developed for multiple viewers. This is an important future work. Currently, a background plane is adjusted by giving background parameters manually. The development of an efficient way to determine optimal parameters automatically or semi-automatically is also a future work. In the current practice, a camera to capture a background and a tracking sensor to track a viewer are fixed near a real object and directed toward the sides opposite to each other. This configuration allows a viewer to see only one side of a virtual object. Two Kinect sensors are used as the camera and the tracking sensor, respectively. If each Kinect sensor works as both a camera and a tracking sensor, the viewer can see two opposite sides of the virtual object. By expanding this configuration, we are planning to develop a system to allow a viewer to enjoy a 360-degree view of a virtual object merging into its real background by using multiple Kinect sensors.

There are definite possibilities to refine and further our study in advance. We also would like to multiply the number of 3D Models to view and add more variety. Switching between different Objects and different 3D Models would be preferable. The further configuration will give this study the ability to be viewed by multiple viewers at the same time. In future work, there will be an adjustment of the brightness and contrast of the projected backgrounds and the directly-observed background. This should be appropriately improved by photometric calibration. The development of an efficient way to determine optimal parameters automatically or semi-automatically is also in plan for future work.

REFERENCES

- [1] Oliver Bimber and Ramesh Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*, A K Peters/CRC Press, 2005.
- [2] Alessandra Meschini, Daniele Rossi, Enrica Petrucci, and Filippo Sicuranza, *Expanded Cultural Heritage Representation: Digital Applications for Mixed-Reality Experiences*, *Handbook of Research on Emerging Technologies for Digital Preservation and Information Modeling*, pp.256-287, IGI Global, 2017.
- [3] Microsoft Research, *User Interfaces for Spatial Augmented Reality*, 2016.
<https://www.youtube.com/watch?v=qZWpQ8Tuweo>
- [4] Ramesh Raskar, *Spatial Augmented Reality: Inserting in the Real World*, 2018.
<https://www.youtube.com/watch?v=4tRmDDM8ITs>
- [5] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs, *The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays*, *Proc. of ACM SIGGRAPH 1998*, pp.179-188, 1998.
- [6] Oliver Bimber, Daisuke Iwai, Gordon Wetzstein, and Anselm Grundhöfer, *The Visual Computing of Projector-Camera Systems*, *Computer Graphics Forum*, vol.27, 8, pp.2219-2245, 2008.
- [7] Anselm Grundhöfer and Daisuke Iwai, *Recent Advances in Projection Mapping Algorithms, Hardware and Applications*, *Computer Graphics Forum*, vol.37, 2, pp.653-675, 2018.
- [8] Ramesh Raskar, Greg Welch, Kok-lim Low, Deepak Bandyopadhyay, *Shader Lamps: Animating Real Objects With Image-Based Illumination*, *Proc. of Eurographics Workshop on Rendering Techniques 2001*, pp.89-102, 2001.
- [9] Ramesh Raskar and Paul Beardsley, *A Self-Correcting Projector*, *Proc. of IEEE Computer Society*

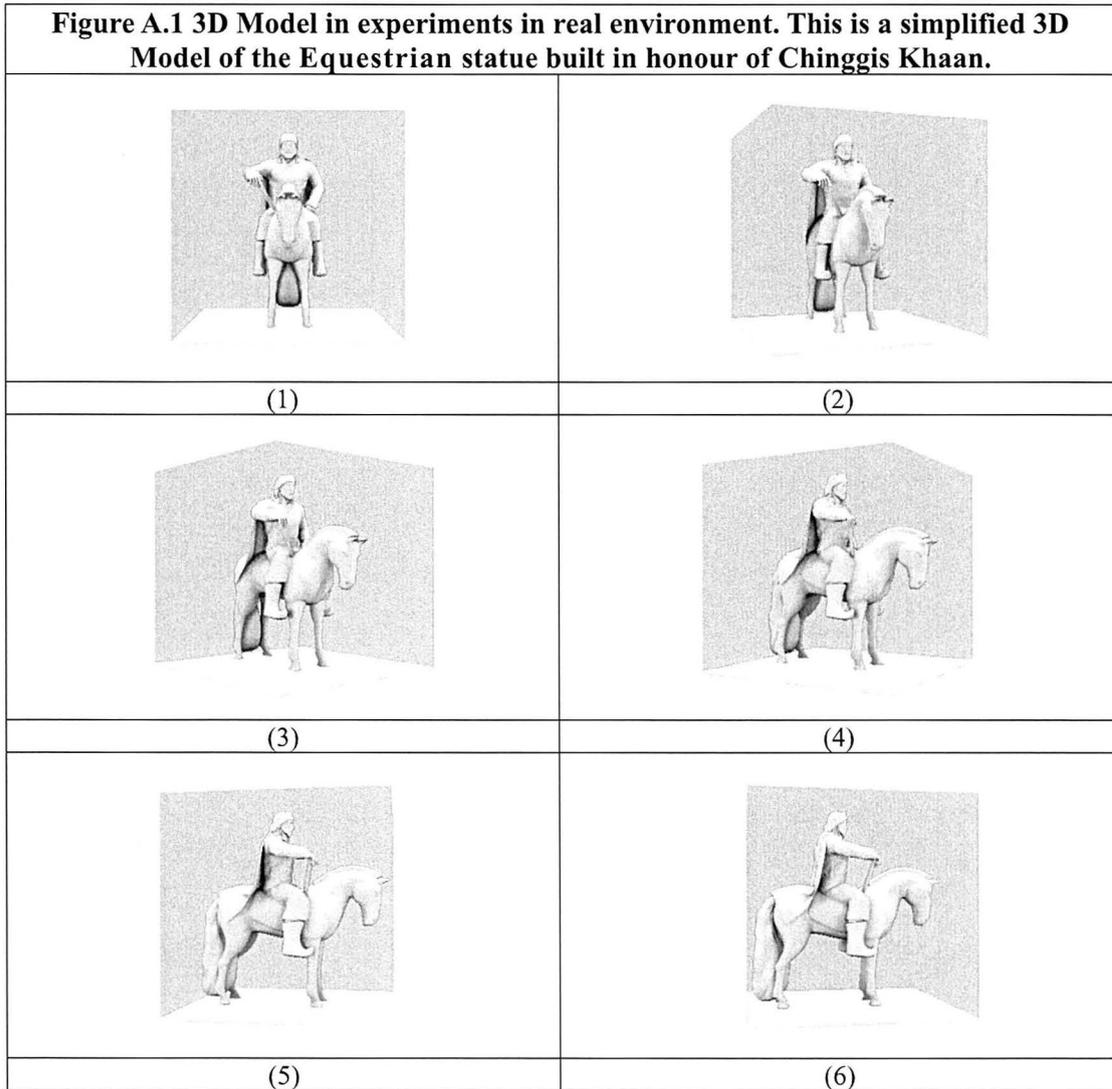
- Conference on Computer Vision and Pattern Recognition (CVPR) 2001, pp.626-631, 2001.
- [10]Samuel Audet and Masatoshi Okutomi, A User-Friendly Method to Geometrically Calibrate Projector-Camera Systems, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2009, pp. 47-54, 2009.
- [11]Tuotuo Li, Feng Hu, and Zheng Geng, Geometric Calibration of a Camera-Projector 3D Imaging System, Proc. of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI 2011), pp.187-194, 2011.
- [12]Daniel Moreno and Gabriel Taubin, Simple, Accurate, and Robust Projector-Camera Calibration, Proc. of the 2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission, pp.464-471, 2012.
- [13]I. Din, H. Anwar, I. Syed, H. Zafar, and L. Hasan, Projector Calibration for Pattern Projection Systems, Journal of Applied Research and Technology, vol.12, 1, pp.80-86, 2014.
- [14]Tyler Johnson and Henry Fuchs, Real-Time Projector Tracking on Complex Geometry Using Ordinary Imagery, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2007, pp.1-8, 2007.
- [15]Shuntaro Yamazaki, Masaaki Mochimaru, and Takeo Kanade, Simultaneous Self-Calibration of a Projector and a Camera Using Structured Light, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2011, pp.60-67, 2011.
- [16]Behzad Sajadi, Mahdi Abbaspour Tehrani, Mehdi Rahimzadeh, Aditi Majumder, High-Resolution Lighting of 3D Reliefs Using a Network of Projectors and Cameras, Proc. of 2015 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp.1-4, 2015.
- [17]Oliver Fleischmann and Reinhard Koch, Fast Projector-Camera Calibration for Interactive Projection Mapping, Proc. of 2016 23rd International Conference on Pattern Recognition (ICPR), pp.3798-3803, 2016.
- [18]Simon Willi and Anselm Grundhöfer, Robust Geometric Self-Calibration of Generic Multi-Projector Camera Systems, Proc. of 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp.42-51, 2017.
- [19]Jaewoon Lee, Yeonjin Kim, Myeong-Hyeon Heo, Dongho Kim, and Byeong-Seok Shin, Real-Time Projection-Based Augmented Reality System for Dynamic Objects in the Performing Arts, Symmetry, vol.7,

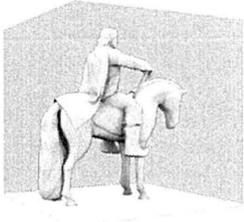
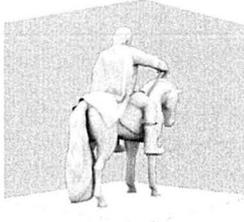
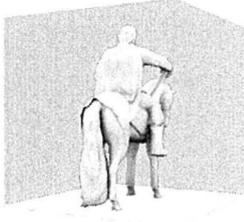
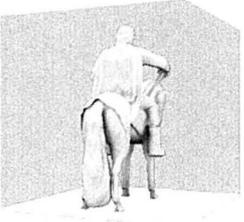
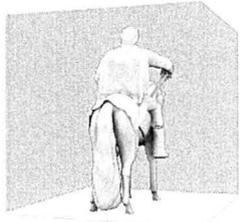
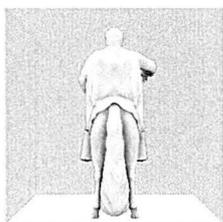
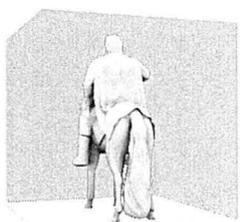
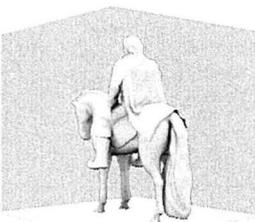
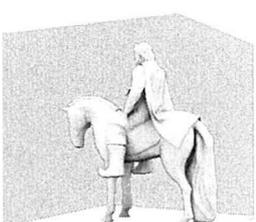
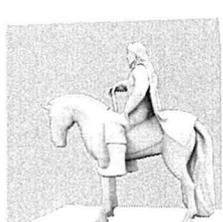
- 1, pp.182-192, 2015.
- [20]Christian Siegl, Matteo Colaianni, Lucas Thies, Justus Thies, Michael Zollhöfer, Shahram Izadi, Marc Stamminger, and Frank Bauer, Real-Time Pixel Luminance Optimization for Dynamic Multi-Projector Mapping, *ACM Transactions on Graphics*, vol.34, 6, article no.237, pp.1-11, 2015.
- [21]Yuichiro Fujimoto, Ross T. Smith, Takafumi Taketomi, Goshiro Yamamoto, Jun Miyazaki, Hirokazu Kato, and Bruce H. Thomas, Geometrically-Correct Projection-Based Texture Mapping onto a Deformable Object, *IEEE Transactions on Visualization and Computer Graphics*, vol.20, 4, pp.540-549, 2014.
- [22]Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa, Dynamic Projection Mapping onto Deforming Non-Rigid Surface Using Deformable Dot Cluster Marker, *IEEE Transactions on Visualization and Computer Graphics*, vol.23, 3, pp.1235-1248, 2017.
- [23]Amit H. Bermano, Markus Billeter, Daisuke Iwai, and Anselm Grundhöfer, Makeup Lamps: Live Augmentation of Human Faces via Projection, *Computer Graphics Forum*, vol.36, 2, pp.311-323, 2017.
- [24]Andreas Fender, David Lindlbauer, Philipp Herholz, Marc Alexa, and Jörg Müller, HeatSpace: Automatic Placement of Displays by Empirical Analysis of User Behavior, *Proc. of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST 2017)*, pp.611-621, 2017.
- [25]Andreas Fender, Philipp Herholz, Marc Alexa, Jörg Müller, OptiSpace: Automated Placement of Interactive 3D Projection Mapping Content, *Proc. of the 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018)*, paper no.269, pp.1-11, 2018.
- [26]Hrvoje Benko, Andrew D. Wilson, and Federico Zannier, Dyadic Projected Spatial Augmented Reality, *Proc. of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST 2014)*, pp.645-655, 2014.
- <https://www.youtube.com/watch?v=D6yWWMwjkg>
- <https://www.youtube.com/watch?v=Df7fZAYVAIE>
- [27]Wojciech Matusik and Hanspeter Pfister, 3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes, *Proc. of ACM SIGGRAPH 2004*, pp.814-824, 2004.
- [28]Tibor Balogh, Péter Tamás Kovács, and Zoltán Megyesi, HoloVizio 3D Display System, *Proc. of 2007 3DTV Conference*, pp.1-4, 2007.

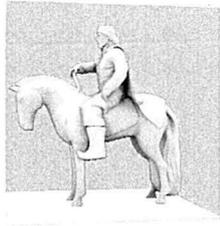
- [29] Andrew Jones, Ian McDowall, Hideshi Yamada, Mark Bolas, Paul Debevec, Rendering for an Interactive 360° Light Field Display, Proc. of ACM SIGGRAPH 2007, article no.40, pp.1-10, 2007.
- [30] Joel Jurik, Andrew Jones, Mark Bolas, and Paul Debevec, Prototyping a Light Field Display Involving Direct Observation of a Video Projector Array, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2011, pp.15-20, 2011.
- [31] Toshiyuki Amano and Kensuke Minami, Structural Color Display on Retro-Reflective Objects, Proc. of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments (ICAT-EGVE), pp.37-44, 2015.
- [32] Andrew Jones, Jonas Unger, Koki Nagano, Jay Busch, Xueming Yu, Hsuan-Yueh Peng, Oleg Alexander, Mark Bolas, and Paul Debevec, An Automultiscopic Projector Array for Interactive Digital Humans, Proc. of ACM SIGGRAPH 2015 Emerging Technologies, article no.6, 2015.
- [33] Shunsuke Yoshida, fVisiOn: Interactive Glasses-Free Tabletop 3D Images Floated By Conical Screen and Modular Projector Arrays, Proc. of ACM SIGGRAPH Asia 2015 Emerging Technologies, article no.12, pp 1-3, 2015.
- [34] Min Ki Park, Kyu Je Lim, Myoung Kook Seo, Soon Jong Jung, Kwan H. Lee, Spatial Augmented Reality for Product Appearance Design Evaluation, Journal of Computational Design and Engineering, vol.2, 1, pp.38-46, 2015.
- [35] Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs, Multi-Projector Displays Using Camera-Based Registration, Proc. of IEEE Visualization, pp.161-168, 1999.
- [36] Ruigang Yang and Greg Welch, Automatic and Continuous Projector Display Surface Calibration Using Every-day Imagery, Proc. of the 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2001), pp.328-335, 2001.
- [37] Azuma, Ronald T. "A survey of augmented reality." MIT Press, Presence 6.4 (1997): 355-385.
- [38] Milgram, P., Kishino, F. A Taxonomy of Mixed Reality Visual Displays. IECE Trans. on Information and Systems (Special Issue on Networked Reality), vol. E77-D, no. 12, pp.1321-1329, 1994.

Appendix

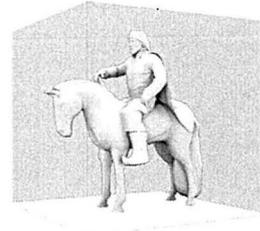
A. Images in experiments in real environment



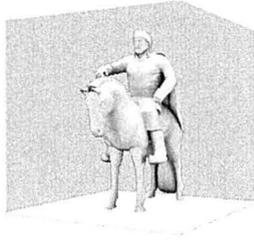
	
(7)	(8)
	
(9)	(10)
	
(11)	(12)
	
(13)	(14)
	
(15)	(16)



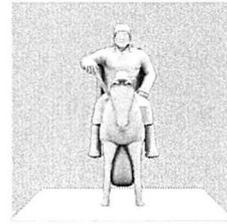
(17)



(18)

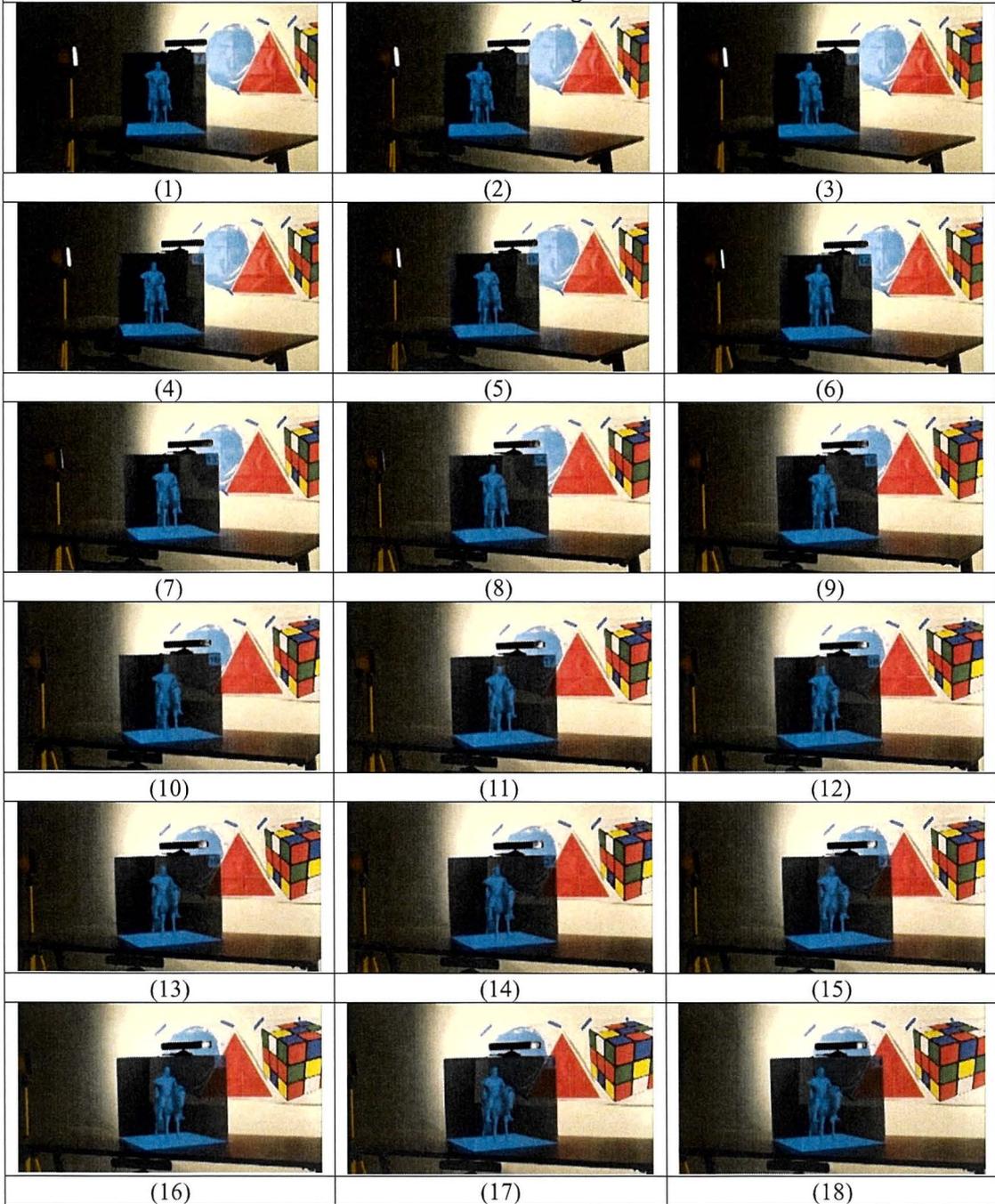


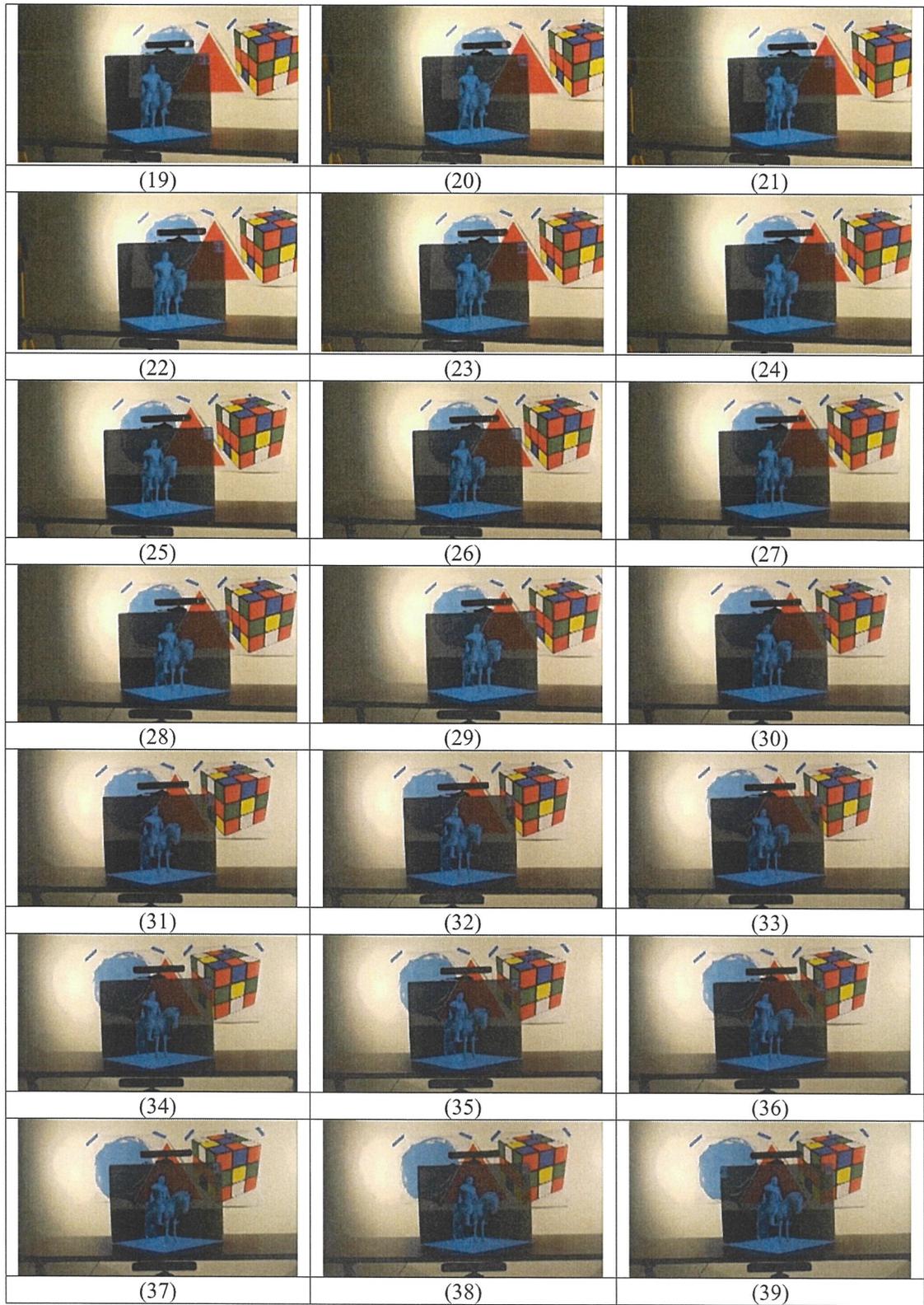
(19)



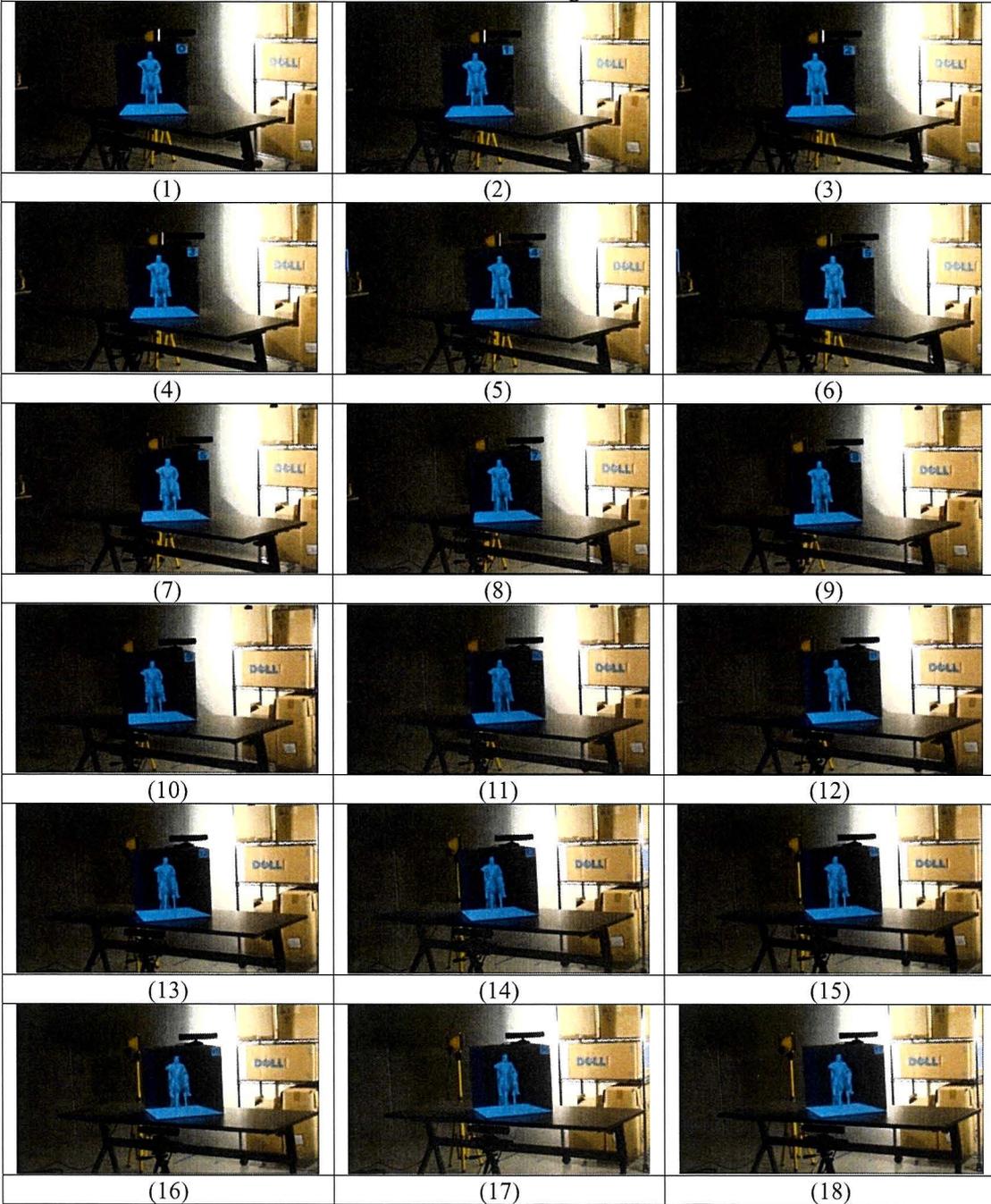
(20)

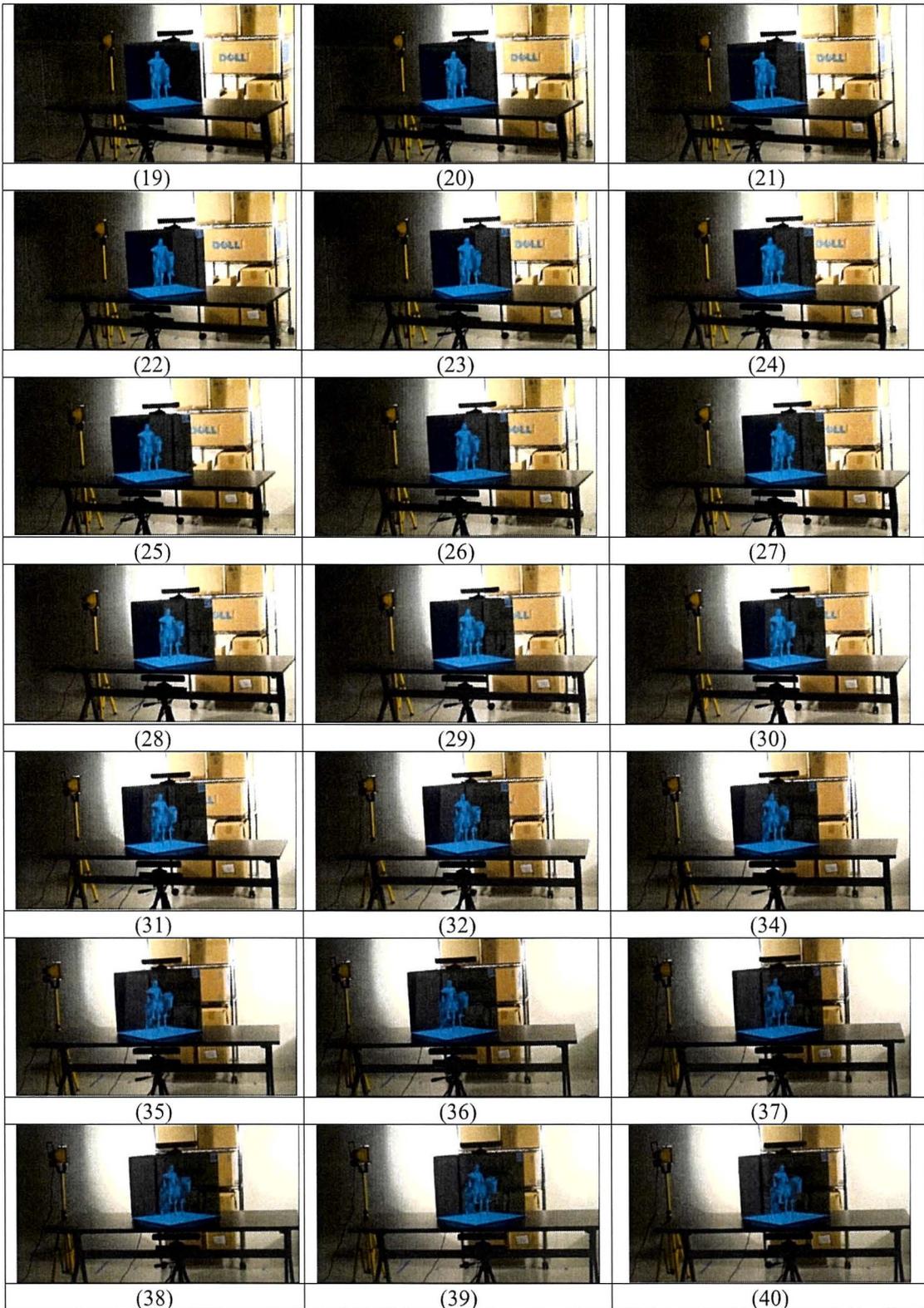
A.2 Augmented views in real environment with near planar background. The Distance between the cube and the background is about 1.5 meters.

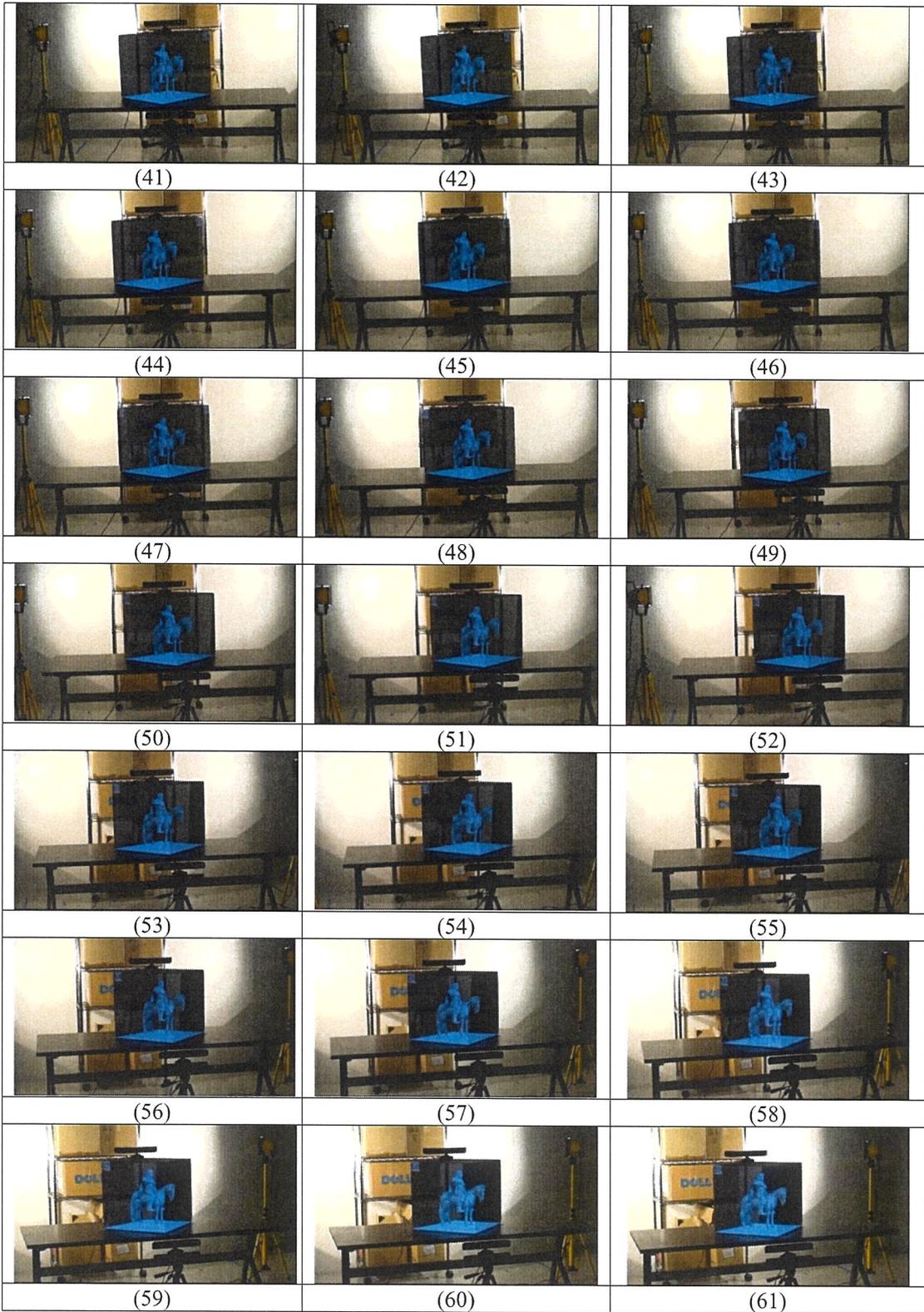


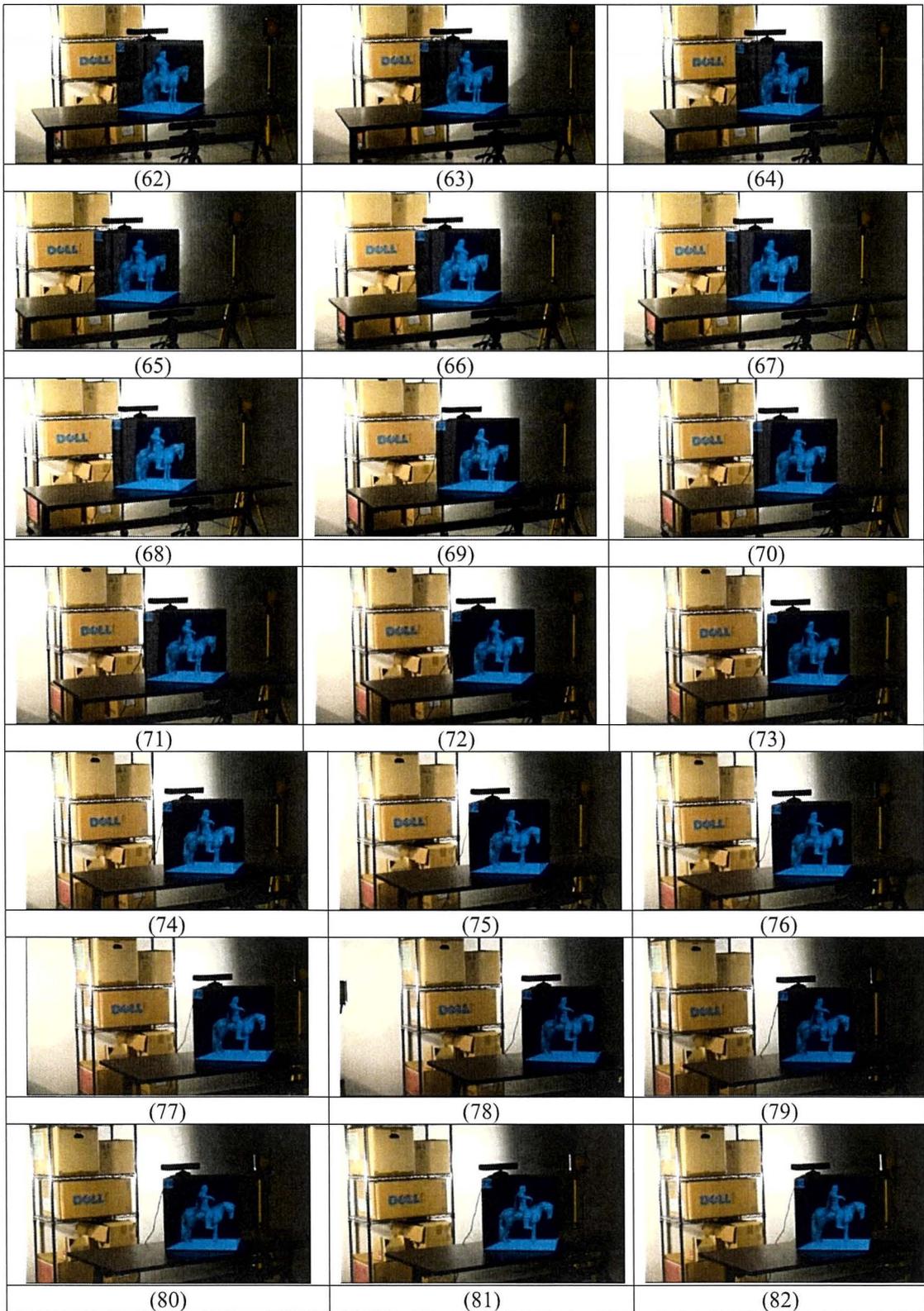


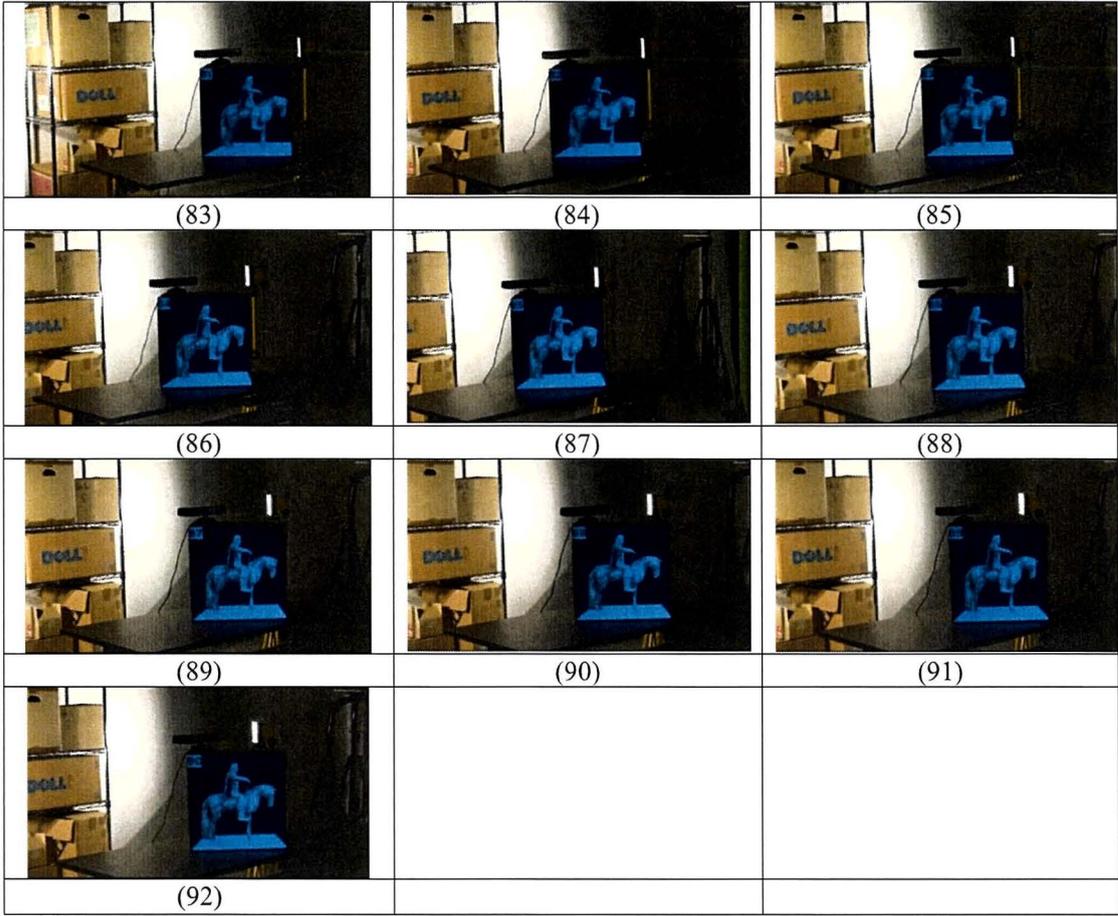
A.3 Augmented views in real environment with near non-planar background. The distance between the cube and the background is about 1.5 meters.



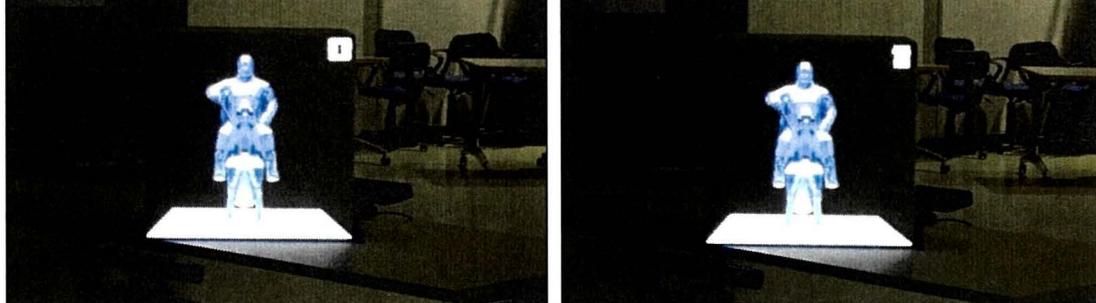




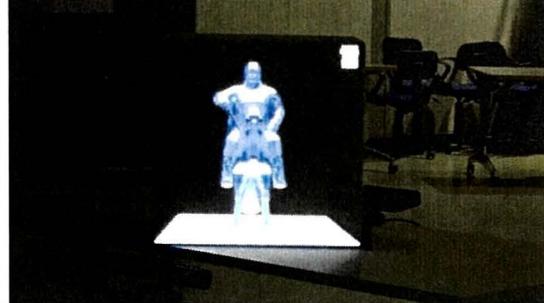




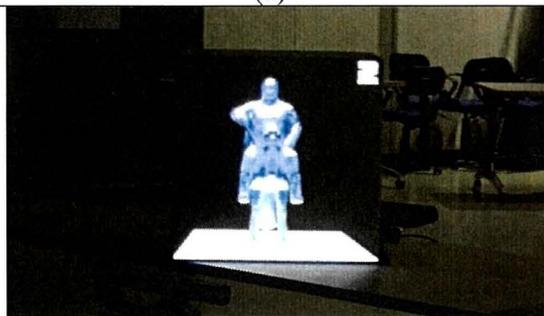
**A.4 Augmented views in real environment with far complicated-shaped background.
The distance between the cube and the background is about 9 to 14 meters.**



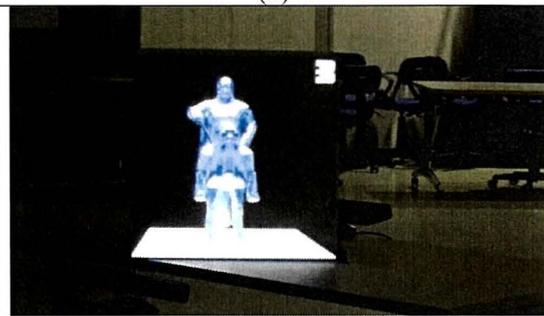
(1)



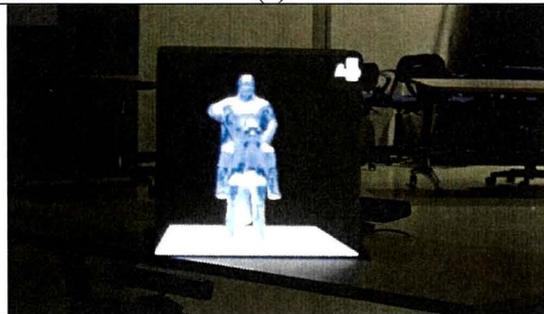
(2)



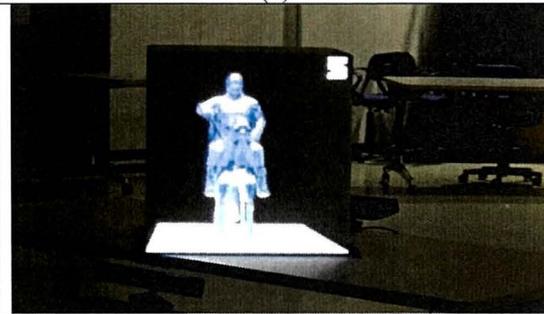
(3)



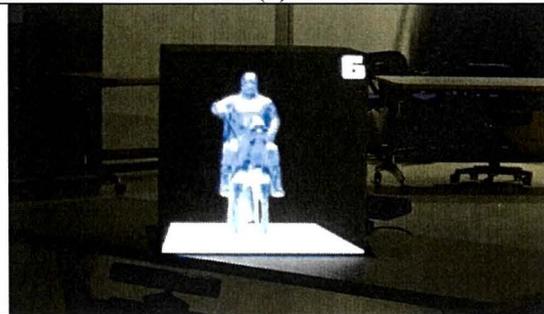
(4)



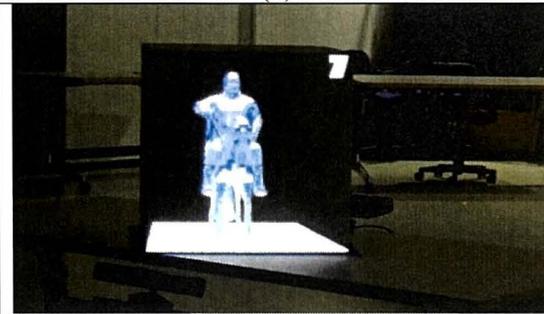
(5)



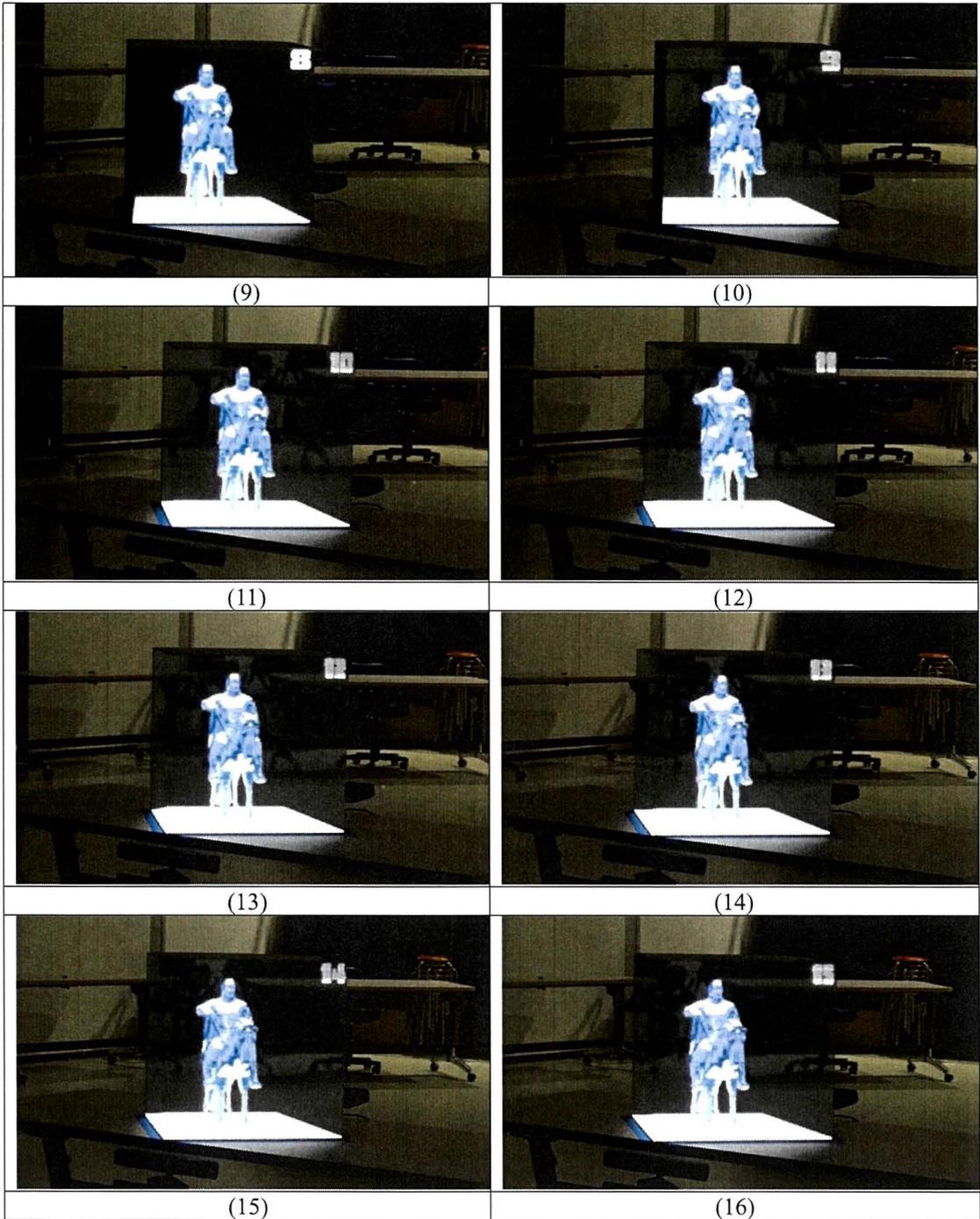
(6)

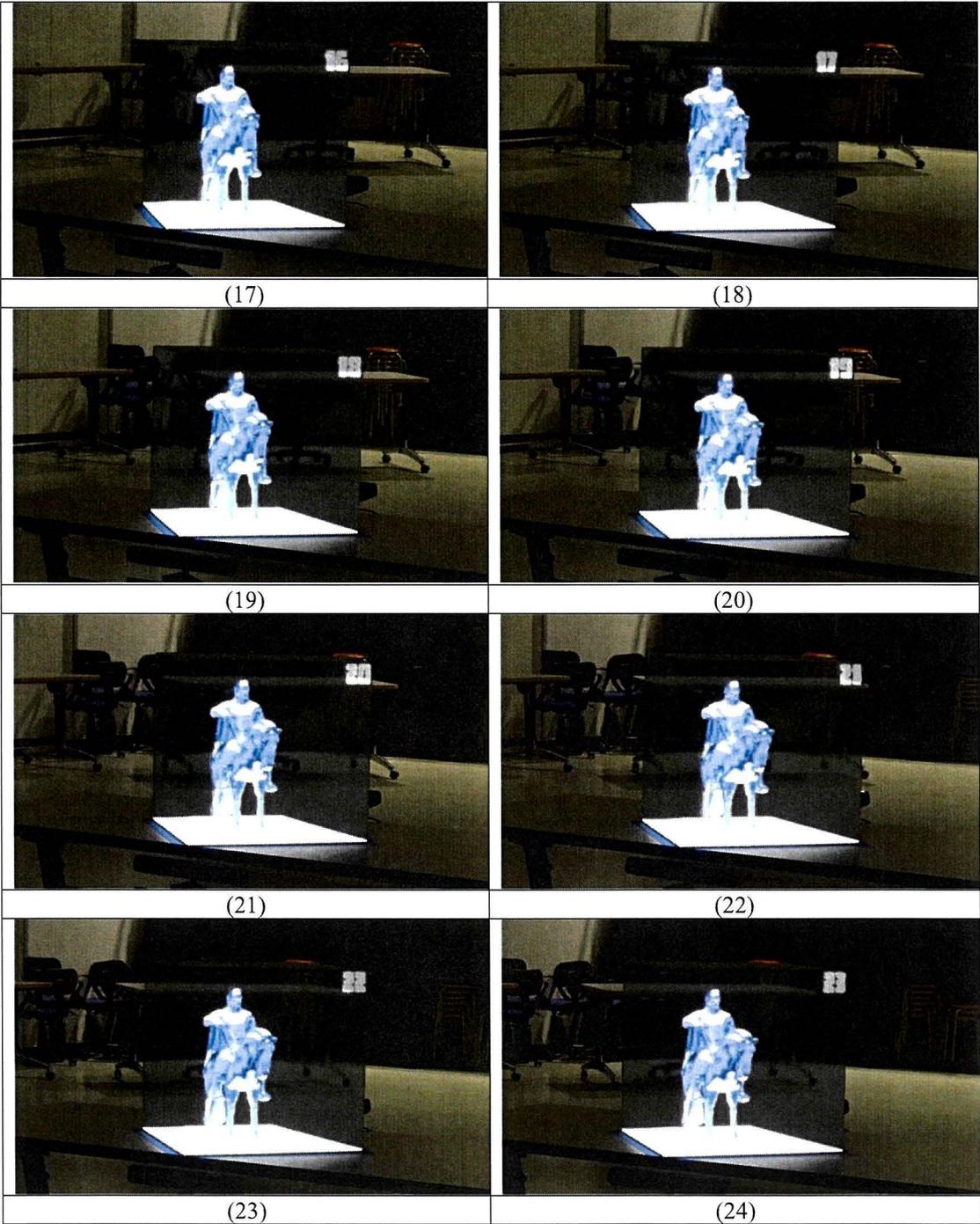


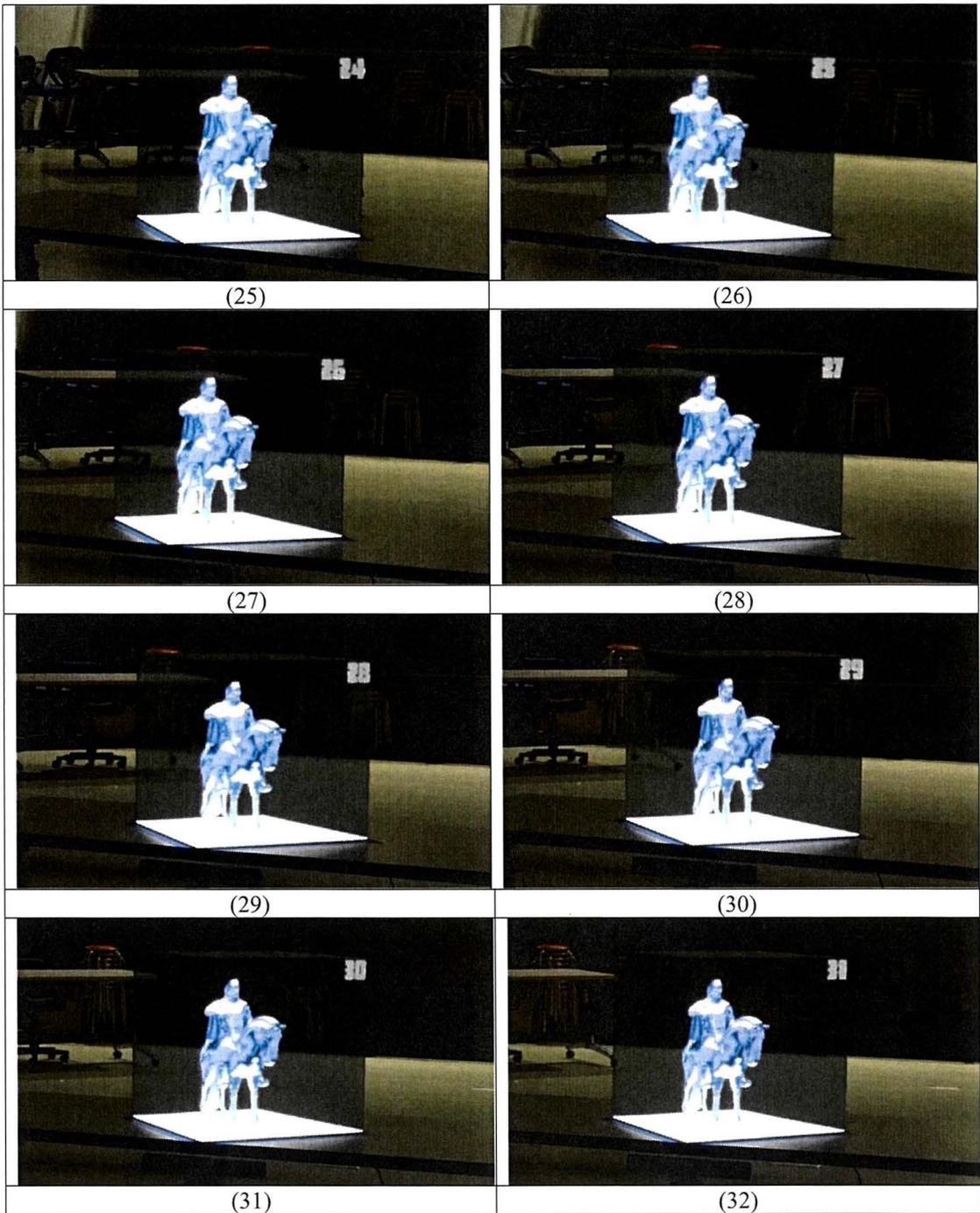
(7)

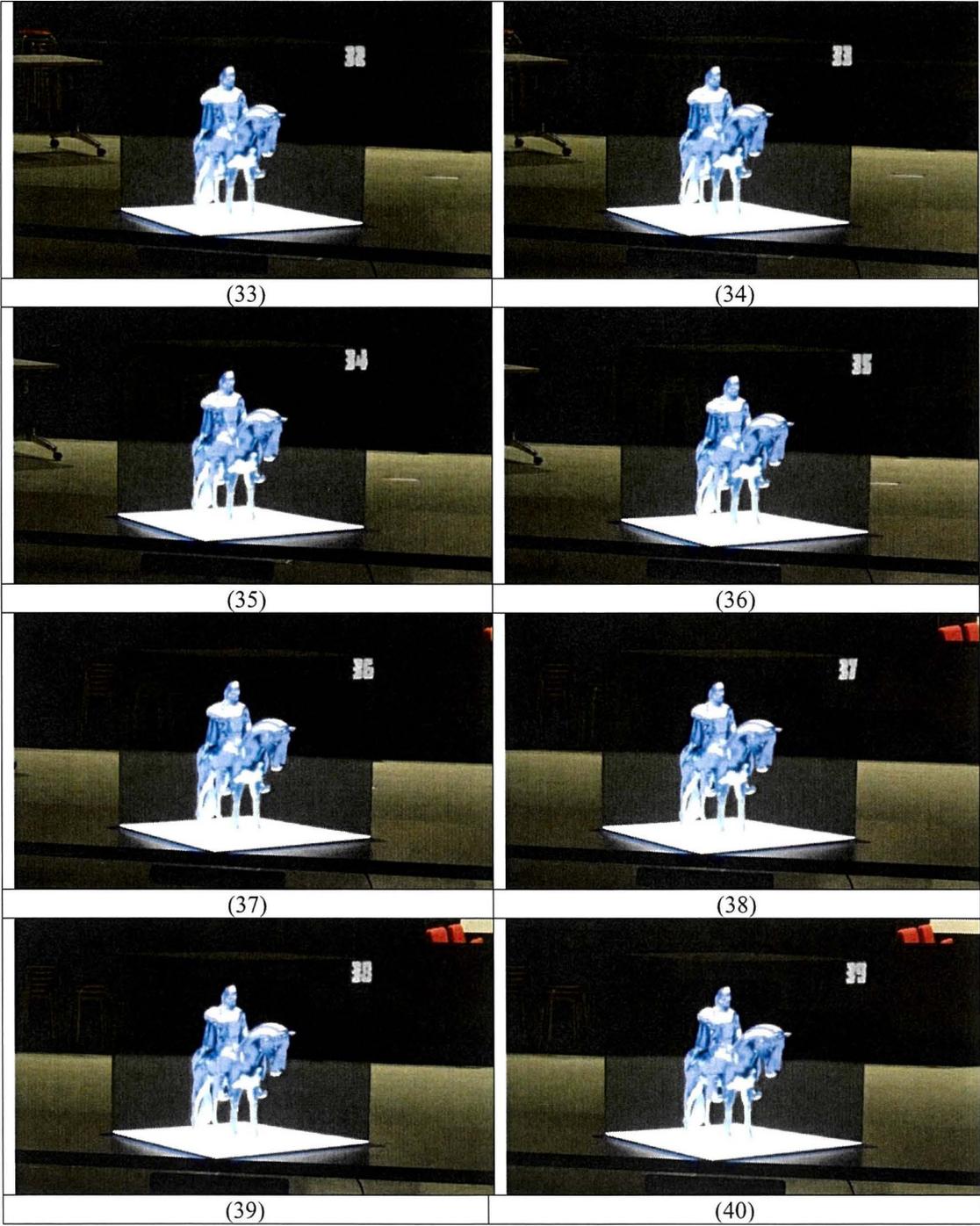


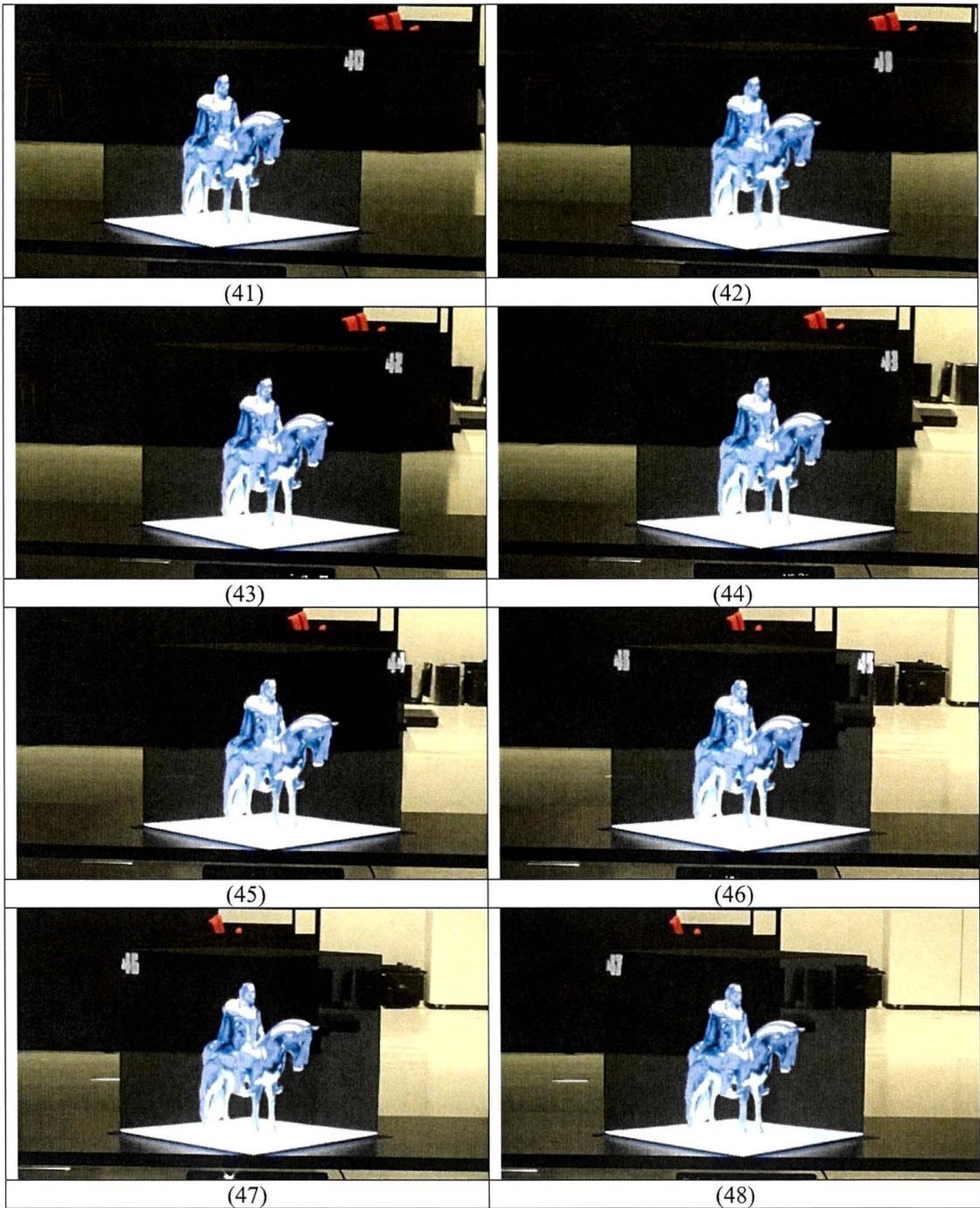
(8)

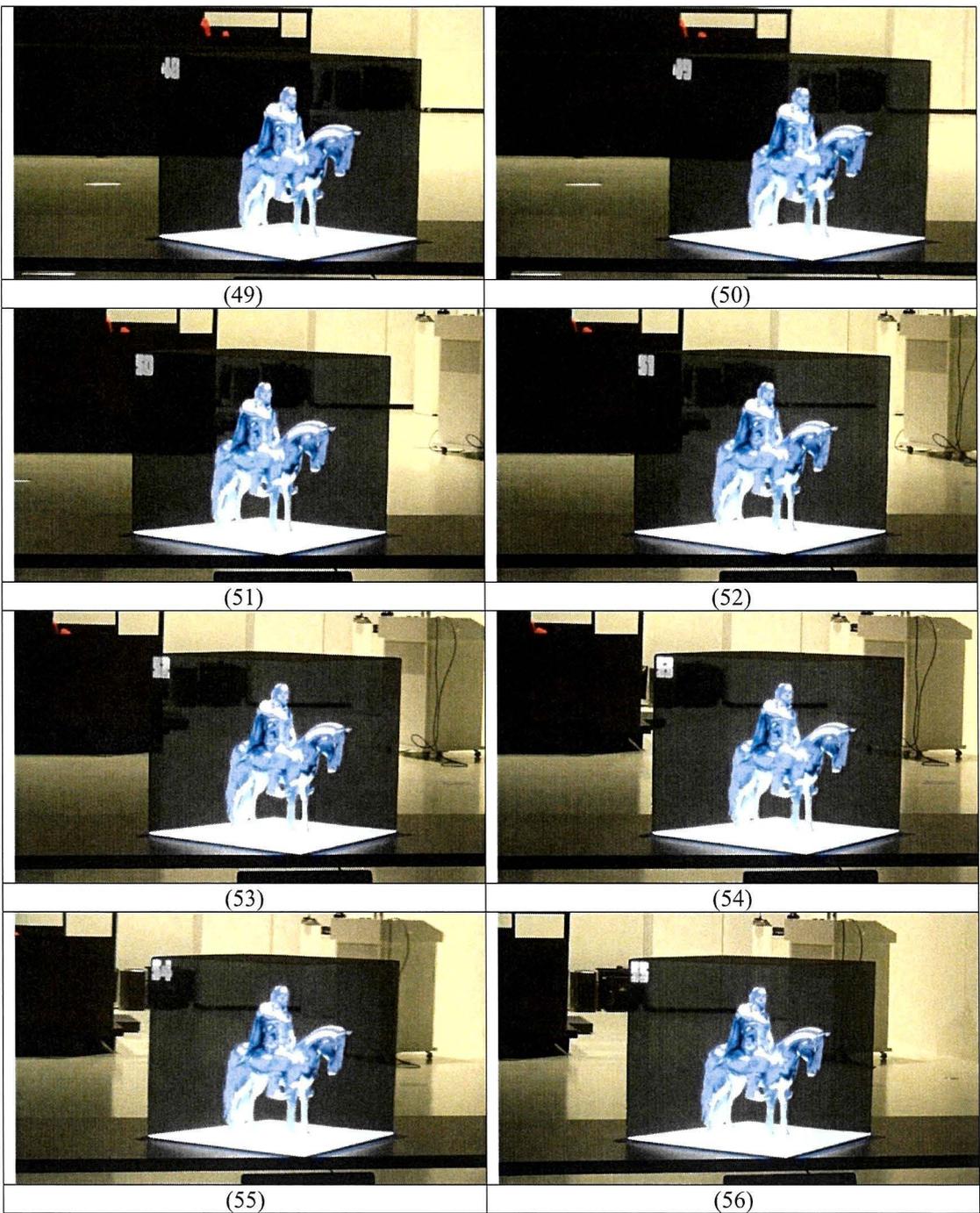


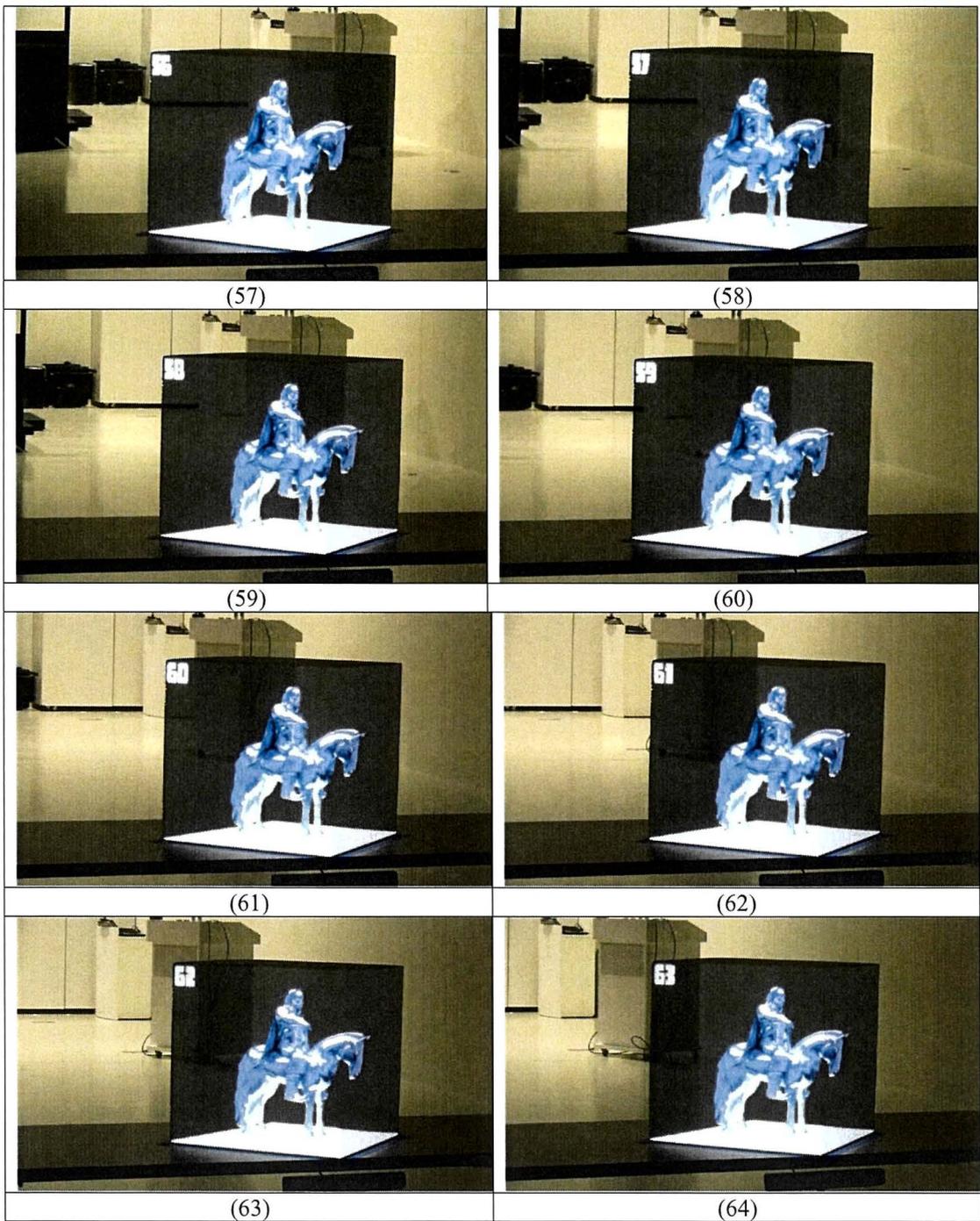


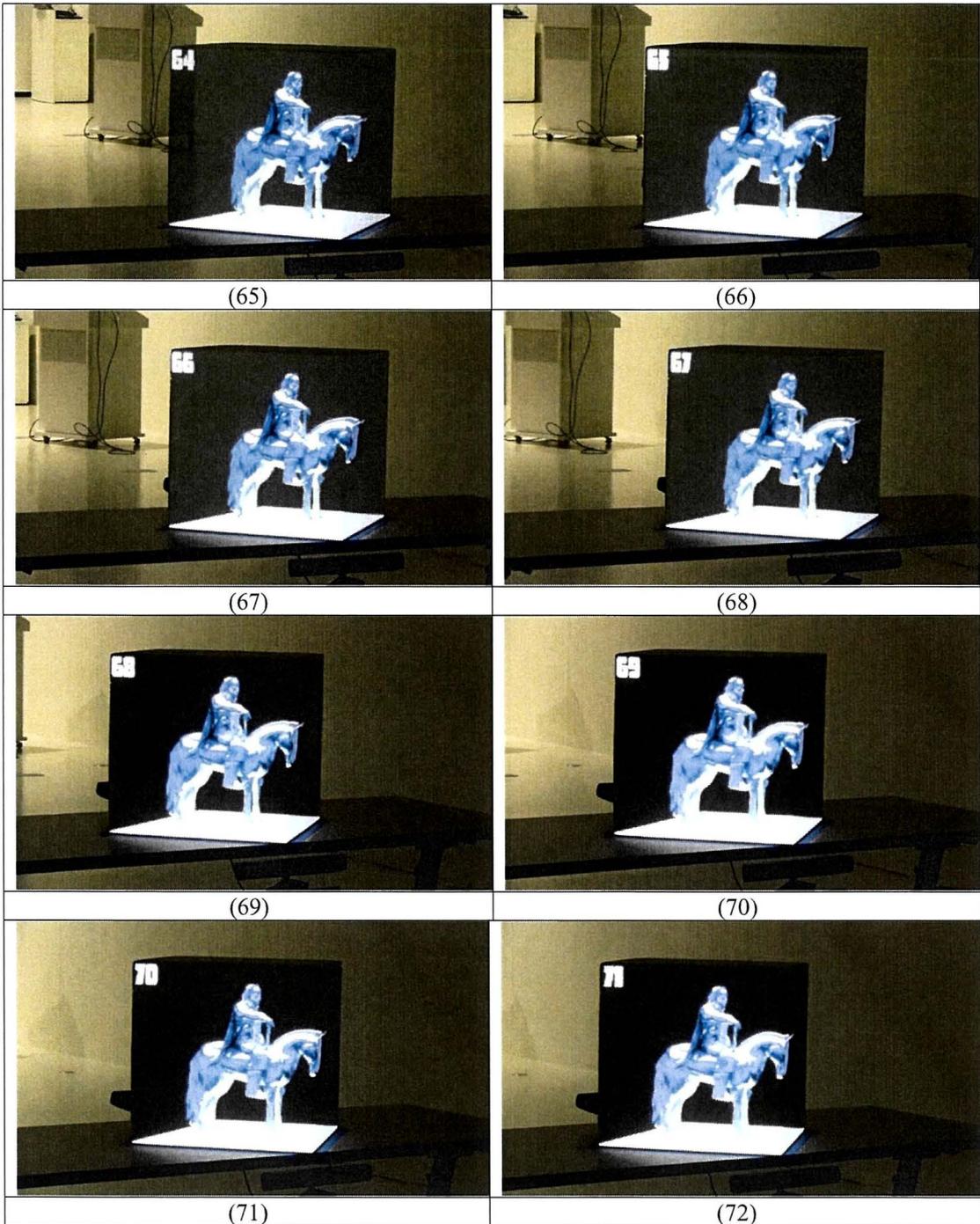


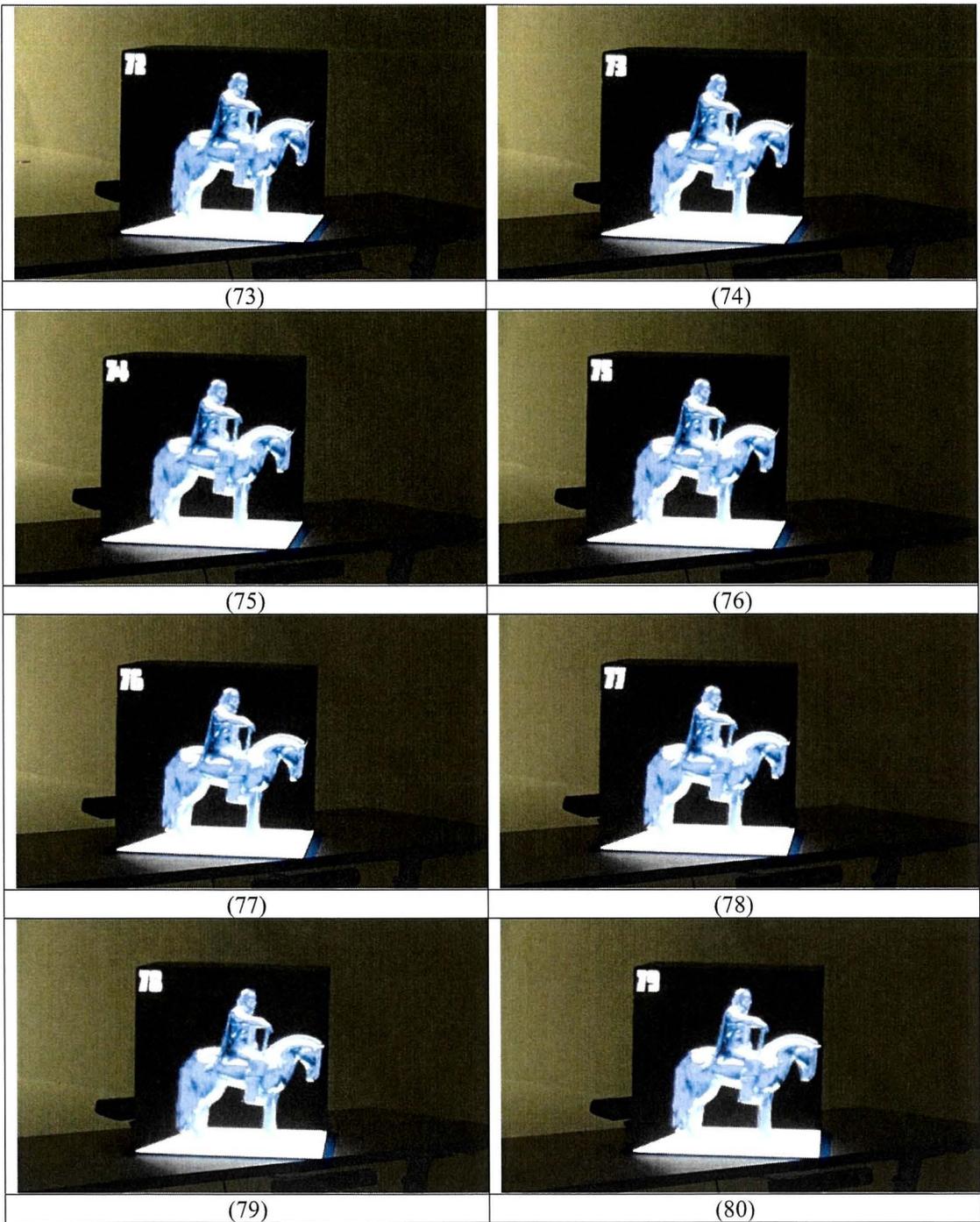


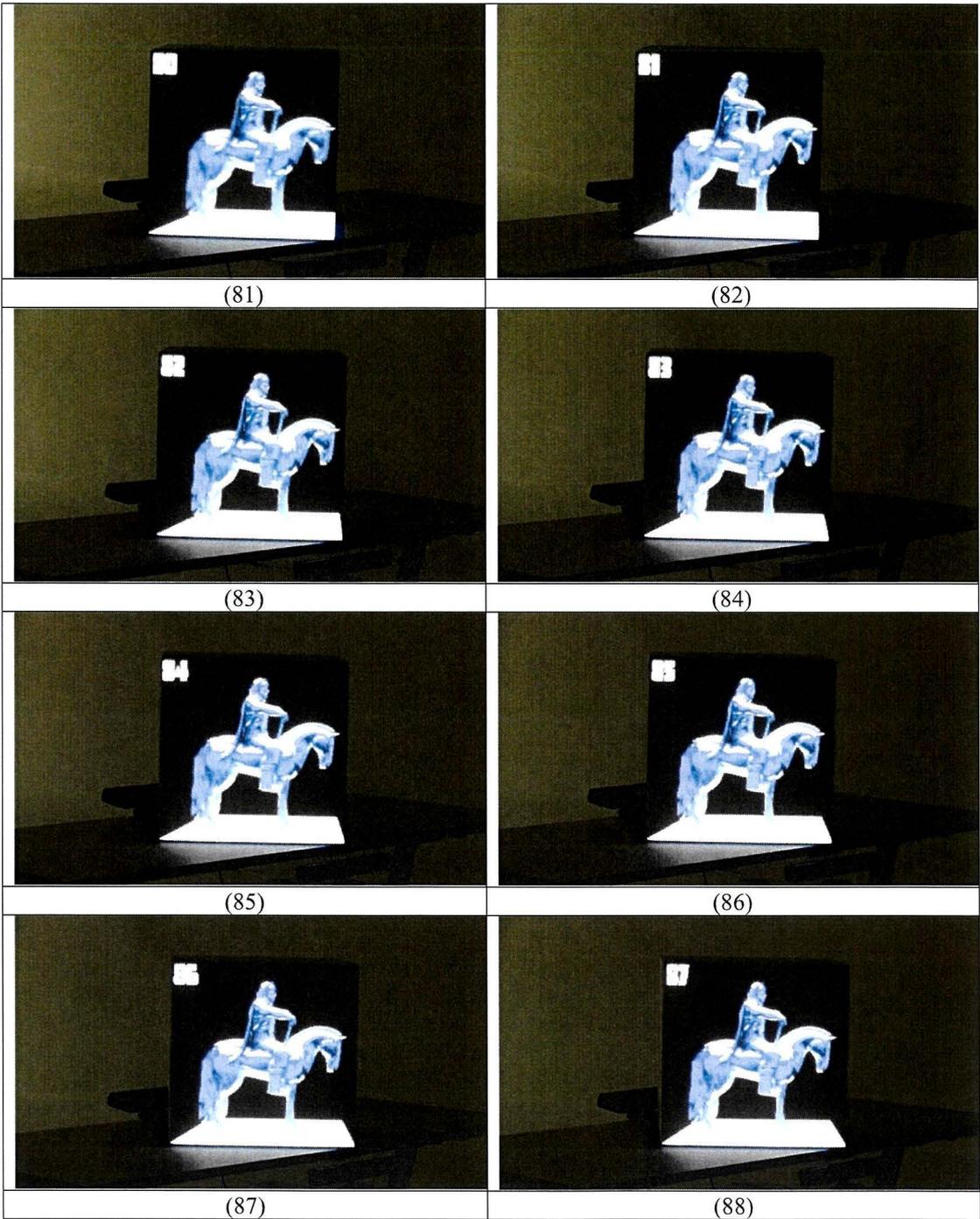


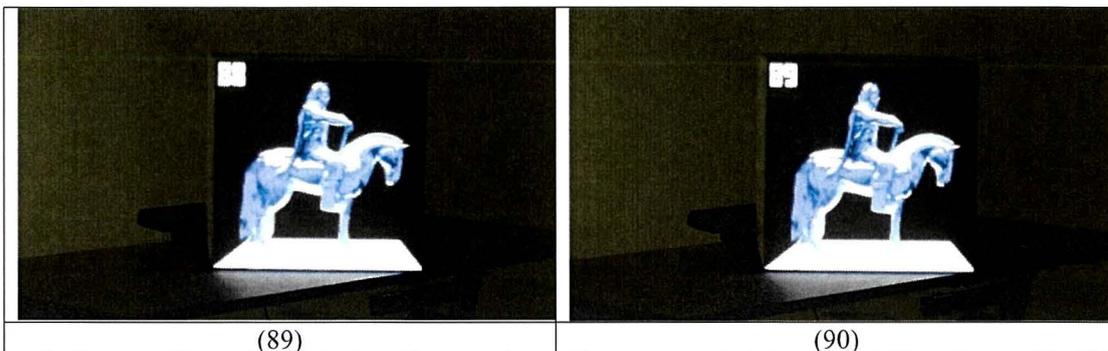












A.5 Augmented views in real environment with far complicated-shaped background.
These were obtained in a different lighting condition from the condition of Figure A.4.

