OCCLUSION METHOD BASED ON THREE-DIMENSIONAL FEATURE DATA FOR MOBILE AUGMENTED REALITY APPLICATIONS[1]

Hiroki Ito & Koji Makanae
Miyagi University, Japan

ABSTRACT: *In recent years, the demand for augmented reality (AR) applications has been increasing owing to the spread of smartphones equipped with cameras and various sensors. Hence, there is a need for a high-speed processing method to build three-dimensional (3D) AR applications on mobile devices. In particular, an occlusion process, which determines the virtual object to be drawn on the real image, is indispensable. However, this process requires dynamic depth data of objects in real and virtual spaces in accordance with the movement of the camera. In this paper, we developed an AR tsunami simulation, which uses an occlusion method that cuts out a virtual water-wave by the 3D feature model obtained from real space. The application was developed by using the Unity game engine, which provides realistic rendering functions and compatibility with various sensors in smartphones with an occlusion program that uses OpenGL shader. As a result, the developed AR application performed the occlusion process without sacrificing processing power.*

The development of AR applications requires careful consideration of processing speed and accuracy in accordance with the requirements of presentation. The visualization of tsunami waves should also be improved; for example, sprays of wave collision should be represented.

In the future, development of a more realistic representation technique and a simple method to retrieve depth data dynamically from real space will be necessary.

KEYWORDS: *Augmented Reality, Occlusion, OpenGL, Smartphone, Game Engine.*

1. INTRODUCTION

Augmented reality (AR) technology, a technique that can present information effectively, has attracted wide attention in various fields such as construction, education, entertainment, and health care.

In comparison with virtual reality (VR) technology, the advantages of AR are that it is unnecessary to prepare 3D models that correspond to real space and that it enables an intuitive understanding because it adds information to real space.

Kakuda et al. (2007) developed a “Virtual Asukakyo” system that enables one to view restored ruins in an outdoor environment by using AR technology. The system superimposes the ruins reproduced in computer graphics (CGs) on the physical space by examining where they existed with head-mounted display (HMD).

Meng et al. (2012) developed an AR system that aids the understanding of human anatomy by using Microsoft’s RGB-D camera “Kinect” to track the skeleton of a person and superimpose the internal image of the human body as if seen in an X-ray camera.

In recent years, the demand for AR applications is increasing with the spread of mobile terminals including smartphones. Smartphones, which make a significant contribution to the spread of AR technology as portable hardware devices, can retrieve external information from GPS, cameras, and various sensors to present information intuitively by touch displays and speakers.

However, a high-speed processing technique is required to implement AR expression with high accuracy on mobile devices. In particular, occlusion processes are indispensable for superimposing objects of virtual space

[1] Citation: Ito, H. & Makanae, K. (2013). Occlusion method based on three-dimensional feature data for mobile augmented reality applications. In: N. Dawood and M. Kassem (Eds.), Proceedings of the 13th International Conference on Construction Applications of Virtual Reality, 30-31 October 2013, London, UK.

onto physical space. High processing power is required to calculate dynamic depth data in accordance with the movement of a camera.

In this paper, we propose and verify occlusion methods of virtual objects on smartphones by registration through the use of existing real space information. We also describe the experimental results of an AR tsunami simulation by using a game engine to validate this method.

2. OCCLUSION IN AR SCENES

2.1 Occlusion problem

Occlusion processes are required to represent the positional relationships between virtual and real objects in AR. Figure. 1 shows an example of occlusion images. Figure 1(a) and (b) show the AR images from a camera positioned in bird's eye view. In (a), reality has been impaired by the virtual blue cylinder object displayed in front of the real building because the positional relationship is incorrectly expressed. In contrast, (b) correctly represents the positional relationship because the cylinder clipped according to the shape of the building. Depth information in real space is required to correctly superimpose the virtual object, as shown in (b).

2.2 Occlusion methods and challenges

Suyang and Vineet. (2010) studied a robust occlusion method that uses real-time Time-of-flight (TOF) camera data to correctly resolve the depth of real and virtual objects in AR images. Obtaining high-accuracy depth information without being affected by the environment is possible by using LiDAR and ToF cameras. However, these methods cannot take advantage of the portability of a smartphone because a depth camera must be carried and set up.

The methods for acquiring depth information by image processing procedures such as pattern matching do not require additional hardware other than smartphones. However, the smartphones do need high processing power. Image processing is also difficult to use on smartphones because it is strongly influenced by outdoor environments.

The aim of this research is to present a method for solving the occlusion problem of AR by aligning the virtual and real objects in virtual space by using existing spatial information (e.g., created, for instance, by aerial surveys) to take advantages of the convenience portability of smartphones.

3. RESOLUTION OF OCCLUSION PROBLEM BY USING GEO LOCATION

3.1 Combining 3D models and registration

In the first phase, the proposed method constructs a virtual space to simulate the occlusion of virtual objects by real objects. Figure 2 shows the relations between the virtual and real spaces. The virtual space contains real objects as feature information of real space, virtual objects as additional information to be superimposed, and a viewpoint as user's position in real space.

Real objects are composed of their geographic coordinates and 3D model data that represent the shape of the feature in real space. These objects are located in the virtual space at a position based on its geographic coordinates; the occlusion of virtual objects is processed by the 3D model data.

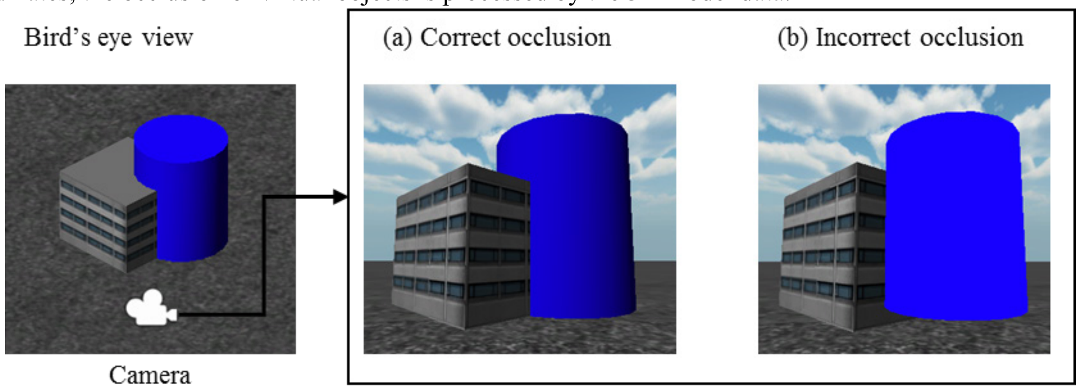

Fig. 1: Examples of incorrect and correct occlusion in AR scene

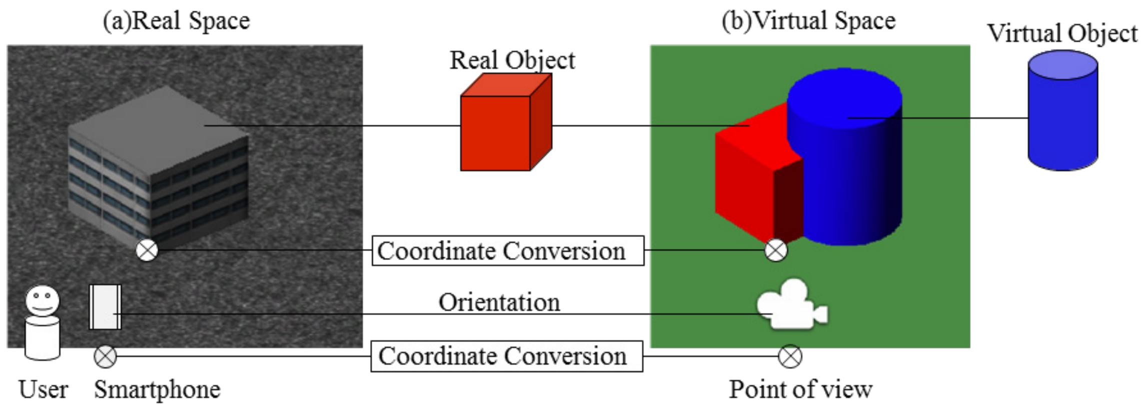

Fig. 2: Construction of virtual space

The position of a viewpoint in the virtual space is determined on the basis of the geographical coordinates acquired by the GPS sensor of the smartphone. The orientation of a viewpoint is updated on the basis of the angle and direction of the smartphone camera, which is obtained from the acceleration, magnetometric sensor, gyro sensors, and so on.

The geographic coordinates obtained from the GPS sensor must be translated to CG coordinates to place real objects and the viewpoint based on real space positions. A virtual object is positioned anywhere; depending on its information, however, coordinate conversion is performed as well to specify its position by geographic coordinates.

The movement of the user (smartphone) in real space can be expressed by that of the viewpoint in the virtual space. AR expression and occlusion in accord with the video image for each frame can be performed by updating the parameters of the viewpoint on the basis of the orientation information for the camera and GPS coordinates.

3.2 AR system using a smartphone

Figure 3 shows a hardware diagram of the AR system involving the use of a smartphone. The AR system is installed as an application to the smartphone. The application updates the orientation and position of the viewpoint in virtual space by obtaining the orientation from various sensors and the geographical coordinates from the GPS. The application obtains frame-by-frame video images of the physical space taken by the camera to display the AR Image and superimposed virtual objects simultaneously.

3.3 Generation of AR Images

Figure 4 shows the processing flow of rendering AR images. (c) is an example of a rendered image of the virtual space from a viewpoint. (a) and (b) are images of rendered real and virtual objects respectively. The occluded mask image shown in (d) can be obtained by extracting a part of the virtual object from (c). In the final step, (d) is superimposed on (e), which is obtained from the smartphone camera, to create the AR image (f).

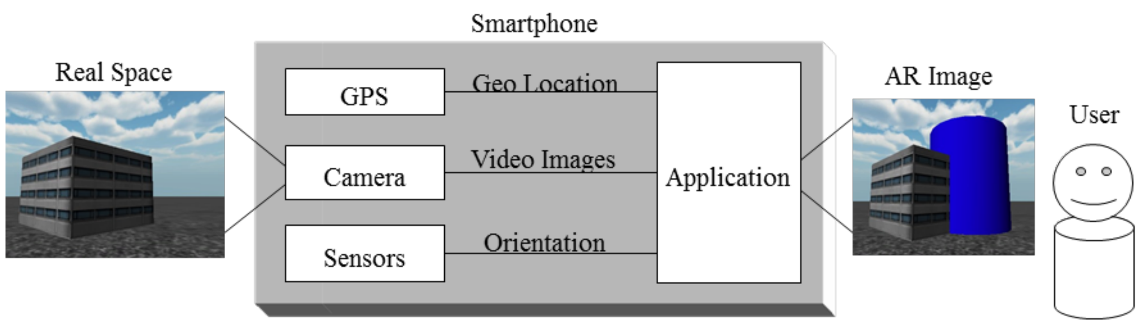

Fig. 3: Diagram of an AR smartphone system

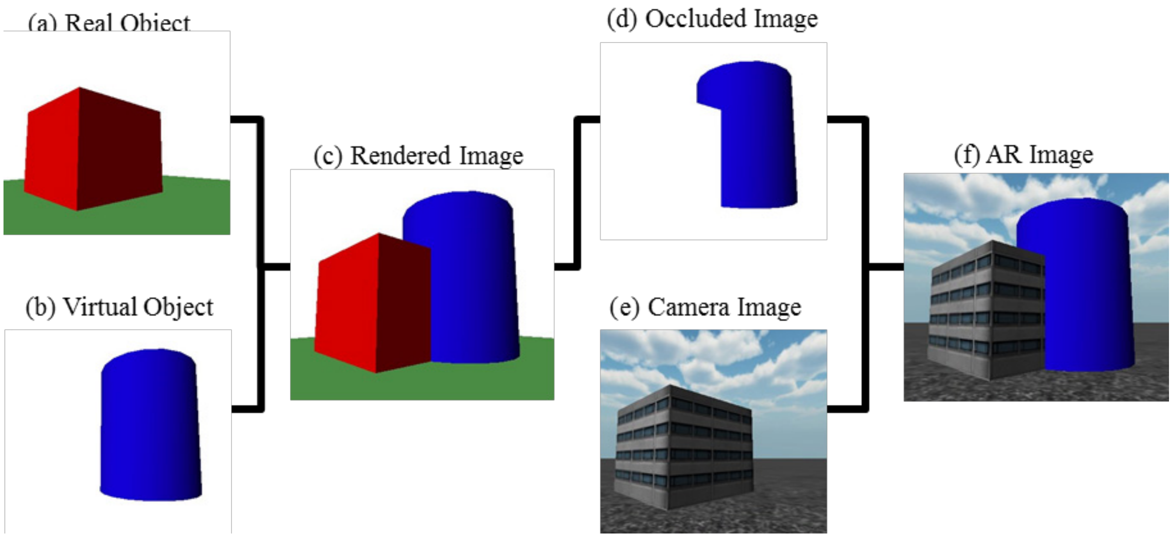

Fig. 4: Processing flow of rendering AR images

4. USE OF UNITY GAME ENGINE TO BUILD AR APPLICATION FOR SMARTPHONE

4.1 Tsunami Simulation Application

In this study, we developed a tsunami simulation for the Android operating system (OS) to evaluate the method described in the previous section. This simulation has been developed for simulating tsunami. By superimposing a tsunami CG as a virtual object on real space, it displays a tsunami image when the user is standing in the flooded area.

We used Google Nexus7 as an alternate device of smartphone. We also used the Unity 3.5 game engine and Android SDK as the development environment.

4.2 Construction of flood virtual space

The first step is to place existing feature models and a viewpoint in a virtual space by using the geographic coordinates obtained from the GPS of the smartphone.

We made feature models from cartographic information and acquired geographic coordinates as the real objects. If the geographical coordinates of a model exist, its position can be adjusted accordingly. Table 1 shows the conversion of each coordinate system. The application converts WGS84 coordinates obtained from the GPS to the Japanese plane rectangular coordinates, which in turn are converted to CG coordinates in the game engine. We included feature model data in the application on a trial basis in this time. The application reads the feature model data as real objects and convert the geographic coordinates to CG coordinates to place them in virtual space.

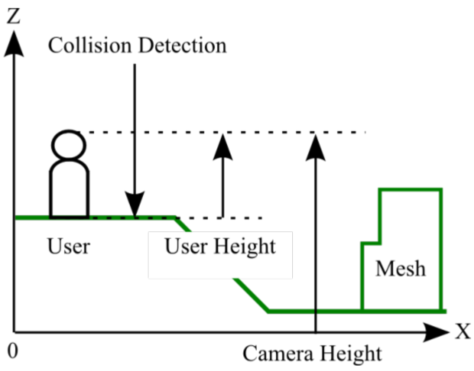

Fig. 5: Conversion of altitude

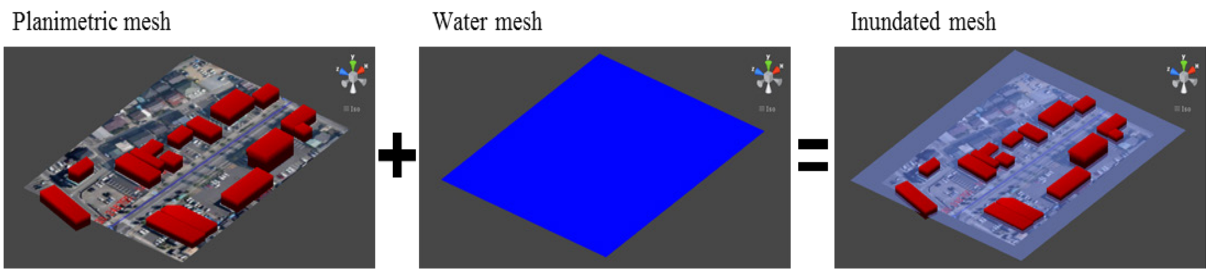

Fig. 6: Composition of planimetric mesh and wave mesh

A viewpoint is located in virtual space after the coordinates are converted as well as the feature objects. The position of this viewpoint is constantly updated to match the movement of the user in real space. In addition, its orientation is updated constantly based on the various sensors of the smartphone. Thus, the pointing direction of the smartphone camera in real space and the direction of the viewpoint in virtual space are matched. Figure 5 shows conversion of the height of the viewpoint. First, we calculate the height of the viewpoint by performing collision detection for a feature mesh on the axial height of the current position. Subsequently, we calculate the height of the viewpoint by adding a user-specified height to mesh height.

The application then places the virtual wave model in virtual space. The wave model is a simple plane mesh, as shown in Fig. 6. It expresses the simulated flood height by moving the wave model at an altitude specified by the user. By combining the wave mesh and feature mesh as shown in Fig. 6, the application can create a flood image clipped by real objects. These models are used in the occlusion process to create the final AR images.

4.3 Occlusion process and rendering

The application creates AR images by using video images and a constantly updated virtual space image. A rendered image (e.g., Fig. 7) can be obtained by rendering the flood model from the viewpoint. In this image, the wave model as virtual object is clipped by the feature mesh. This result is the same as that of virtual objects

Table 1: Coordinate conversion

WGS	Japan PRCS	Unity
Latitude	X	Z
Longitude	Y	X
Altitude	-	Y

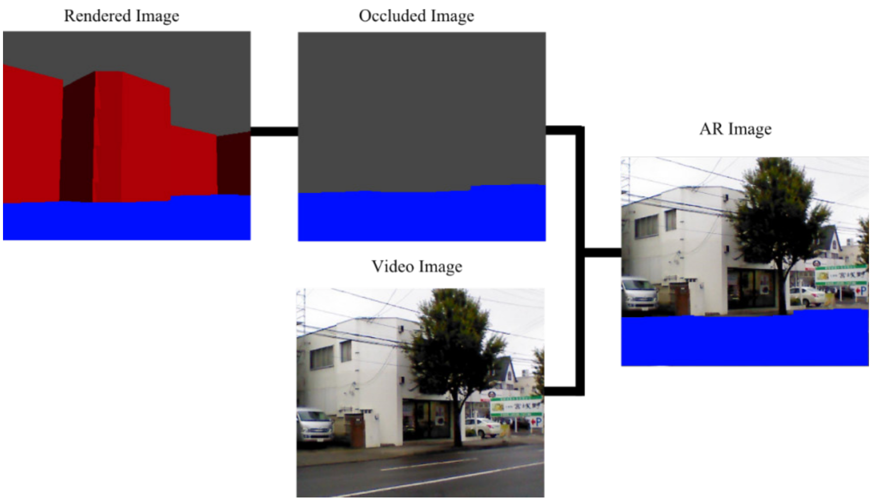

Fig. 7: Superimposition of occlusion image on video image

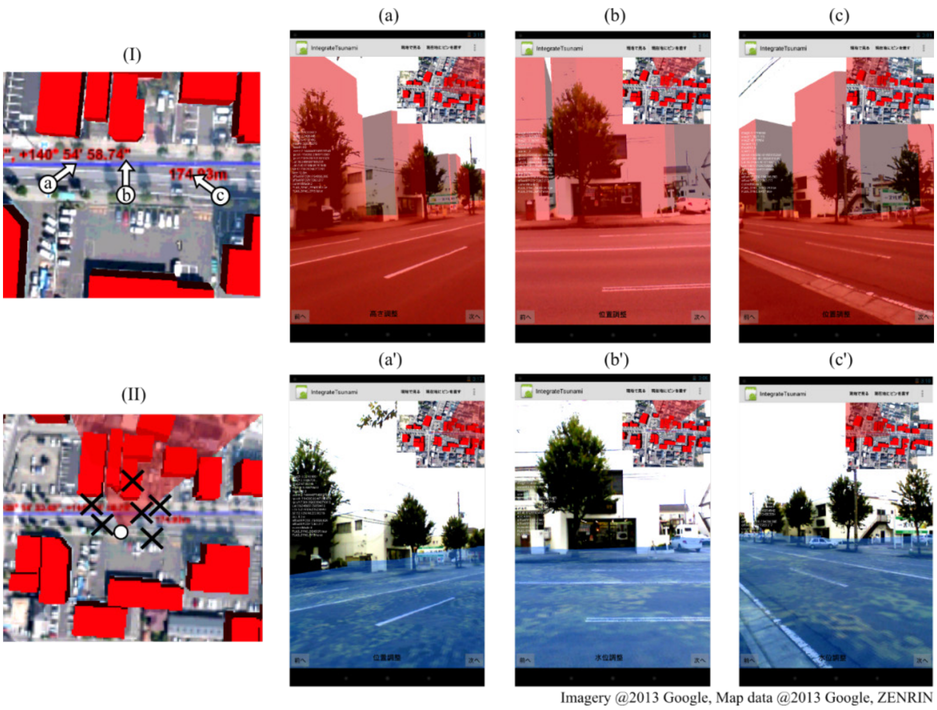

Fig. 8: Application images

occluded by objects in real space. The occluded image is obtained by extracting a portion of this wave model.

We created an occlusion shader by using OpenGL shader to mask the feature model on the game engine. The pixels that draw the models by using the occlusion shader copy the background video image instead of drawing its model. By specifying the occlusion shader to apply to the feature models, an AR image masked by feature models can be obtained, as shown in Fig. 7.

4.4 Validation and results

Figure 8 shows AR images of the actual application screen. Image (I) is a top-down view of the feature mesh. The arrows indicate the position and direction of the camera which took images (a), (b), and (c). In image (a), (b), and (c) the real objects to clip the virtual objects are drawn as transparent meshes. Image (a'), (b'), and (c') are final AR images with superimposed waves clipped by buildings. Figures represent the virtual objects moving in response to movement of the camera position.

As a result, position misalignment has occurred. Image (II) shows an example of misalignment. The cross marks indicate the positions that are obtained by GPS while taking AR image (b). The circle mark indicates the actual position where the AR image was taken. Also, misalignment of camera orientation has occurred owing to attitude sensors. Figure 9(a) represents an example of misalignment of camera orientation. In order to solve these misalignments, a function of on-site manual adjusting is added to the application. Figure 9(b) shows an example image after adjusting the position. An automatic adjusting method should be developed in the future work.

5. CONCLUSION

In this paper, we described a method of occlusion processing by using existing spatial information to register on the smartphone. We also developed a tsunami simulation to evaluate this method. As a result, an application involving use of the occlusion process operated at a speed tolerable even on smartphones. We confirmed that the advantage of this method is its ability to handle occlusion in AR by using only a smartphone to eliminate the need to provide additional hardware (e.g., an RGB-D camera). Furthermore, the method could be used while moving by constant updating of the location information in GPS.

One problem was that the method resulted in impairing the AR for occlusion handling of dynamic objects such as people or vehicles. Furthermore, occlusion handling of small objects such as trees and utility poles was not performed. A possible solution to those problems is to uses a combination of existing data for large objects and distance sensor data for small objects.

Problems in understanding occurred because the tsunami application provided only AR images in the first-person view. This hampers the user's ability to understand their situation (e.g., flood level, position). Continued investigations on information presentation methods that are easier to express in AR images (e.g., using VR as well as AR to provide information on this issue) will be necessary.

We entered 3D model data in the application to enable reading on a trial basis in this study. In future work, we will continue to improve this application so that is can be used anywhere by obtaining the model data dynamically through a network. In addition, consideration of how existing model data are used and stored for this application is necessary.

6. REFERENCES

Meng, M. (2012), Augmented Reality Magic Mirror using the Kinect. Technische Universität München. http://campar.in.tum.de/Chair/ProjectKinectMagicMirror, accessed May 05, 2013.

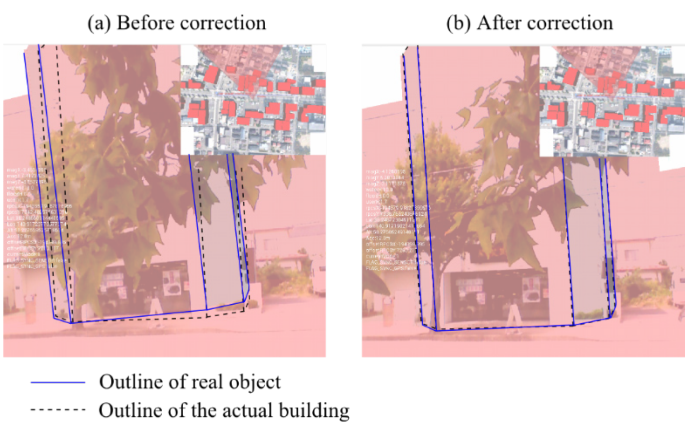

Fig. 9: Correction of camera orientation

Kakuta T., Oishi T., Ono S., Ikeuchi K. (2007). Virtual Asukakyo: A restoration of an archaeological site with mixed reality technology and expansion into a tour guide system. Monthly journal of the Institute of Industrial Science, University of Tokyo, vol. 59, no. 3, 172-175.

Suyang D., Kamat. V. R. (2010). Resolving Incorrect Visual Occlusion in Outdoor Augmented Reality Using TOF Camera and OpenGL Frame Buffer. Proceedings of CONVR2010, 55-63.

Azuma, R. T. (1997). A Survey of Augmented Reality. In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), 355-385.

Azuma, R.T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, L. (2001). Recent Advances in Augmented Reality.

7. ACKNOWLEDGEMENT

This research was partially supported by JSPS KAKENHI Grant Number 24560569. The authors sincerely thanks for Dr. Hidekazu Masaki, Mr. Koji Mizukami, and Mr. Kinya Tanaka of KOKUSAI KOGYO Co., Ltd.