International Journal of Advanced Smart Convergence Vol.6 No.4 80-87 (2017)
http://dx.doi.org/10.7236/IJASC.2017.6.4.12

IJASC 17-4-12

A Study on Virtual Studio Application using Microsoft Hololens

Jaehyun Lee[1], Seunghyeon Kim[1], Lyounghui Kim[2], Jinwook Kang[3], Seunghyun Lee[2], Soonchul Kwon[2†]

[1]Graduate School of Information and Contents, Kwangwoon University, Seoul, Korea
[2†]Department of Plasma Bio Display, Kwangwoon University, Seoul, Korea
[3]Geminisoft Co., Ltd., Korea
e-mail : {[1]noa6142, [1]eifsksj1, [2]gogorhyung}@gmail.com, jay@gemini-soft.com, {[2]shlee, [2†]ksc0226}@kw.ac.kr

Abstract

Mixed Reality (MR) shows a composite image of a virtual object in the real world. It has been applied to various fields by the introduction of head mounted display (HMD) such as Microsoft's Hololens [1-3]. The virtual studio in broadcasting combines the contents created by computer graphics with the actual set to reproduce the 3D image screen. This requires physical space such as a set of chroma keys. It also requires professional knowledge and manpower and costly equipment to post-process the graphics and information for long periods of time. Therefore, in this paper, we aim to study the implementation of virtual studio based on Mixed Reality using Microsoft Hololens. Through the implementation of 'Holo-studio' application, realistic and virtual objects of broadcasting camera viewpoint were acquired at the same time. Using Microsoft's spectator view library, the frame rate is degraded in objects with high polygons (100,000 polygons). The proposed method maintains 60 fps image transmission in high polygon objects. The results of this paper show the possibility of using virtual studio at low cost which does not need separate physical space.

Keywords: *Mixed Reality, Augmented Reality, Virtual Studio, Hololens, Head Mounted Display.*

1. Introduction

Virtual Studio is a synthesis of a virtual set created by computer graphics on a real set to reproduce a 3D image screen. Virtual studio overcomes the problems of broadcast production environment on cost and time. Then, the program provides the viewer with a high-quality graphic effect. Virtual studio uses chroma key synthesis technique, 3D graphics and camera tracking technology [4]. The chroma key synthesis technique requires a specific set such as a blue screen. Various methods such as depth keying have been proposed to overcome these physical space constraints. [5-9] Mixed Reality (MR) is a technique to synthesize a virtual object in the real world seen by the user [1]. This is the same effect as a virtual studio. Since no separate physical space is required, it can be implemented in an outdoor environment. It also has the advantage of allowing users to interact in real time. Therefore, in this paper, we aim to implement virtual studio function based on Mixed Reality using Microsoft's Hololens.

Manuscript Received: Nov. 15, 2017 / Revised: Nov. 27, 2017 / Accepted: Dec. 10, 2017
Corresponding Author: ksc0226@kw.ac.kr
Tel: +82-2-940-8637, Fax: +82-50-4174-3258
Graduate School of Information and Contents, Kwangwoon University

The composition of this paper is as follows. Chapter 2 explains Hololens, and Chapter 3 describes the proposed system. Section 4 shows the performance evaluation results of the proposed system. Section 5 concludes the paper.

2. Microsoft Hololens

Microsoft's Hololens is a typical HMD (Head Mounted Display) that implements mixed reality. The user wears an HMD and displays virtual information in the real world [10][11]. This chapter introduces Microsoft's Hololens composition and development libraries.

2.1 Components

Hololens is an all-in-one device that provides virtual information to the real world. This includes accelerometers, gyroscopes, and earth-based sensors. It is composed of 4 environment cameras, 1 RGB camera and 1 Depth camera. Figure 1 shows the composition of Hololens. The Depth camera scans the space using an infrared light and a Tof sensor. It obtains the 3D modeling information value for the surrounding environment. Environment cameras recognize environmental changes in real time. Based on this, physical phenomena such as gravity and collision of virtual objects are applied. Therefore, it is possible to accurately position a virtual object in a surrounding environment such as a floor or a desk. Hololens do not have a separate controller like a keyboard or a mouse, but they use their gaze and gestures. Users use eye movements to target objects in mixed reality. When a user sees the object they want to interact with, a small circle called Gaze moves along the line of sight [12][13].

Figure 1. Microsoft Hololens Head Mounted Display Components

2.2 Development Library

Microsoft provides a development library for Hololens [14]. The Holotool kit includes libraries for development, including input, sharing, and spatial mapping. Mixed reality companion kit includes Holographic Remoting Host, Kinect IPD, Mixed Remote View, Compositor, Spectator View and Windows Mixed Reality Commander.

1) Holographic Remoting Host enables real-time communication between Hololens and Unity and sets object position and size value.

2) The Kinect IPD measures the user's IPD (inter-pupillary distance) and sets the IPD automatically on

the Hololens device.

3) The Mixed Remote View Compositor (MRVC) is a live streaming project that outputs images viewed through Hololens. There are two types of 'Low Latency MRC' (720p) and 'MRVC' (1080p).

4) Windows Mixed Reality Commander has the function of device portal and controls many Hololens.

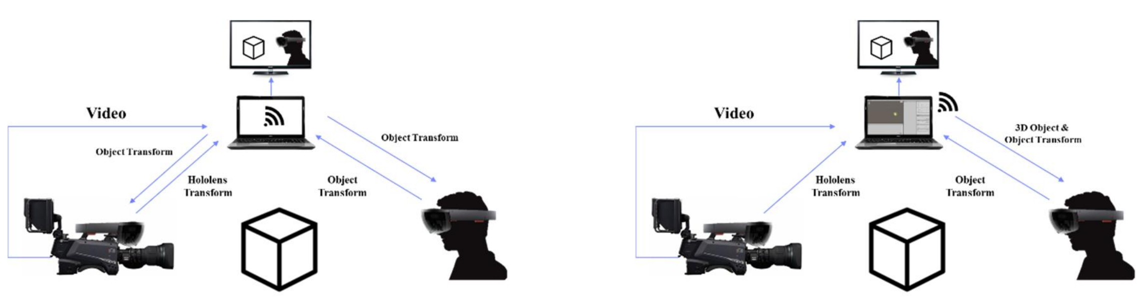
5) Spectator View is a project to output 3-point viewpoint by linking camera and Hololens.

3. Proposed system

The virtual studio implementation using Hololens utilizes Microsoft Hololens and broadcast cameras. It acquires the real world and virtual objects through a broadcast camera. The Microsoft library 'Spectator view' links the camera and Hololens to output images at the camera viewpoint. This is a constraint that the frame rate is degraded for high polygon virtual objects. Therefore, the proposed system aims to implement stable virtual studio. To do this, we develop an optimized system by developing 'Holo-studio' application.

Two Hololens and PCs with the 'Holo-studio' application communicate with each other via Wi-Fi. One Hololens is 'Hololens for user' which is worn by the user and the other is 'Hololens for broadcasting' which is calibrated with a broadcasting camera. 3D object file is imported by 'Holo-studio' application. It shares information about the position, size, and orientation of the 3D object with the connected 'Hololens for user'. 'Hololens for user' receives 3D object from PC and outputs it. The user can then control the 3D object in real time. When the position, size, and direction of the 3D object are changed by the user, the defective information is transmitted to the PC.

In the proposed system, 'Hololens for broadcasting' transmits only transform values to PC. Based on the transmitted transform value, a virtual camera is created on 'Holo-studio'. The final output video is a combination of 3D objects and video signals coming from broadcasting cameras. This is based on the position value received from 'Hololens for broadcasting'. In addition, the information values changed by 'Hololens for user' are applied in real time. The final video can be output via HD-SDI and HDMI. Figure 2 shows a comparison between the Microsoft library 'Spectator view' and the proposed system 'Holo-studio'.



(a) (b)

Figure 2. Comparison of System Configuration
(a) Spectator view (b) Proposed system (Holo-studio)

3.1 Experiment environment

Figure 3 shows Holo-studio execution flow chart. The first step is to combine the broadcast camera with the Hololens and calibrate them. The Holo-studio project consists of three scenes: 'Hololens for user', 'Hololens for broadcasting' and 'Holo-studio application'. The 'Hololens for user' scene is configured to receive object information from the PC. The scene of 'Hololens for broadcasting' is composed of different scenes because only Hololens' position value is sent to PC. The scene of 'Holo-studio application' runs the main UI of Holo-studio. As a next step, each scene is built on Hololens and PC. Next, the Sharing service connects the PC to the Hololens. The PC broadcasts virtual object information to connected Hololens for user except Hololens for broadcasting. In 'Hololens for user', the transform value of the object changed by the user is transmitted to the PC in real time.

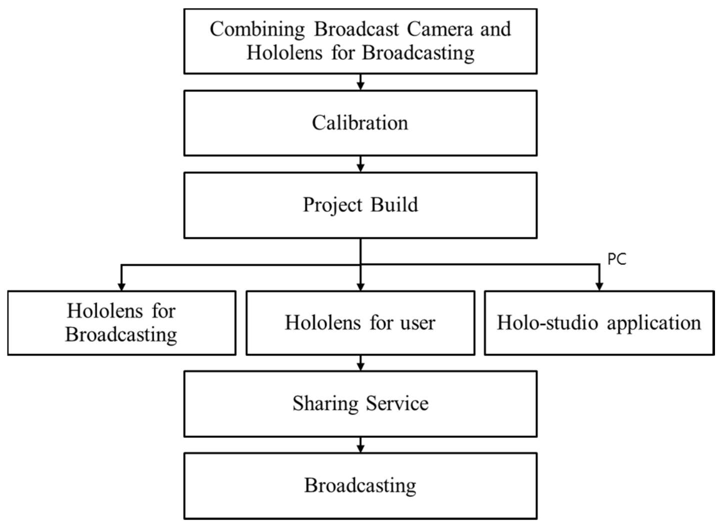


Figure 3. Execution Flowchart of Holo-studio

In this experiment, 5G Wi-Fi internal network was used for stable network environment. The camera used was set at 1080p at 60 fps and the capture board used black magic deck link 4K. Table 1 shows the PC performance used in the experiment.

Table 1. Experiment Specifications of PC

Classification	Specification
CPU	i7-7700
Mainboard	ASUS Z270-A
RAM	S/S 17000 64GB
HDD	S/T 2TB
SSD	intel 600P 256G
Graphic card	MSI GTX1080ti 8G Founders Edition

3.2 Holo-studio UI

The application UI consists of ① Game View, ② Project, ③ Hierarchy, and ④ Inspector area. The 'Game View' window is a space for viewing virtual objects and shows the objects that are actually displayed in Hololens. In the 'Game View' window, the position of the virtual space is adjusted, and the position of the connected Hololens and the position of the virtual object can be adjusted. In the 'Game View' window, 'Virtual camera' receives the transform information value from 'Hololens for Broadcasting' attached to the broadcasting camera and synchronizes the camera position. It has two display modes, 2D and 3D. The 2D display shows the currently displayed spectator view window, and the 3D display shows the overall bird's-eye view. The 'Project' window is where the 3D modeling file is imported. The 'Hierarchy' window is a space that shows the list of objects expressed in 'Game View' and is expressed as 'object name (Clone)'. The objects selected in the 'Hierarchy' window have their own Inspector values. The 'Inspector' window is set for the object's Transform, Collider Type, and Interaction. Figure 4 shows the Holo-studio UI.

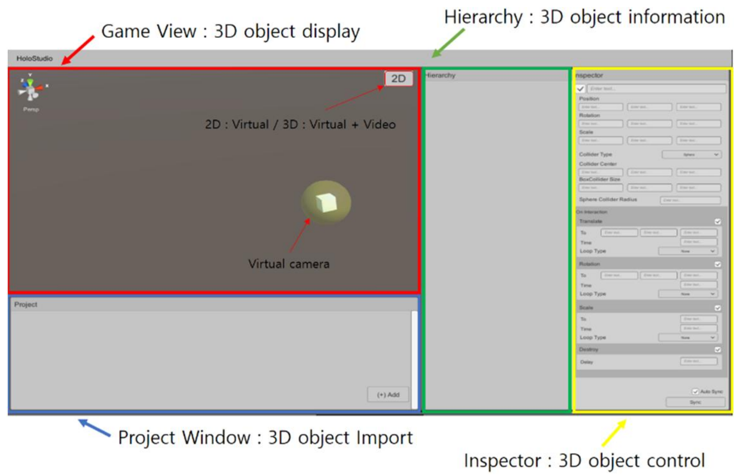


Figure 4. Holo-studio UI

3.3 Holo-studio function

The functions related to the project are made in the 'Inspector' window and classified as Transform, Collider Type, Interaction. Transforms can be divided into Position, Rotation, and Scale. Position is the position value of the object, Rotation is the rotation value, and Scale means the magnitude value. In addition to being able to adjust the object's Transform value numerically in the Inspector window, it can be adjusted using Q (select), W (position value), E (rotation value), R (size value) in the 'Game View'. Collider means 'Collision object' which can detect collision. Collider types have two forms, 'Sphere' and 'Box', which are resized with the Collider Center. Interaction is the ability to add specific interactions to objects, and the types of interactions are Translate, Rotation, Scale, Destroy. Translate is the setting for how many (m) moves in seconds on the X, Y, and Z axes. Rotation is the setting for how many degrees (degree) of rotation in several seconds on a specific axis in X, Y, Z axis. Scale is how many times the number of times to zoom in and out. Destroy is the setting for whether the object should disappear after a few seconds.

Elements that change in real time after Hololens and PC are connected to the Sharing service can be updated via the Sync button at the bottom of the 'Inspector' window. When Auto Sync is selected, real time update is performed. In 'Game View' 3D mode, real-time subtitle input / output is possible, and output subtitles are displayed only on the final monitor. The subtitles are edited in real time, the update button is applied to the output, and the submit button is applied after the modification. Figure 5 shows an example of a virtual studio application screen implemented with Holo-studio.

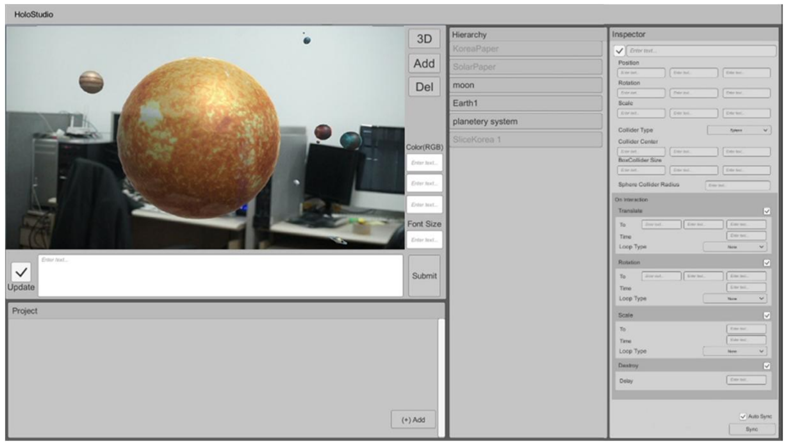


Figure 5. Implementation example of Holo-studio

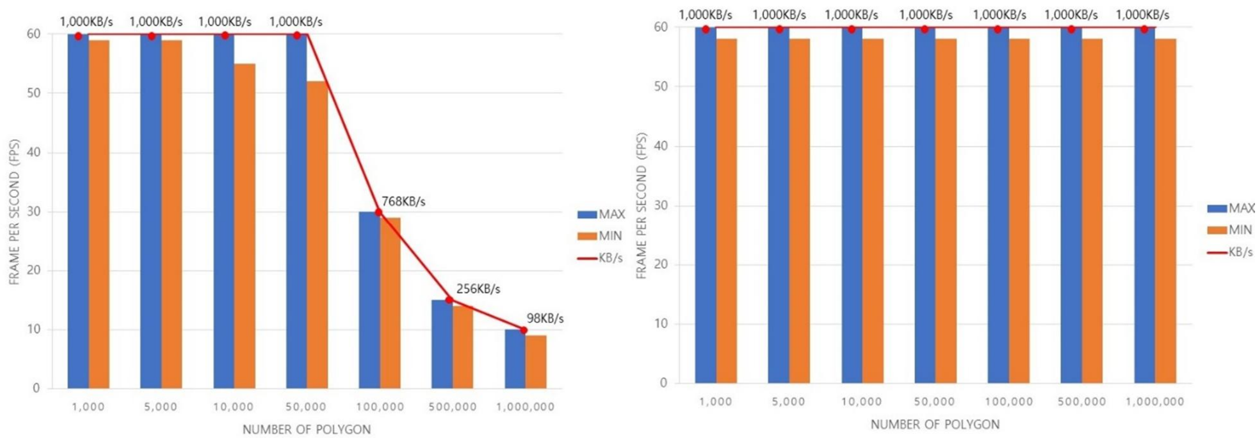
4. Performance Evaluation

Microsoft Hololens has a frame rate of 60 fps. As the number of polygons increases, the number of frames per second decreases. At this time, the video is interrupted and the color and shape of the object are distorted. Table 2 shows the maximum / minimum number of frames per second according to the number of polygons in Microsoft Hololens.

Table 2. Result of Hololens polygon test

Polygon	Frame per second (Maximum)	Frame per second (Minimum)
1,000	60fps	59fps
5,000	60fps	59fps
10,000	60fps	56fps
50,000	60fps	55fps
100,000	30fps	29fps
500,000	15fps	13fps
1,000,000	10fps	9fps

Both the Microsoft library 'Spectator view' and the proposed 'Holo-studio' receive Transform values from 'Hololens for Broadcasting' to the PC. PC receives location value of Hololens in real time via Wi-Fi. At this time, output video is interrupted according to the network transmission status. The network data transmission state depends on the Wi-Fi state and is affected by the number of frames per second. In this paper, the performance of the proposed method is verified by comparing the number of frames per second and the data transmission state according to the number of polygons. In the experiment, the differences according to the number of polygons between 1,000 and 100,000 were compared. Figure 5 compares the number of frames per second and the network data transmission (kilobyte per second) according to the number of polygons in the Microsoft library 'Spectator view' and the proposed 'Holo-studio'.



(a) (b)

Figure 6. Performance Comparison (a) Spectator view (b) Proposed system (Holo-studio)

Figure 6 (a) shows the experimental results in the Microsoft library 'Spectator view'. When the number of polygons exceeds 100,000, the number of frames per second is reduced to 30 fps or less. If the number of polygons is more than 500,000, it decreases to 15 fps or less. In addition, the number of frames per second in network data transmission is reduced to 765 KB / s and 254 KB / s, while the number of frames per second is decreasing. When the number of Hololens frames per second is reduced, the virtual objects seen by the user appear to be broken and colors are separated. The main reason is that 'Hololens for broadcasting' is set to output the virtual object transmitted from the PC, so the transmission amount is limited. Specifically, when the virtual object transform value is changed by the user Hololens, the changed value is received from the PC. Therefore, when the number of polygons exceeds 50,000, the number of frames per second decreases to 30 fps, and the number of kilobytes per second gradually decreases. Due to the decrease in the number of kilobytes per second, when the amount of transforms transmitted from 'Hololens for broadcasting' to the PC decreases, the final output video of the spectator view appears to be disconnected.

Figure 6 (b) shows the experimental results in the proposed 'Holo-studio'. In Holo-studio, 'Hololens for broadcasting' does not output virtual objects on Hololens. It does not receive the virtual object and the changed position information value broadcast from the PC. Hololens is responsible for delivering only its location information to the PC. Therefore, even in an object implementation environment having 100,000 polygons, the frame rate per second can be maintained at a maximum of 60 fps. The maximum number of kilobytes per second in network data transmission can be maintained from 800 KB/s to 1,000 KB/s. This reduces stuttering and produces stable video output.

5. Conclusion

Mixed reality can be applied to various fields with realistic and mixed reality information. In particular, domestic and foreign broadcasters are using VR / AR contents and new technology to show broadcasting contents. Experimental, immersive, and augmented reality technologies for realistic images are widely demonstrated, showing mixed reality in broadcasting contents. In this paper, we show that the proposed mixed reality system can be applied to virtual studio space with limited space and equipment. We implemented a system with virtual studio function by connecting Hololens to a camera that outputs suitable video for broadcasting. The proposed 'Holo-studio' application showed the ability to import objects and edit in real time by using various interactions such as location, size and direction. In addition, this is video export to broadcast screen with view point of broadcasting camera. This shows the possibility of replacing existing chroma key sets with virtual studios.

Acknowledgement

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-R0992-16-1008) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] Mixed Reality, *https://en.wikipedia.org/wiki/Mixed_reality.*

[2] Tepper, Oren M., Rudy, Hayeem L., Lefkowitz, Aaron,Weimer, Katie A, Marks, Shelby M, Stern, Carrie S., Garfein, and Evan S., " Mixed Reality with HoloLens: Where Virtual Reality Meets Augmented Reality in the Operating Room," Journal of the American society of plastic Surgeons, Vol. 140, No. 5, pp. 1066-1070, 2017.

[3] Ryan P. Spicer, Stephen M. Russell, and Evan Suma Rosenberg, "The mixed reality of things: emerging challenges for human-information interaction," Stereoscopic Displays and Virtual Reality Systems X(SPIE), Vol. 10207, 2017.

[4] Branimir Jakšić, Boris Gara, Mile Petrovic, Petar Spalevic, and Ljubomir Lazic, "Analysis of the Impact of Front and Back light on Image Compression with SPIHT Method during Realization of Chorma Key Effect in Virtual TV Studio," Journal of Applied Science Acta Polytechica Hungaica, Vol. 12, No. 2, pp. 71-88, 2015.

[5] Ronen Gvili, Amir Kaplan, Eyal Ofek and Giora Yahav, "Depth keying," Stereoscopic Displays and Virtual Reality Systems X(SPIE), Vol. 5006, pp. 554-563, 2003.

[6] Darryl Charles, Katy Pedlow, Suzanne McDonough, Ka Shek, and Therese Charles, "Close range depth sensing cameras for virtual reality based hand rehabilitation," Journal of Assistive Technologies, Vol. 8, No. 3, pp. 138-149, 2014.

[7] Kellner Falko, Schiller Ingo, Bartczak Bogumil, and Koch Reinhard, "Increasing Realism and Supporting Content Planning for Dynamic Scenes in a Mixed Reality System incorporating a Time-of-Flight Camera," Journal of Virtual Reality and Broadcasting, Vol. 7, No. 4, 2010.

[8] Andreas Pomi and Philipp Slusallek, "Interactive Ray Tracing for Virtual TV Studio Applications," Journal of Virtual Reality and Broadcasting, Vol. 2, No. 2, 2005.

[9] M. Ben-Ezra, "Segmentation with invisible keying signal", Proc. Of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 32–37, 2000.

[10] S. Karthika, P. Praveena, and M.GokilaMani "Hololens," International Journal of Computer Science and Mobile Computing (IJSCMC), Vol. 6, No. 2, pp. 41-50, 2017.

[11] Yadav Ashok kumar Chintamani, "Research Issues in Hololens," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 5, No. 5, pp. 1657-1663, 2016.

[12] Leehwan Hwang, Jaehyun Lee, Jahanzeb Hafeez, Jinwook Kang, Seunghyun Lee, and Soonchul Kwon, "A Study on Optimized Mapping Environment for Real-time Spatial Mapping of HoloLens," International Journal of Internet, Broadcasting and Communication, Vol. 9, No. 3, pp. 1-8, 2017.

[13] Reid Vassallo; Adam Rankin; Elvis C. S. Chen; Terry M. Peters, "Hologram stability evaluation for Microsoft HoloLens," Stereoscopic Displays and Virtual Reality Systems X(SPIE), Vol. 10136, 2017.

[14] Microsoft Hololens Library. *https://developer.microsoft.com/en-us/windows/mixed-reality.*