# Real-Time Integral Photography Using a Game Engine

Shohei Anraku
Kanagawa Institute of Technology
Japan
s1785004@cce.kanagawa-it.ac.jp

Fumihiko Ishiwata
Kanagawa Institute of Technology
Japan
s1323068@cce.kanagawa-it.ac.jp

Nahomi Maki
Kanagawa Institute of Technology
Japan
maki@ic.kanagawa-it.ac.jp

Toshiaki Yamanouchi
Kanagawa Institute of Technology
Japan
yama@ic.kanagawa-it.ac.jp

Kazuhisa Yanaka
Kanagawa Institute of Technology
Japan
yanaka@ic.kanagawa-it.ac.jp

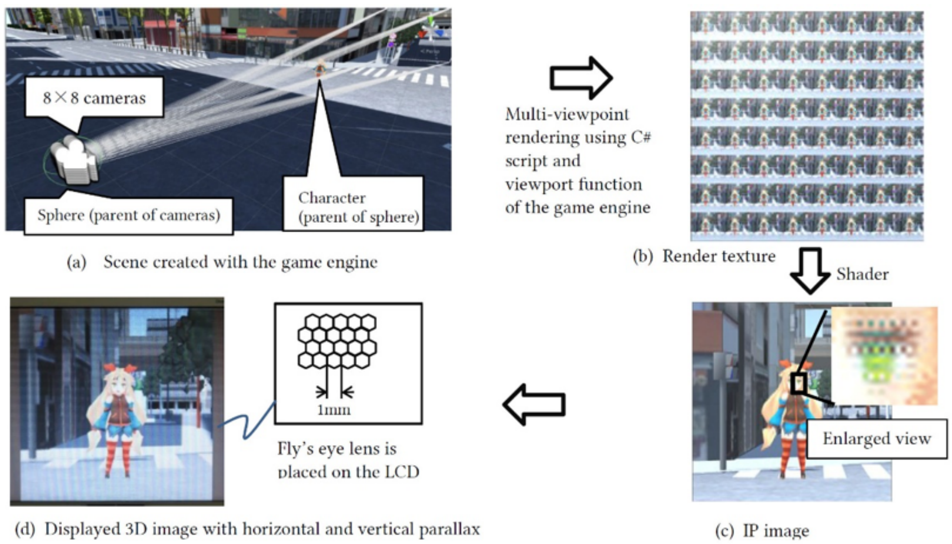


**Figure 1: Real-time synthesis of an IP image using a game engine. ©UTJ/UCL.**

## ABSTRACT

To use the advanced content creation functions of a game engine and develop contents in which displaying real-time integral photography images is important, we implemented multi-viewpoint rendering and IP image synthesis functions by adding a shader and C# scripts to the game engine.

## CCS CONCEPTS

• **Hardware → Displays and imagers**;

## KEYWORDS

Integral photography, game engine, shader


*SIGGRAPH '17 Posters, July 30 - August 03, 2017, Los Angeles, CA, USA*

ACM ISBN 978-1-4503-5015-0/17/07.
https://doi.org/10.1145/3102163.3102187

**ACM Reference format:**
Shohei Anraku, Fumihiko Ishiwata, Nahomi Maki, Toshiaki Yamanouchi, and Kazuhisa Yanaka. 2017. Real-Time Integral Photography Using a Game Engine. In *Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/10.1145/3102163.3102187

## 1 INTRODUCTION

Integral photography (IP) [Lippmann 1908] is an excellent 3D image display method because horizontal and vertical parallax are produced without the need for wearing stereo glasses. 3D images can be observed when a special image called an IP image is displayed on a liquid crystal display (LCD) and observed through a lens array called a fly's eye lens. In the past, an expensive custom-made fly's eye lens was necessary. However, this problem was solved by introducing the extended fractional view method [Yanaka 2008]. Another problem is the difficulty in real-time rendering because the amount of processing is large. This problem is solved by using a shader, which is executed in the graphical processing units (GPUs) [Yanaka and Kimura 2013]. However, it was still difficult to create high-quality 3D content.

## 2 METHOD

Therefore, we introduced a game engine. Synthesis of an IP image requires rendering a scene from many (e.g., 8 × 8) viewpoints and combining them into one IP image. Considering that the processing must be completed within one frame time, a very high processing capability is necessary. Therefore, we developed a shader for the game engine.

## 3 EXPERIMENT

Fig. 1 shows the real-time synthesis of an IP image using the game engine Unity Ver. 5.6.0 [Technologies 2017]. Using an object with as few polygons as possible is desirable to obtain a high frame rate. We constructed a scene by using Japanese Otaku City by ZENRIN Co. Ltd. [Zenrin 2017], which is licensed under a Creative Commons Attribution 4.0 International License (CC-BY). For a character walking around the scene, we used LowPolyUnityChan [2016], which is distributed by Unity Technologies Japan.

As shown in Fig. 1(a), the camera system that follows the character is made up of 8 × 8 (total of 64) cameras. Their parent object (a dummy object) is a sphere that allows all the 64 cameras to translate and rotate simultaneously. The parent object of the sphere is the character. When the character is moved forward, backward, or rotated by hitting one of the arrow keys on the keyboard, the 64 cameras follow the character.

The images obtained by the 64 cameras were transferred to a "render texture," which is a memory area, whose size is 4096 × 4096 pixels, and used for the texture mapping of a square plane. Each camera image was drawn in a different area of the render texture because different viewports were assigned to each camera, as shown in Fig. 1 (b).

Considering that a shader was assigned to the plane, an IP image was created and displayed on the surface of the plane according to the following procedure. As shown in Fig. 2, the x and y coordinates of an LCD pixel relative to the reference point of the nearest hexagonal convex lens were calculated. Subsequently, the x and y coordinates were quantized to eight levels, which served as the view numbers. Based on the view numbers, one of the 64 square areas of the render texture was selected, and the corresponding pixel value was read from the render texture and set to the pixel of the LCD. The plane was captured by the main camera of Unity, which was placed just above the plane, and the image was displayed on the LCD (Diamondcrystal WIDE RDT202WM, 20.1 inch, 1680 × 1050 pixels) of the PC (Intel Core i7-6700 and NVIDIA GeForce GTX 970).

A 3D image was created when the LCD was observed through a fly's eye lens overlaid on the screen. The frame rate was about 4 fps to 20 fps, and it varied considerably depending on the complexity of the objects, as shown in Fig. 3.

## 4 CONCLUSIONS

Using objects with a few polygons to obtain a smooth motion is desirable. If the performance of GPUs improves in the future, previous restrictions will be eliminated, and high-quality autostereoscopic 3D images will be obtained. By using the method described in this study, interactive contents, such as games developed by game engines, can be converted easily into autostereoscopic 3D contents by using the IP method. Doing so would result in substantial improvement.

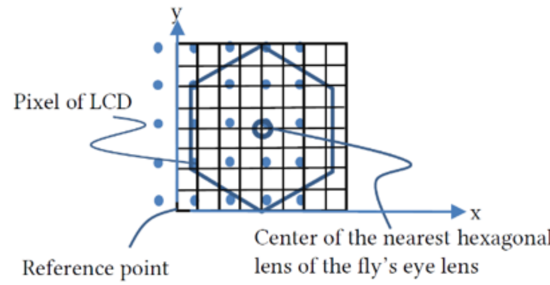


**Figure 2: Method of calculating the view number from the pixel position of the LCD.**

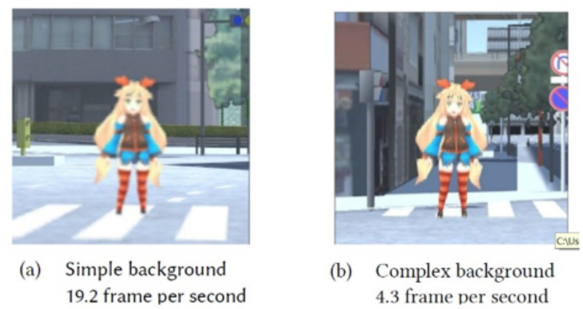


**Figure 3: Variation of frame rate due to background complexity. ©UTJ/UCL.**

## ACKNOWLEDGEMENT

This work was supported by KAKENHI Grant Number 16K00284.

## REFERENCES

Unity Technologies Japan. 2016. LowPolyUnityChan. http://unity-chan.com/download/releaseNote.php?id=LowPolyUnityChan&lang=en. (2016).

Gabriel Lippmann. 1908. Epreuves Reversibles Donnant la Sensation du Relief. *Journal of Theoretical and Applied Physics* 7, 1 (1908), 821–825. https://doi.org/10.1051/jphystap:019080070082100

Unity Technologies. 2017. Unity. https://unity3d.com/jp. (2017).

Kazuhisa Yanaka. 2008. Integral photography using hexagonal fly's eye lens and fractional view. *Proc. SPIE* 6803, 68031K–68031K–8. https://doi.org/10.1117/12.766247

Kazuhisa Yanaka and Sho Kimura. 2013. GPU Accelerated Interactive Integral Photography System Using Extended Fractional View Method. In *ACM SIGGRAPH 2013 Posters (SIGGRAPH '13)*. ACM, New York, NY, USA, Article 42, 1 pages. https://doi.org/10.1145/2503385.2503432

Zenrin 2017. Zenrin Co. Ltd. Japanese Otaku City. http://www.zenrin.co.jp/product/service/3d/asset. (2017).