

# Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

## **Real object-based integral imaging system using a depth camera and a polygon model**

Ji-Seong Jeong  
Munkh-Uchral Erdenebat  
Ki-Chul Kwon  
Byung-Muk Lim  
Ho-Wook Jang  
Nam Kim  
Kwan-Hee Yoo

# Real object-based integral imaging system using a depth camera and a polygon model

Ji-Seong Jeong,<sup>a</sup> Munkh-Uchral Erdenebat,<sup>b</sup> Ki-Chul Kwon,<sup>b</sup> Byung-Muk Lim,<sup>c</sup> Ho-Wook Jang,<sup>c</sup> Nam Kim,<sup>b</sup> and Kwan-Hee Yoo<sup>a,\*</sup>

<sup>a</sup>Chungbuk National University, Department of Computer Science, Chungbuk, Republic of Korea

<sup>b</sup>Chungbuk National University, School of Electrical Engineering and Computer Science, Chungbuk, Republic of Korea

<sup>c</sup>Chungbuk National University, Department of Digital Informatics and Convergence, Chungbuk, Republic of Korea

**Abstract.** An integral imaging system using a polygon model for a real object is proposed. After depth and color data of the real object are acquired by a depth camera, the grid of the polygon model is converted from the initially reconstructed point cloud model. The elemental image array is generated from the polygon model and directly reconstructed. The polygon model eliminates the failed picking area between the points of a point cloud model, so at least the quality of the reconstructed 3-D image is significantly improved. The theory is verified experimentally, and higher-quality images are obtained. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.OE.56.1.013110]

Keywords: integral imaging; polygon model; image quality enhancement; depth camera; three-dimensional display.

Paper 161704 received Oct. 30, 2016; accepted for publication Jan. 6, 2017; published online Jan. 31, 2017.

## 1 Introduction

Integral imaging is an autostereoscopic three-dimensional (3-D) display method that presents natural full-parallax and continuous-viewing 3-D images via common incoherent illumination. A two-dimensional elemental image array (EIA) is recorded from the object through the lens array, and the 3-D image is reconstructed from the recorded EIA.<sup>1-3</sup> However, the quality of the final 3-D image is degraded due to several problems in the optical pickup process, such as distortion and contamination of the lens array, and imperfect matching of devices. The computer-generated integral imaging (CGII) technique eliminates the optical problems of integral imaging and generates high-quality EIAs through the virtual lens array in the computer graphics, and real-time computation is also possible.<sup>4-7</sup> However, most CGII methods generate the EIAs from virtual objects, not real ones.

Recently suggested is a simplified integral imaging pickup method that is a combination of both optical pickup and CGII techniques, which displays a 3-D visualization of a real-world object through a CGII algorithm.<sup>8</sup> Here, a depth camera acquires the 3-D information of the real-world object (depth and color) at the same time, and a point cloud object space is reconstructed initially based on the acquired 3-D information. When the positions for all object points are set, the EIA is generated from the object space through a virtual lens array, i.e., a CGII algorithm, for each object point and is sent directly to the display device, while the lens array, which has the same specifications as the virtual lens array, reconstructs it as a 3-D image. By applying the image space parallel processing method using a graphics processing unit (GPU),<sup>9-14</sup> a real-time depth camera-based integral imaging display for real objects can be visualized.<sup>15</sup> Although this system eliminates the optical issues of the lens array, the failed

picking area (FPA) of the point cloud affects the final image quality. Note that an FPA is the empty area between the neighboring object points of the point cloud model, and these FPAs are visible in the generated EIA and the reconstructed 3-D image as black lines and/or portions. Also, there is another big issue in that the resolution of depth data is much lower than that of the color data, and a large amount of object information can be lost when the EIA is directly generated from the point cloud model. Several methods have been proposed to improve reconstructed image quality by enhancing the resolution of the EIA.<sup>16-19</sup>

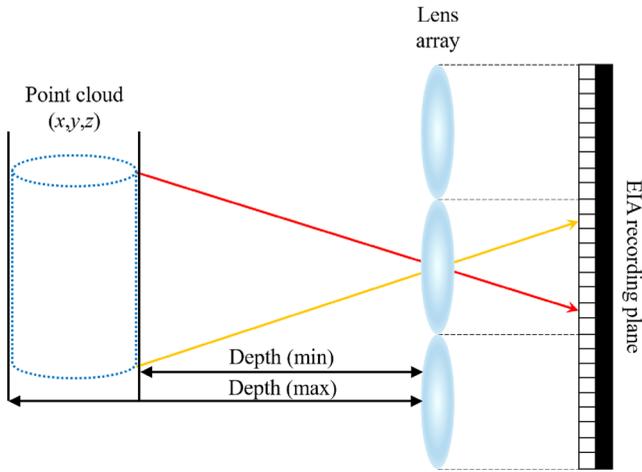
Thus, in this paper, to improve the reconstructed real 3-D image quality and keep all the information of the object, a depth camera-based integral imaging system using a polygon mesh model, which is an FPA-free 3-D model with a solid and smooth surface, is proposed. Also, the proposed method accelerates the generation time of the polygon model by utilizing a polygonal-selection CGII technique due to the complexity of the polygon generation process. In the experiment, a higher-quality 3-D image based on the newly generated polygon model is obtained.

## 2 Proposed System

### 2.1 Depth Camera-Based Integral Imaging System Using a Point Cloud Model

As mentioned in the previous section, the earlier depth camera-based integral imaging system generates EIAs from a point cloud model. When the real depth and color data of the object are acquired through a depth camera, the point cloud model is reconstructed based on the distance-coded depth information and corresponding color information of the object. The desired light rays reflected from each object point pass through the center of each virtual elemental lens and are recorded as the pixels of elemental images, according to the general integral imaging pickup process, and the

\*Address all correspondence to: Kwan-Hee Yoo, E-mail: [khyoo@chungbuk.ac.kr](mailto:khyoo@chungbuk.ac.kr)



**Fig. 1** The EIA generation process of the depth camera-based integral imaging system using a point cloud model.

corresponding color information for each object point is matched to each pixel of an EIA. Figure 1 shows the schematic configuration of the EIA generation process from the point cloud model via the depth camera-based integral imaging system.

However, during the EIA generation process for the point cloud model, some data for specific parts can be lost, and this happens often. This lost data are called the FPA, which is the empty space between the object points, and the FPAs significantly affect the reconstructed 3-D image quality.

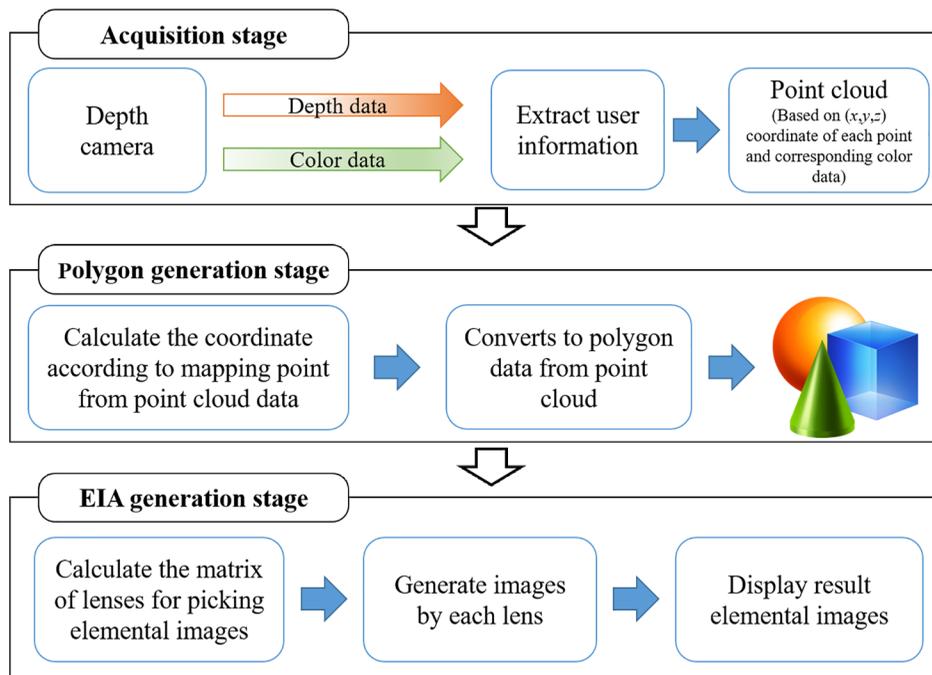
### 2.2 Depth Camera-Based Integral Imaging System Using a Polygon Model

Figure 2 shows the overall scheme of the proposed system, which consists of three main processes: acquisition, polygon generation, and EIA generation/display. Generally, the depth camera acquires 3-D depth and color information of the real

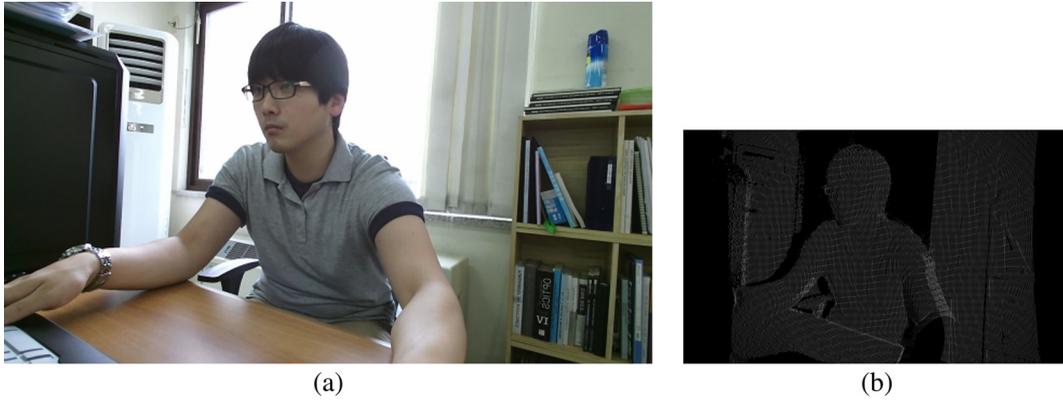
object at the same time by determining the object space for the entire depth area and the corresponding color data for the object space during acquisition. Based on the acquired data, a point cloud model that includes the real object space information is created. Then the coordinates of each pixel in the depth data and the corresponding pixel information for color data are transmitted to the next process, and the initial point cloud is converted into the polygon model by filling the empty spaces between object points, i.e., the FPAs. In the final process, the EIA is generated for the newly generated polygon model, and the 3-D image is reconstructed from the EIA.

Generally, the color information of the real object, i.e., the resolution of color sensor of the depth camera is much larger than the depth sensor. Clearly, if the EIA is generated for the initially acquired data, much of the information about the object can be lost, where the real 3-D information of the object is stored as the specific coordinates, as shown in Fig. 3.

The polygon model has a solid and smooth outer surface compared with a point cloud model, so it can be an effective solution for reconstructed image quality of the depth camera-based integral imaging system. Unlike the point cloud model, the polygon model consists of a many vertices and triangular mesh elements instead of points. First, each pixel of the depth information in Fig. 4(a) is matched up to a corresponding pixel in the color information, as shown in Fig. 4(b), where the dark circles represent the corresponded pixels (visible in both color and depth information), and the white circles represent the noncorresponded pixels (visible only in color information). Basically, if a conventional Delaunay triangulation algorithm has been applied, it would have generated the polygon model for the points with corresponding color information after the corresponded pixels are detected, as shown in Fig. 4(b).<sup>20</sup> When the number of object points is given by  $n$ , the entire process is



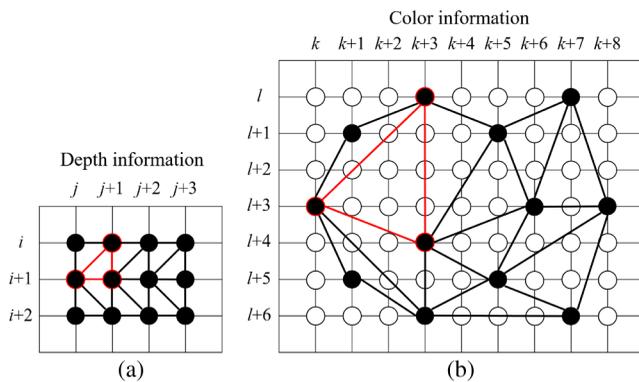
**Fig. 2** An overall procedure of the proposed method.



**Fig. 3** The resolution difference between (a) the color and (b) the depth information of a real object acquired by the depth camera.

performed in  $O(n^2 \log n)$  computational complexity. The Delaunay triangulation process requires a long processing time for a larger  $n$ . So, in this paper, a simple triangulation method is proposed, and it is performed by arranging the vertices of the polygon model in grid form directly from the depth information. The triangulation results for depth information can be preserved as they are in color information. For example, from Figs. 4(a) and 4(b), when assuming that three neighboring points of depth information  $(i, j + 1)$ ,  $(i + 1, j)$ , and  $(i + 1, j + 1)$ , which can be included in single triangle, are corresponded to  $(l, k + 3)$ ,  $(l + 3, k + 1)$ , and  $(l + 4, k + 3)$  points of color information, respectively, the triangle  $(i, j + 1)$ ,  $(i + 1, j)$ , and  $(i + 1, j + 1)$  of depth data preserves the information of the triangle  $(l, k + 3)$ ,  $(l + 3, k + 1)$ , and  $(l + 4, k + 3)$  in color information. The entire surface of the polygon model is generated using the neighboring vertical and/or horizontal vertices of the depth information, and it can save a great deal of processing time due to having a lower complexity than color information.

Assume that the nearest pixels in the depth information are determined like  $V_1$ ,  $V_2$ ,  $V_3$ , and  $V_4$  in Fig. 5. Clearly, depth information is used to generate a 3-D polygon model, and color information is used as texture mapping data on the generated 3-D polygon model. Two polygons consisting of vertices  $(V_1, V_2, V_4)$  and  $(V_2, V_3, V_4)$  and their corresponding texture coordinates are obtained; then the information is added into an array of polygons and



**Fig. 4** The principle of the grid-based 3-D polygon model generation: (a) the proposed triangulation method from neighboring points of depth information to vertices and (b) application of the simple triangulation process for color information.

texture coordinates. A set of polygons is generated from all the nearest pixels in an inputted depth image as shown in Fig. 5(a). Figure 5(b) shows the 3-D polygon model-generation process from the depth and color information acquired from the depth camera.

On generating a 3-D polygonal model, we consider the coordinates of each point of a polygon. Note that the polygon model is made up of vertices that are defined by the coordinates of the image space. We assume that  $DI_w$  and  $DI_h$  are the width and height of the color information, i.e., a color image, and  $DI_d$  is the depth information acquired by the depth sensor. Obviously, the  $(x, y, z)$  coordinates for all vertices are set to the initial values of  $DI_w$ ,  $DI_h$ , and  $DI_d$  ranges, respectively. However, these coordinates need to be converted into a new coordinate system whose origin is located at the center of object space, as shown in Fig. 6, to match to the virtual lens array and normalize the polygon model to the depth range, which a lens array can properly acquire.

The specific vertex  $(D_x, D_y, D_z)$  defined in the image coordinate system can be transformed to the new coordinate system  $O(x, y, z)$  on the virtual lens array space as follows:

$$\begin{aligned} O \left[ x = D_x \times OP - \frac{DI_w}{2}, y = D_y \times OP - \frac{DI_h}{2}, \right. \\ \left. z = \frac{2D_z z_{CDP}}{\max(DI_d) + \min(DI_d)} \right], \end{aligned}$$

$$\text{where } OP = \frac{z_{CDP} \times P_D}{g} \quad \text{and} \quad z_{CDP} = \frac{g \times f_{LA}}{g - f_{LA}}, \quad (1)$$

where  $OP$  is the distance between the virtual lens array and corresponding orthogonal plane that can occur in each depth value of object points,  $f_{LA}$  is the focal length of the virtual lens array,  $g$  is the gap between the EIA plane and the virtual lens array,  $z_{CDP}$  is the central depth plane that is the distance between the central plane of the polygon model and the virtual lens array, and  $P_D$  is a pixel pitch of the EIA plane or display device. The EIA is generated from this depth-matched polygon model, the details of which are illustrated later.

Figures 7(a) and 7(b) show the rendering results, with a point cloud representing a scene captured from a depth camera, and with the 3-D polygon model generated by the proposed algorithm, respectively. Figure 7(a) shows an example

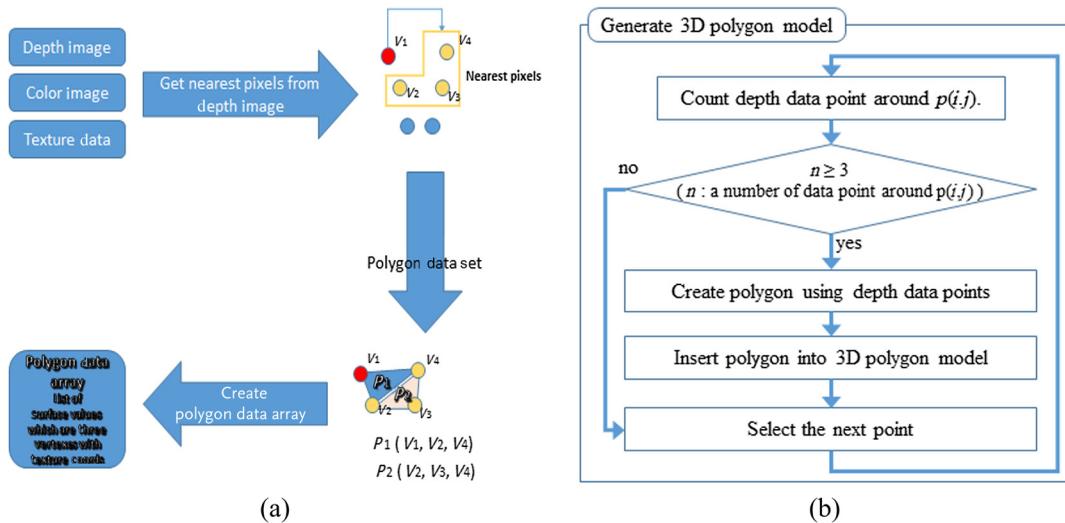


Fig. 5 (a) Triangulation from nearest points of depth information and (b) generation step for a 3-D polygon model with depth and color data.

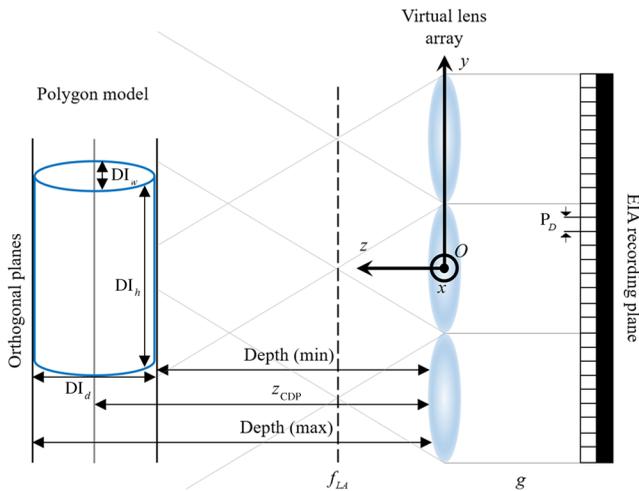


Fig. 6 Geometry of transformed coordinate system on the virtual lens array space.

of the point cloud model after color information is plated on the depth information, an enlarged detailed representation of a region marked as a yellow rectangle, and a rotated representation of the point cloud model to present it more precisely. In Fig. 7(b), the texture mapping result applied to the generated polygon model is presented.

In the EIA generation stage, the EIA is generated for the newly generated depth-matched polygon model, and a fast computation method is applied that generates and displays the entire EIA within the shortest possible time. Here, a visible image from each elemental lens with respect to the polygon model is generated via a single thread using the virtual ray-tracing method; thus, the entire EIA generation time can be reduced since all threads can work simultaneously for every elemental lens. The entire detailed process of EIA generation from the polygon model is shown in Fig. 8.

The EIA generation process consists of three substages: preprocessing, EIA generation, and display. In the preprocessing, to prepare the pickup process for elemental images, a virtual space that contains the polygon model, a virtual lens

array, and an EIA plane is built, where specifications of the virtual lens array are set by the user. The position of each elemental lens inside the virtual space is calculated as follows:

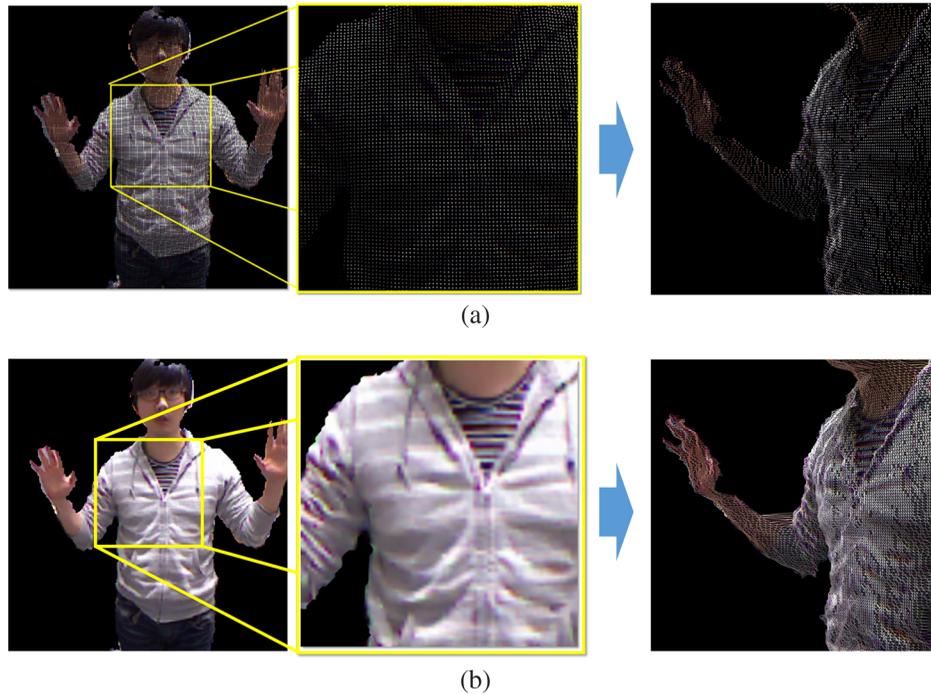
$$EL \left\{ \begin{aligned} x &= \frac{1}{2} [-LA_w + 2LP_w(i-1)], \\ y &= \frac{1}{2} [-LA_h + 2LP_h(j-1)] \end{aligned} \right\},$$

where  $EL_{fov} = \tan^{-1} \left( \frac{f_{LA}}{2g} \right)$ . (2)

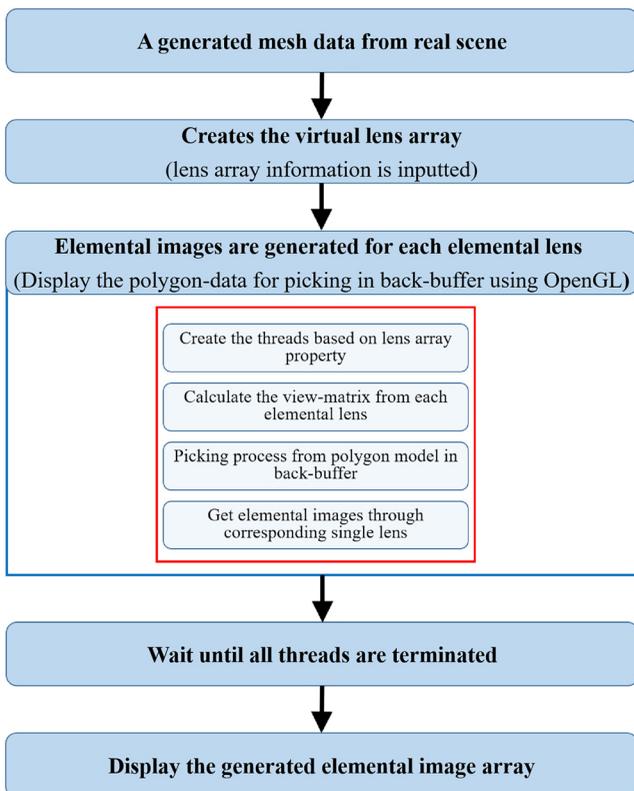
From Eq. (2),  $EL(x, y)$  is the position of each elemental lens,  $LA_w$  and  $LA_h$  are the width and height of the entire virtual lens array, and  $LP_w$  and  $LP_h$  are the width and height of any  $(i, j)$ 'th elemental lens. Equation (2) is carried out only once in the preprocessing substage of EIA generation.

The next substage is the EIA pickup that generates the elemental images based on the preprocessing stage from the polygon model. It creates the same number of threads as the number of elemental lenses, and a ray group is generated that has the same number of rays as the number of pixels of each elemental lens within each thread, as shown in Fig. 9. Each ray passes through the center of an elemental lens from the EIA plane and visualizes only the plane information intersecting with the corresponding ray, where this process is run simultaneously for every ray in every thread; at the least, the huge computation processing time is greatly reduced.

Finally, in the display substage, the generated EIA is projected on the display device, while the lens array reconstructs it as a 3-D image for the observer. An example of an EIA generated from the polygon model is shown in Fig. 10. Here, it can be seen that the FPAs are invisible in the EIA generated from the polygon model, as shown in Fig. 10(b), whereas too many FPAs are visible in the point cloud-based EIA, as shown in Fig. 10(a).



**Fig. 7** Rendering results difference between (a) a point cloud and (b) a polygon model generated by the proposed algorithm.

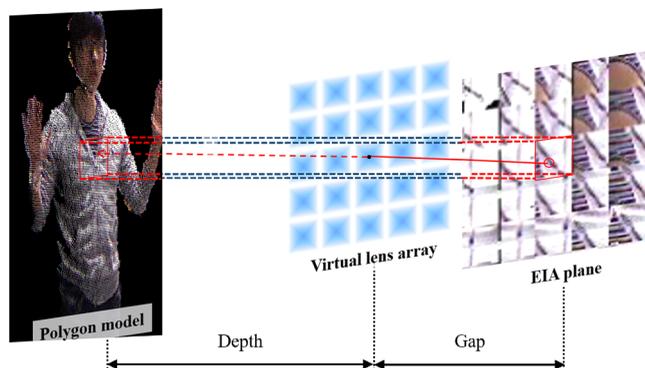


**Fig. 8** EIA generation process for a newly generated polygon model.

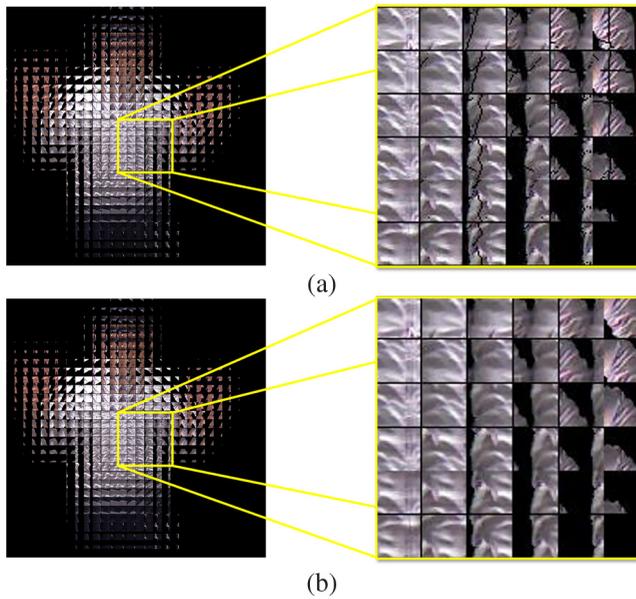
### 3 Experimental Results

The specifications of the experimental object and devices are listed in Table 1, and the experimental environment is presented in Fig. 11.

The depth camera acquires depth information with  $512 \times 424$  pixels, and color information with  $1920 \times 1080$  pixels, and the polygon model consisting of 217,088 vertices (0.2 MB) is generated based on the acquired data. We prepared three kinds of color images, from simple to complicated, to provide the experimental results of the proposed method, as shown in Fig. 12(a). The EIAs generated from the initial point cloud and the newly generated polygon models are presented in Figs. 12(b) and 12(c), respectively. The proposed method generates a  $28 \times 28$  elemental image for each lens and generates the entire EIA resolution at  $840 \times 840$  pixels by considering the lens array size, where the entire size of lens array is  $150 \times 150 \text{ mm}^2$ . Compared with the EIAs generated directly from point cloud models, the EIAs for polygon models are FPA-free, which means that the EIA does not have the empty areas (i.e., black lines in the image), and the 3-D information of the real object can be exactly recorded to the EIA. Therefore, it can be verified that the polygon model has an FPA-free solid outer surface, whereas the point cloud model has too many FPAs.



**Fig. 9** The EIA generation process using the ray group.

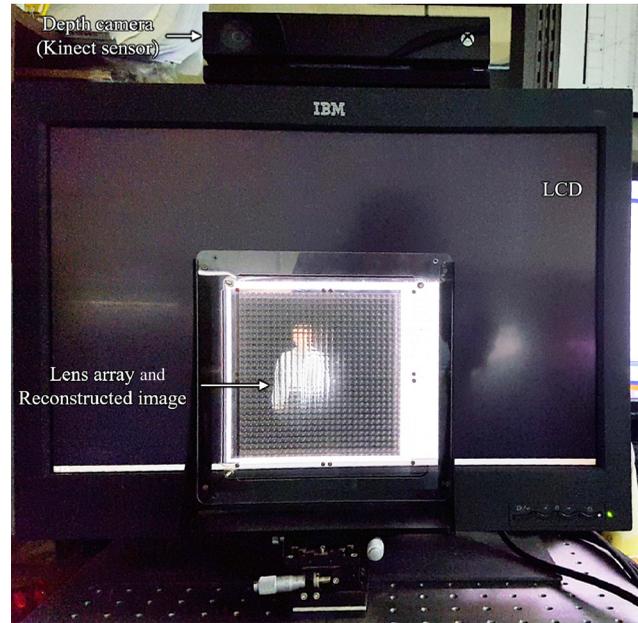


**Fig. 10** The comparison between the EIAs from point cloud and polygon models: (a) EIAs generated from the point cloud includes many FPAs and (b) for the polygon model, the EIA is generated as FPA-free.

**Table 1** Specifications of experimental devices and EIA

Key components	Specifications	Characteristics
Lens array	Focal length	10 mm
	Number of lenses	30 × 30
	Pitch of elemental lens	5 mm
	Overall size	150 × 150 mm
Depth camera (Kinect sensor)	Resolution of depth information	521 × 424 pixels
	Resolution of color information	1920 × 1080 pixels
Computer	Performance	CPU: Intel Core i7-4770 3.4 GHz
		RAM: 12 GB
		GPU: NVIDIA GeForce GTX 780 (Core: 2304)
Display device	Resolution of screen	3840 × 2160 pixels
	Pixel pitch of screen	0.1796 mm
EIA	Resolution of EIA	840 × 840 pixels

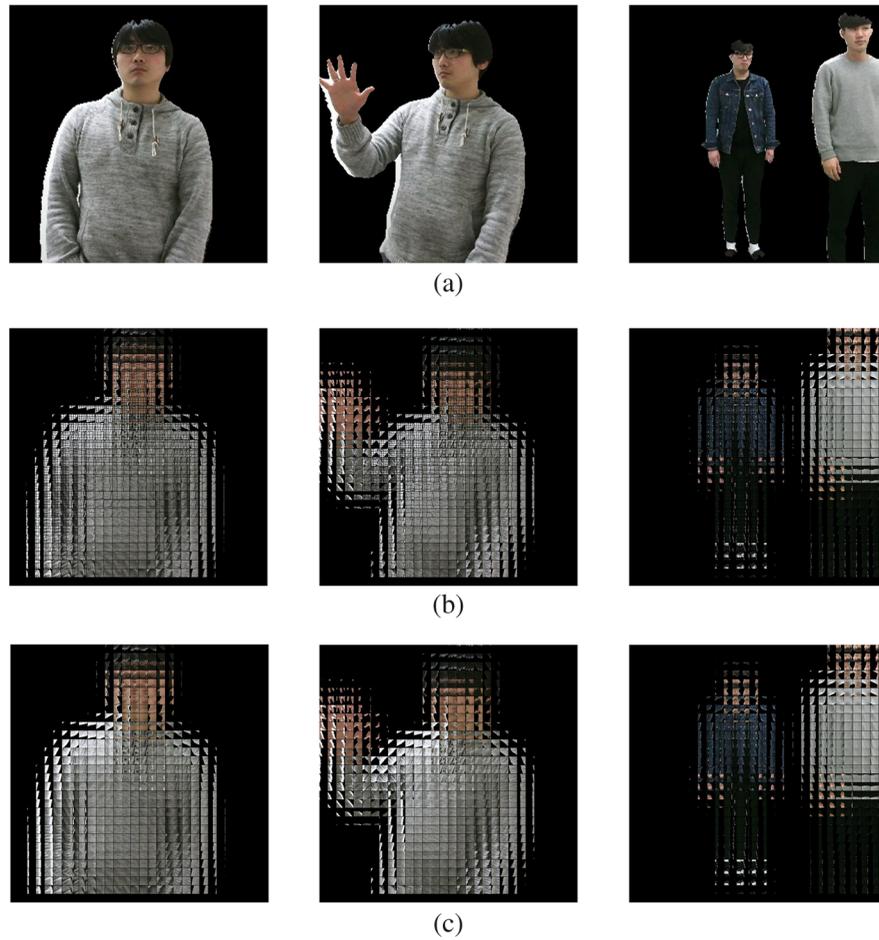
Figure 13 shows the numerically reconstructed 3-D images using the computational integral imaging reconstruction algorithm<sup>21</sup> for the generated point cloud-based and polygon-based EIAs. To present the real 3-D information of the object, the images captured from different viewpoints along the different reconstruction distances are presented. In the



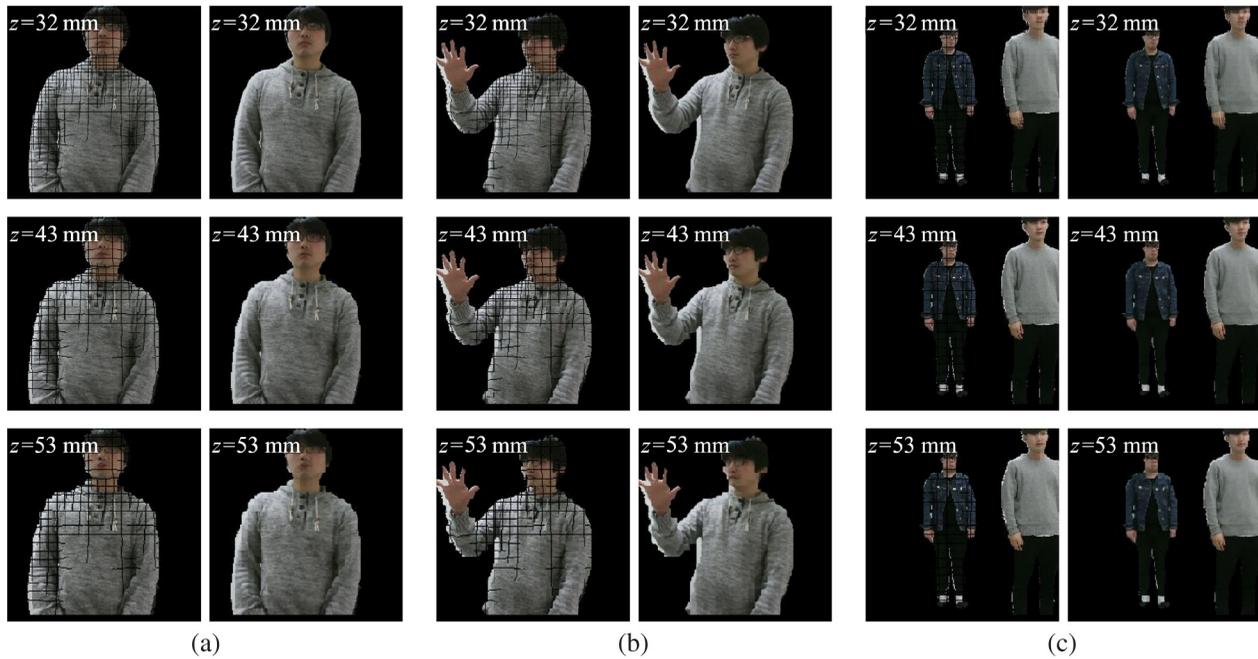
**Fig. 11** The system implementation for the proposed method.

experiments, three values (32, 43, and 53 mm) are given as the depth planes. There are 18 images that consist of three rows and six columns in Fig. 13. The images in the top, middle, and bottom rows represent the reconstructed 3-D images at 32, 43, and 53 mm reconstruction distances, respectively, for the generated EIAs. The images in the odd column and the even column represent the reconstructed 3-D images for point cloud-based EIAs and polygon-based EIAs. For example, the image in the second row and third column shows the reconstructed 3-D image generated from 43 mm for a point cloud-based EIA. Now, let us compare the quality of the reconstructed images. There are many FPAs that appear as black lines in the images in the odd column, but on the other side, there are FPA-free images in the even column. From these images, it can be verified that a quality difference exists between the images according to the depth of field. Especially for the shoulder, silhouette, and face areas, the image quality degrades when the depth of field increases.

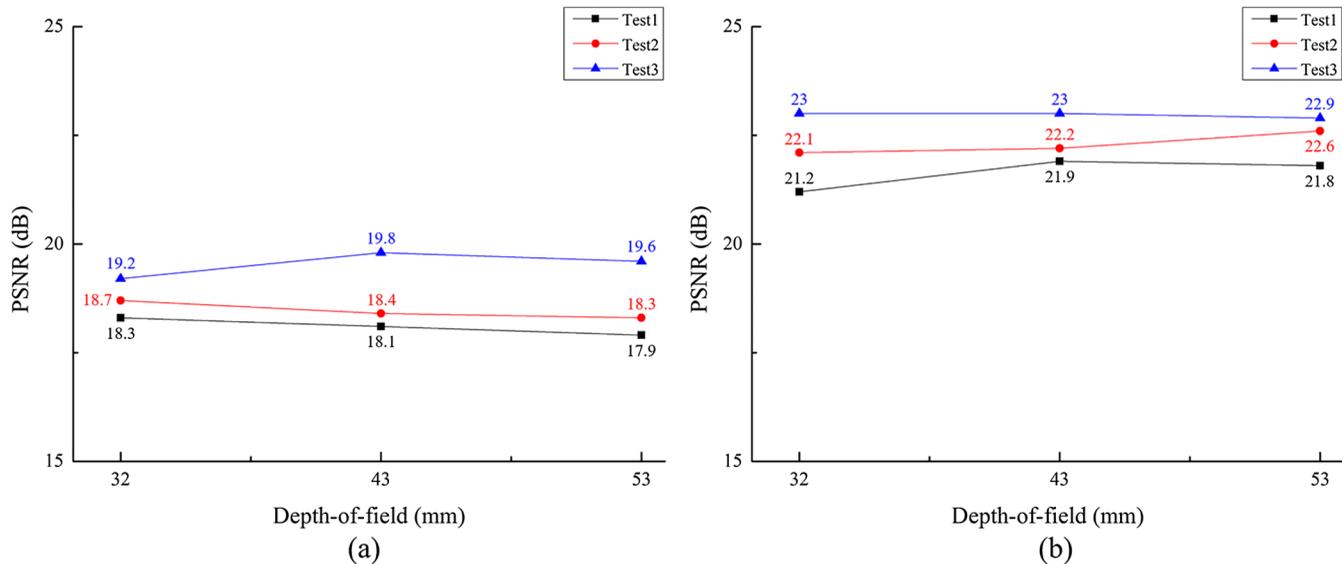
To measure the 3-D image quality and compare the previous point cloud-based system to the proposed polygon-based one, peak signal-to-noise ratio (PSNR) is utilized for the three data cases (test 1 for 32 mm, test 2 for 43 mm, and test 3 for 53 mm), as shown in Fig. 12, each of which consists of the color and depth information, the corresponding point cloud, and the polygon-based EIAs. PSNR for test 1, test 2, and test 3 are represented with blue, orange, and gray colors, respectively. The measured PSNR values for the polygon-based model and the point cloud-based model are presented in Figs. 14(a) and 14(b), respectively. Here, PSNR values are measured for different depth planes (32, 43, and 53 mm) in each reconstruction. PSNR values from three polygon-based models in Fig. 14(b) were measured at ~21.2, 22.1, and 23 dB for a 32-mm depth plane at 21.9, 22.2, and 23 dB for a 43-mm depth plane and at 21.8, 22.6, and 22.9 dB for a 53-mm depth plane; the PSNR values for point cloud-based cases in Fig. 14(a) were measured at ~18.3, 18.7, and 19.2 dB for a 32-mm depth plane, at 18.1, 18.4, and 19.8 dB for a 43-mm depth plane, and at



**Fig. 12** (a) For the real object from simple to complex, (b) the point cloud-based EIA is generated, where the FPAs are visible as black lines, and (c) the clearer EIA generated (without FPA) from the polygon model.



**Fig. 13** The reconstructed 3-D image generated on the different depth planes for given three objects: (a) basic single object, (b) complicated single object, and (c) multiple objects.



**Fig. 14** Comparison of the PSNR values for (a) point cloud-based models and (b) polygon-based models generated from three different cases that are 32, 43, and 53 mm reconstruction distances (test 1, test 2, and test 3).

17.9, 18.3, and 19.6 dB for a 53-mm depth plane. Thus, we can see that the proposed method successfully improves the reconstructed image quality of the depth camera-based integral imaging system at all the allowable depth planes when compared with the conventional case.

#### 4 Conclusions

A polygon model-based quality-enhanced integral imaging display system is proposed and implemented. The initial point-cloud model is generated based on the acquired depth and color information of real-world objects through a depth camera, and the polygon model is converted from the point cloud by applying a proposed triangulation algorithm for each object point. The final reconstructed image has better quality than a point cloud model-based method, in that the PSNR value is higher by  $\sim 0.5$  to 2.0 dB. However, the proposed method cannot satisfy a real-time display due to the huge computation time for conversion from real 3-D data to the virtual 3-D object. To provide high-speed computation, intermediate view image generation and/or GPU-based parallel processing algorithms are required because these kinds of methods are able to shorten the time needed to generate the entire image.

#### Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2014R1A1A2055379); 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 and Communications Technology Promotion); and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A2A01003934).

#### References

- G. Lippmann, "La photographie integrale," *C. R. Acad. Sci.* **146**, 446–451 (1908).
- J.-H. Park et al., "Recent progress in three-dimensional information processing based on integral imaging," *Appl. Opt.* **48**, H77–H94 (2009).
- N. Kim et al., "Advances in the light field displays based on integral imaging and holographic techniques," *Chin. Opt. Lett.* **12**(6), 060005 (2014).
- N. Kim et al., *3-D Integral Photography*, SPIE Spotlight SL18, SPIE Press, Bellingham, USA (2016).
- Y. Igarashi et al., "3D display system using a computer generated integral photography," *J. Appl. Phys.* **17**, 1683–1684 (1978).
- M. Halle, "Multiple viewpoint rendering," in *Proc. ACM (SIGGRAPH 1998)*, pp. 243–254 (1998).
- S.-W. Min, "Enhanced image mapping algorithm for computer-generated integral imaging system," *Jpn. J. Appl. Phys.* **45**, L744–L747 (2006).
- K. S. Park et al., "Viewpoint vector rendering for efficient elemental image generation," *IEICE Trans. Inf. Syst.* **E90-D**, 233–241 (2007).
- G. Li et al., "Simplified integral imaging pickup method for real objects using a depth camera," *J. Opt. Soc. Korea* **16**, 381–385 (2012).
- K.-C. Kwon et al., "High speed image space parallel processing for computer-generated integral imaging system," *Opt. Express* **20**, 732–740 (2012).
- D.-H. Kim et al., "Real-time 3D display system based on computer generated integral imaging technique using enhanced ISPP for hexagonal lens array," *Appl. Opt.* **52**, 8411–8418 (2013).
- K.-C. Kwon et al., "Resolution-enhancement for an orthographic-view image display in an integral imaging microscope system," *Biomed. Opt. Express* **6**, 736–746 (2015).
- OpenCL Programming Guide for the CUDA Architecture*, NVIDIA, California (2010).
- CUDA C Programming Guide*, NVIDIA, California (2014).
- Kinect for Windows SDK Programming Guide*, Microsoft, Washington (2016).
- J.-S. Jeong et al., "Development of a real-time integral imaging display system based on graphics processing unit parallel processing using a depth camera," *Opt. Eng.* **53**, 015103 (2014).
- Q. Zhang et al., "Integral imaging display for natural scene based on KinectFusion," *Optik* **127**, 791–794 (2016).
- H. Navarro et al., "High-resolution far-field integral-imaging camera by double snapshot," *Opt. Express* **20**, 890 (2012).
- E. A. Karabassi et al., "A fast depth-buffer-based voxelization algorithm," *J. Graph. Tools* **4**, 5–10 (1999).
- X. Wang et al., "Performance characterization of integral imaging systems based on human vision," *Appl. Opt.* **48**, 183–188 (2009).
- J. O'Rourke, *Computational Geometry in C*, 2nd ed., Cambridge University Press, Cambridge, England (1998).

**Ji-Seong Jeong** received his MS degree in computer education and his PhD in information and computer science from Chungbuk National University, Republic of Korea, in 2011 and 2015, respectively. His research interests include computer graphics, integral imaging systems, dental/medical systems, smart learning, and mobile applications.

**Munkh-Uchral Erdenebat** received his MS degree in 2011 and his PhD in 2015, respectively, in information and communication engineering from Chungbuk National University, Republic of Korea. He is the author of more than 12 journal papers and has written a professional book. His current research interests include 3-D image processing, 3-D displays, light field displays, 3-D microscopes, and holographic techniques.

**Ki-Chul Kwon** received his PhD in information and communication engineering from Chungbuk National University in 2005. Since 2008, he has been a researcher/visiting professor at BK21Plus Program in the School of Electrical Engineering and Computer Science, Chungbuk National University. His research interests include three-dimensional imaging systems, medical imaging, and computer vision.

**Byung-Muk Lim** is an MS candidate who is working for the Department of Computer Science at Chungbuk National University, Republic of Korea. He received his BS degree in computer education from Chungbuk National University in 2015. His research interests include computer graphics and 3-D digital content.

**Ho-Wook Jang** is a PhD candidate who is working for the Department of Digital Information and Convergence at Chungbuk National University, Republic of Korea, and is also a principal member of research staff who is working for the Department of Next Generation Content Research Division at Electronics and Telecommunications Research

Institute, Republic of Korea. He received his BS degree in computer engineering from Kyungpook National University, Republic of Korea, in 1986, and also received his MS degree in computer science from Korea Advanced Institute of Science and Technology, Republic of Korea, in 1988. His research interests include computer graphics, 3-D character animation, and 3-D digital content.

**Nam Kim** received his PhD in electronic engineering from Yonsei University, Seoul, Republic of Korea, in 1988. Since 1989, he has been a professor in the Department of Computer and Communication Engineering, Chungbuk National University. From 1992 to 1993, he spent a year as a visiting researcher in Dr. Goodman's Group at Stanford University. In addition, he attended Caltech as a visiting professor from 2000 to 2001. His research interests include the holographic technique, integral imaging, diffractive optics, and optical memory systems.

**Kwan-Hee Yoo** is a professor working for the Department of Computer Science at Chungbuk National University, Republic of Korea. He received his BS degree in computer science from Chonbuk National University, Republic of Korea, in 1985, and his MS and PhD degrees in computer science from Korea Advanced Institute of Science and Technology, Republic of Korea, in 1988 and 1995, respectively. His research interests include computer graphics, integral imaging systems, dental/medical systems, and smart learning.