Transformations and Their Analysis from a RGBD Image to Elemental Image Array for 3D Integral Imaging and Coding

Hoon Yoo[1*]
[1] Dept. Electronics, SangMyung University
Seoul, 110-743, KOREA
[e-mail: hunie@smu.ac.kr]
*Corresponding author: Hoon Yoo

Received July 23, 2017; revised December 31, 2017; accepted January 18, 2018; published May 31, 2018

Abstract

This paper describes transformations between elemental image arrays and a RGBD image for three-dimensional integral imaging and transmitting systems. Two transformations are introduced and analyzed in the proposed method. Normally, a RGBD image is utilized in efficient 3D data transmission although 3D imaging and display is restricted. Thus, a pixel-to-pixel mapping is required to obtain an elemental image array from a RGBD image. However, transformations and their analysis have little attention in computational integral imaging and transmission. Thus, in this paper, we introduce two different mapping methods that are called as the forward and backward mapping methods. Also, two mappings are analyzed and compared in terms of complexity and visual quality. In addition, a special condition, named as the hole-free condition in this paper, is proposed to understand the methods analytically. To verify our analysis, we carry out experiments for test images and the results indicate that the proposed methods and their analysis work in terms of the computational cost and visual quality.

Keywords: 3D imaging, computational integral imaging, elemental image array, RGBD image, 3D data coding

1. Introduction

Three-dimensional (3D) imaging and visualization techniques based on integral imaging have been paid increasing attention as various industries require 3D techniques in recent years. Thus lots of studies have been discussed in this literature [1-12]. Basically, integral imaging is a technique to record 3D objects and display 3D images in space using a lens array [1]. It can provide 3D display without eyeglasses and full parallax with color images. The nature of integral imaging can also be applied to various image systems such as depth extraction, noise elimination, 3D-TV systems [6-12].

A typical integral imaging system consists of two parts: pickup and display. The pickup part records 3D objects through a lens array, producing an elemental image array (EIA). The display part provides the auto-stereoscopic 3D display for 3D objects [2-4]. One of the important issues in integral imaging is how to obtain elemental images efficiently. Normally, pickup methods are classified into two categories: optical pickup and computational pickup. The existing optical pickup methods possibly employ a lens array, a camera array, or a moving camera [5-7]. The optical methods are usually used for real 3D objects. On the other hands, the computational pickup method is also known as computer-generated integral imaging (CGII), where elemental images are generated from 3D computer models [5-7]. This method is restrictively utilized for 3D computer objects.

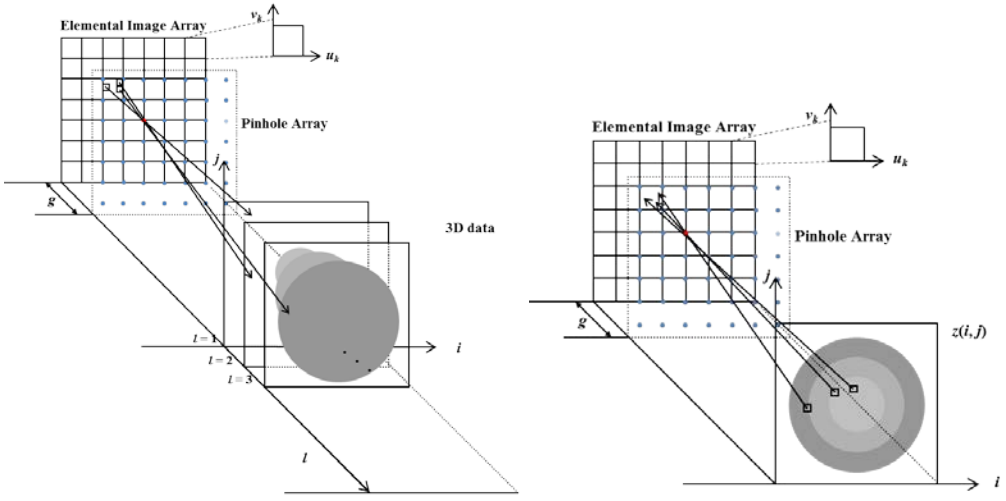
As another 3D imaging technique, a color image and corresponding depth image have been very actively discussed [13-19]. The image format employed in this technique is called as the 2D-plus-depth image or the RGBD image [13-16]. Literally, a RGBD image consists of a RGB color image and its depth map. In this paper, we use the term RGBD images for the depth-based images. The advantages of the RGBD images are that it is very efficient to deliver 3D image data [13-15] and they are obtained from low-price devices such as Kinect [17]. Thus, this image format is currently used in the most practical 3D imaging, transmitting, and display systems. In addition, the RGBD image is increasingly applied to 3D vision systems since some patterns are accurately detectable in the depth image [18-19].

However, the technique using RGBD images has the disadvantage in terms of auto-stereoscopic three-dimensional display. Thus an additional process is required to render RGBD images for 3D display. This is called as the depth-image-based rendering (DIBR) [14-16]. On the other hand, the integral imaging technique requires the elemental image array for 3D image systems. Additionally, instead of compressing EIA data directly by use of a specific coding algorithm, transforming between EIA data and RGBD data is the first step in delivering EIA very efficiently. Autostereoscopic display is possible after recovering EIA data from RGBD data. This means that EIA data is for 3D display and RGBD data is for 3D coding. Therefore, a mapping method between a RGBD image and an elemental image array is required to view the RGBD image with auto-stereoscopic three-dimensional display and to use computational integral imaging systems.

Recently, more attention has been paid to the conversion between elemental images and a RGBD image [12, 20-21]. An integral imaging technique using a camera moving off-axially provided a depth map and a color image and then used the RGBD image to generate an elemental image array for 3D visualization [12]. A RGBD sensor such as the Kinect sensor employed to obtain RGBD images which are used to be converted into an elemental image array [20-21]. Also similar studies on computer-generated integral imaging via RGBD images

have been discussed [22-23]. In addition, to generate depth maps under noisy environments, a general framework and machine learning-based framework have discussed to fuse noisy point clouds from multi-view images of the same object [24-25]. These techniques may be useful to genernate RGBD images.

In this paper, we study the generation of elemental image array from a RGBD image and also introduce two mapping methods and their analysis. Generating EIA from a RGBD image requires a mapping method. We introduce two mapping models: one is a forward mapping and the other is backward mapping. These two mapping methods might have their own problems or advantages. Thus, there needs to be an analysis method to figure out which of mappings is efficient and in what cases they can lead to better performance. Those mappings and their analysis are not discussed yet in computational integral imaging although some existing systems had utilized a mapping method [12, 20-21]. Here, the contribution of this paper are that we define forward and backward mappings for conversion between a RGBD image and EIA with a set of whole optical parameters. Also we provides their analysis in terms of the processing time and visual quality of the reconstructed 3D images. According to this analysis, we can figure out which mapping method is more preferable to apply it to a certain 3D imaging system with a specific parameter set. To verify our methods and analysis, we carry out experiments with RGBD images in terms of the computational processing time and visual quality.



(a) (b)

Fig. 1. Geometry pixel mapping from a RGBD image to an elemental image array (a) backward mapping (b) forward mapping

2. Transformations between EIA and RGBD Image

A pixel-to-pixel mapping method is required to convert a RGBD image into an elemental image array (EIA). The ray tracing model that uses a linear equation in 3D space is very useful to design a mapping method since it is very simple and accurate. Also, the ray tracing model can be implemented by two different mapping methods. In this paper, we define the two mapping methods: one is a backward mapping and the other is a forward mapping. In the backward mapping, a ray starting from a pixel in EIA reaches the corresponding pixel in a RGBD image. On the contrary, in the forward mapping, a ray coming from a pixel in a RGBD image is mapped into the corresponding pixel in EIA, passing through a pinhole. The ray diagrams of the backward and forward mapping methods are shown in **Fig. 1(a)** shows the backward mapping where the rays starting from an elemental image pass through the corresponding pinhole and then they reach the 3D object space. Here, the 3D object space is evaluated by the depth image. However, there is a need for a stopping criterion to determine where to stop the rays at the specific distance, as shown in **Fig. 1(a)**. On the other hand, **Fig. 1(b)** shows the forward mapping where the rays emanating from the RGBD image pass through a pinhole, arriving at a pixel in an elemental image. This mapping needs to be done for all pinholes to obtain an elemental image array. In this case, there is no need for a stopping criterion.

Table 1. Variables in pixel-to-pixel mappings

Variables	Note	Unit
u_k v_k i j	Coordinates of k-th elemental image in width Coordinates of k-th elemental image in height Coordinates of each object plane image in width Coordinates of each object plane image in height	integer number
$N_s \times N_s$ $N_e \times N_e$ $N_k \times N_k$ $N_p \times N_p$ N_d	Number of pixels of elemental image array Number of pixels of each elemental image Number of elemental images Number of pixels of each object plane image Number of depth levels	integer number
$L_s \times L_s$ $L_p \times L_p$ P_L P_I g f	Physical length of elemental image array Physical length of each object plane image Pitch of elemental lens Pitch of object pixel Gap between pinholes and CCD Focal length	*Meter*

To describe the two mapping methods theoretically, a set of physical variables is previously defined in **Table 1**. The variables represent the coordinates for elemental images and object plane images, some numbers for elemental images and object plane images, and physical information for a lens array and object planes. Then, the geometrical mapping relation between an elemental image array and the 3D object space for the backward mapping can be defined by

$$i_l = u_k P_L (u_k P_I - u_k P_L) \frac{z(l)}{g}, \tag{1}$$

$$j_l = v_k P_L (v_k P_I - v_k P_L) \frac{z(l)}{g}. \tag{2}$$

The pixel (u_k,v_k) of the k-th elemental image is mapped into the pixel (i_l, j_l) of the reconstruction plane image for the depth (l). This means that every pixel of all elemental images is calculated from the equations. Thus, the elemental image array is generated without missing any pixel. On the other hand, the forward mapping can be presented by

$$u_k = iP_L (iP_I - iP_L) \frac{g}{z(i,j)}, \tag{3}$$

$$v_k = jP_L (jP_I - jP_L) \frac{g}{z(i,j)}. \tag{4}$$

Here, the pixels (i, j) of the plane image with depth $z(i, j)$ are mapped into corresponding pixels of the elemental images (u_k, v_k). There can exist missing pixels in the elemental image array although the whole pixels of the plane image with depth are employed. This missing pixels are called as holes [26]. Thus the forward mapping can require a post processing such as an image interpolation to fill out the holes.

3. Analysis of Backward and Forward Mappings

3.1 Complexity analysis

Generally, the number of pixels in an elemental image array and that of pixels in each elemental image are very important parameters, rendering 3D images in an integral imaging system, since those numbers of pixels affect directly the cost of systems. Thus, the complexity analysis is important in the computer-generated integral imaging systems. First, let us consider a number of multiplication operations in (1) through (4). According to (1) through (4), it is seen that the number of the operations are the same, five multiplications, one division, and an addition. Next, let us define a basic mapping operation that is required for calculation of a pixel-to-pixel mapping; and it consists of two equations. In other words, a basic operation for the backward mapping is defined by the two equations (1) and (2). Moreover, a basic operation for the forward mapping is represented by (3) and (4). Here, it is seen that the two basic operations for the forward and backward are the same.

In the backward mapping model as shown in **Fig. 1(a)**, the number of the mapping operations to generate the whole elemental image array is calculated by a product of the number of the sensor $(N_e^2 \times N_k^2)$ and the depth level (N_d). The total number (C_b) of mapping operations for the backward mapping is thus given by

$$C_b = N_e^2 \times N_k^2 \times N_d. \tag{5}$$

In the forward mapping model, many rays coming from a pixel in a RGBD image, passing through the associated pinholes, are mapped into all elemental images independently. Thus, the total number of mapping operations (C_f) is evaluated by the product of the number of pixels (N_p^2) of a RGBD image and the number (N_k^2) of pinholes. This is written by

$$C_f = N_p^2 \times N_k^2. \tag{6}$$

Here, let us denote the ratio of the backward and forward operations as R_0. The ratio is represented by

$$R_0 = \frac{C_b}{C_f} = \frac{{N_e}^2 \times {N_k}^2 \times N_d}{{N_p}^2 \times {N_k}^2} = \frac{{N_e}^2 \times N_d}{{N_p}^2}. \tag{7}$$

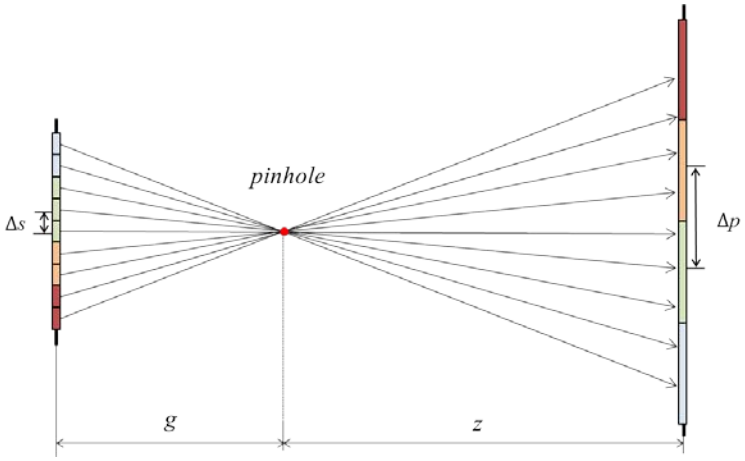
According to (7), it is seen that the common variable ${N_k}^2$ is canceled out. It means that the number of pinholes is regardless in comparison of the two mapping models. Using the ratio R_0, we can select the effective model by considering the computational speed based on R_0. The backward mapping model is superior to the forward mapping model if $R_0 < 1$, whereas the forward mapping is better than backward mapping if $R_0 > 1$. For example, when $R_0 = 0.5$, the generation speed of the forward mapping model is two times faster than that of the backward mapping model.

3.2 Analysis of Considering Hole-Problem

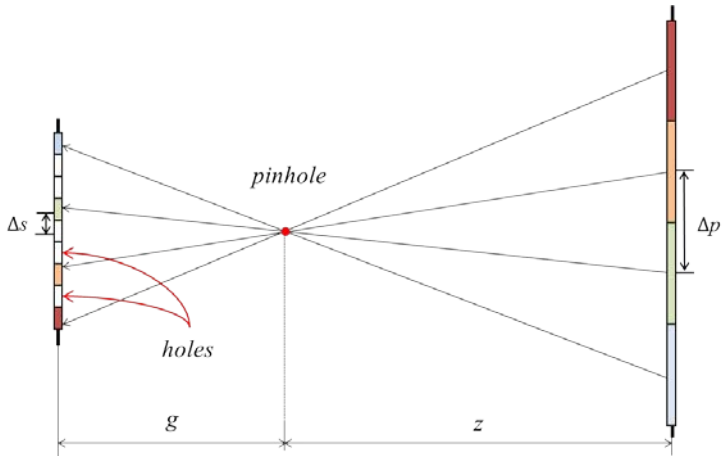
Compared with the backward mapping, the forward mapping can suffer from the holes. It is required to compare the two mappings without the hole problem. Thus, an analysis taking the hole-problem into account presents in this section to evaluate the two mappings more accurately. **Fig. 2** shows the two mappings between an elemental image and an object plane in a RGBD image. The distance between the elemental image and its associated pinhole is g and the object plane is located at z from the pinhole. Let us denote the distance between two adjacent pixels as Δs and the distance between two adjacent pixels in the object plane as Δp, which are written by

$$\Delta s = L_s / N_s, \tag{8}$$

$$\Delta p = L_p / N_p. \tag{9}$$



(a)



(b)

Fig. 2. Mappings between an elemental image and a RGBD image (a) backward (b) forward

As shown in **Fig. 2(a)**, in the backward mapping, every pixel in the given elemental image is associated with the object plane. Thus, no holes in the elemental image exist. On the other hand, in the forward mapping, the rays emanating from the object plane are mapped into a part of pixels in the given elemental image, as shown in **Fig. 2(b)**. Thus, the holes can exit in the elemental image. However, the holes can disappear in some cases. For example, the holes can disappear in case that Δp is much smaller than Δs. Generally, it can be seen that the interval of the adjacent rays coming from the object plane less than Δs. Here, the interval is evaluated by $\Delta p/M$, where $M = z/g$. In other words, the holes disappear in case that Δp is less than $\Delta s \times M$. In this paper, we call this condition as the hole-free condition in the forward mapping, which is written by

$$\Delta p \leq \Delta s \times M = \Delta s \times z/g. \tag{10}$$

The hole-free condition tells that Δp should be smaller than $\Delta s \times M$, thus the condition is directly affected by the magnification factor M. In other words, an object in a far distance has less holes than in a near distance, when generating an elemental image array in integral imaging. In addition, substituting (8) and (9) into (10) give us

$$\frac{L_p}{N_p} \leq \frac{z}{g} \times \frac{L_s}{N_s}. \tag{11}$$

The hole-free condition of (11) can provide the computational cost required for the forward mapping. As mentioned above, the variable N_p is the most important parameter in comparison of the two mappings. Thus, it need to be rearranged based on N_p. Rewriting (11) provides the hole-free condition as a form of

$$N_p \geq \frac{L_p N_s}{L_s M}. \tag{12}$$

This means that N_p needs to be greater than a critical number for the hole-free condition of the forward mapping. The critical point of N_p can be written as $N_p=(L_pN_s)/(L_sM)$.
Furthermore, applying (12) into (7) provides a modified ratio of the backward and forward operations. The ratio is then represented by

$$R_0 = \frac{C_b}{C_f} = \frac{{N_e}^2 \times N_d}{{N_p}^2}, \text{ with } N_p \geq \frac{L_p N_s}{L_s M}. \quad (13)$$

Similarly, as discussed about the ratio R_0, the ratio R_0 determines which is more effective between the two mappings. If the condition regarding N_p holds additionally, then the forward mapping method reconstructs an elemental image array without holes.

4. Experimental Results and Discussion

Experiments are conducted to verify our analysis based on theoretical results discussed above. These experiments provide an experimental ratio to compare it with its theoretical ratio. To do so, we carry out two experiments regarding computational speed and the hole-free condition. The first experiment is performed based on the computational speed according to the several variables to choose an efficient mapping method. The second experiment is carried out in terms of the image quality with the holes. For our experiments, we generated RGBD images, as shown in **Fig. 3(a)-(h)** are a 2D test image and several kinds of depth maps, respectively. Here, a simple computer simulation using concatenating several planes generates the depth maps.

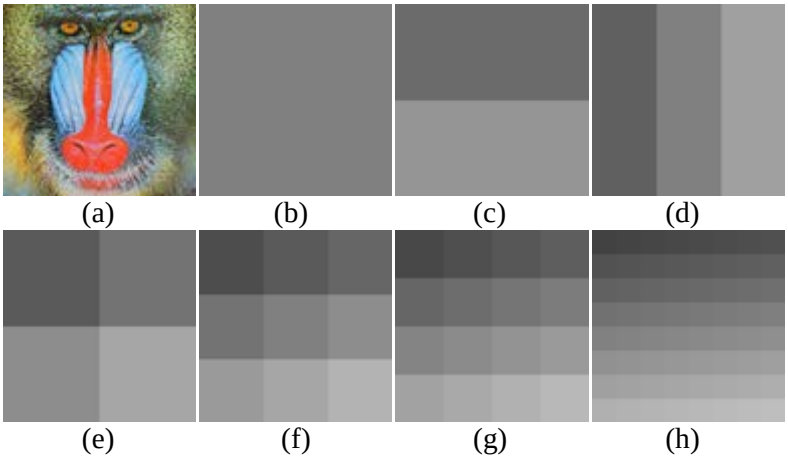


Fig 3. Test RGBD image set (a) 2D image (b) depth image with one level (c) two levels (d) three levels (e) four levels (f) nine levels (g) sixteen levels (h) sixty four levels.

4.1 Experiment regarding Processing Time

The first experiment is to measure the computational cost according to some key variables of the two mapping method. The key variables are the number of pixels in each elemental image and that of pixels in an object plane image or RGBD image, denoted by N_e and N_p, respectively. The length of EIA L_s and that of an object plane L_p are both set to 500 mm. The two mapping algorithms are written in the C-language running over a computer with a CPU of 3.40 GHz. The test images are shown in **Fig. 3**. The number of elemental images is set by $N_k = 16$.

Fig. 4 show the experimental result for the variable N_e. We set N_e= {16, 32, 64 128, 256}. The graph shows changes of the computational processing time for two mapping methods according to the variables N_e, N_p, and N_d. From the result of **Fig. 4**, the computational processing time continues to increase as N_e increases in the backward mapping. In addition, it is seen that a large number of N_d can increase the processing time exponentially. On the other hand, the computational time is constant for the forward mapping, although adjusting N_e. In other word, the forward mapping only depends the number of pixels in an object plane, the size of a RGBD image N_p.

According to (7), the ratio is theoretically evaluated to be R_0 = 1 for N_e=150, N_d=16, and N_p=600. On the other hand, the experiment results in **Fig. 4** show the computational time graphs for the two mapping methods, including the two graphs of backward mapping with N_e=150, N_d=16 and forward mapping with N_p=600. The crossing point of the two graphs means that the computational time is the same, in other word, R_0 = 1. Also, it is seen that the point in **Fig. 4** is located in case of N_e=158, N_d=16, and N_p=600. Therefore, this result indicates the experimental result well matches the theoretical result. In addition, simply calculating (7) determines which mapping is more efficient for a given variables.

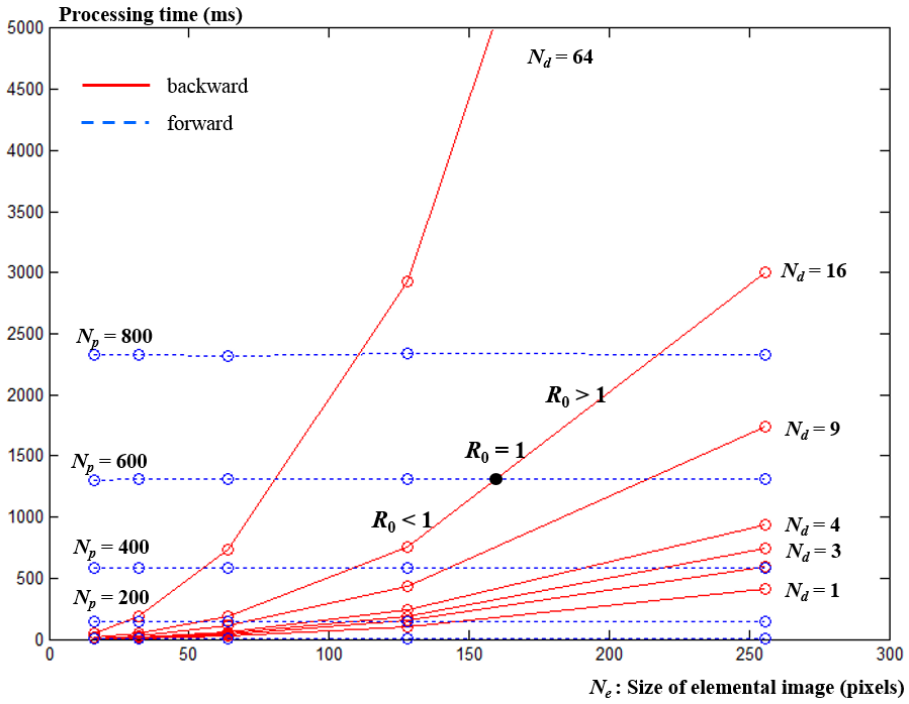


Fig. 4. Processing time adjusting elemental image size

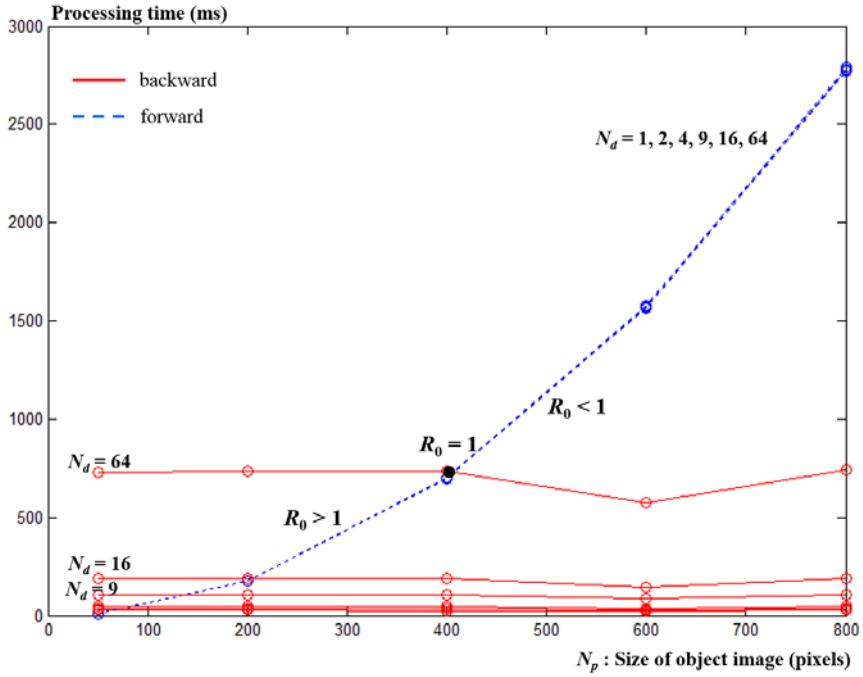


Fig. 5. Processing time adjusting RGBD image size

Next, the processing time adjusting N_p. is shown in **Fig 5**. Under the conditions of N_e = 48 and N_p = {50, 200, 400, 600, 800}, the computational time of the two mapping methods are plotted in terms of N_p and N_d. Here, it is seen that the forward mapping is directly related to the variable N_p, not N_e and N_d. The crossing points for N_p = 200 and 400 in **Fig. 5** mean that the ratio R_0 = 1 experimentally. And the theoretical ratios are both evaluated to be R_0 = 0.9216. Thus, this experiment indicates that the theoretical and experimental ratios are almost the same. In addition, as N_p increases (R_0< 1), the forward mapping method requires more computational time. On the other hand, as N_p decreases (R_0> 1), the processing time for the backward mapping method increases.

4.1 Experiment regarding Hole-Free Condition

In this experiment, the hole-free condition is discussed in terms of the visual quality of the reconstructed integral images for both backward and forward mapping methods. The same test RGBD images shown in **Fig. 3(e)-(f)** are used to reconstruct elemental image arrays. The experimental conditions are followings: N_k=3, N_e=80, L_p = 500 mm, L_s =500 mm, from g = 20 mm, The depth level are 4 and 9 with a step of 20 mm. Thus, it is set to be $N_s = N_k$, N_e= 384 and M = {0.5, 1, 1.5, 2.0} and {0.5, 1, …, 4.5}.

The generated EIAs are shown in **Fig. 6**. The two EIAs in **Fig. 6(a)** and **(b)** are generated from the RGBD image in **Fig. 2(e)** by use of the two mapping methods, where N_p is set to be 200. It is seen that the EIA in **Fig. 6(a)** from the backward mapping has no holes whereas the EIA in **Fig. 6(b)** from forward mapping suffers from the holes. Interestingly, the elemental image in the upper side of **Fig. 6(b)** shows the holes, where the elemental image is located at M=1.0 and thus the critical point $(L_pN_s)/(L_sM)$=240 is greater than 200, which means that the hole-free condition do not hold. On the other hand, the elemental image that is located at

M=2.0, in the lower side of **Fig. 6 (b)**, shows no holes since the critical point $(L_pN_s)/(L_sM)$=120 is lower than 200, which means that the hole-free condition holds.

For another example, two EIAs obtained from the backward and forward mappings are shown in **Fig. 6(c)** and **(d)**, where the depth level is 9 and N_p=800. Here, it is said that there are no holes since a large number of N_p greater than the critical point guarantees the hole-free condition. Thus, two mapping methods can be considered in terms of image quality. However, the ratio R_0 = 0.09 says that the forward mapping is much slower than the backward mapping. Therefore, in this case, the backward mapping is preferable.

It should be noted that there may be black areas in **Fig. 6(c)** and **(d)** although the hole-free condition is satisfied. This is called as the disocclusion, not the holes. The disoccusion occurs when the rays pass through a gap between objects and reach the background. Here, in this experiment, the background is black. Moreover, this disocclusion appears in the backward and forward mapping method. Thus, a method of removing the disoccusion such as image-inpainting is beyond the topic of this paper.

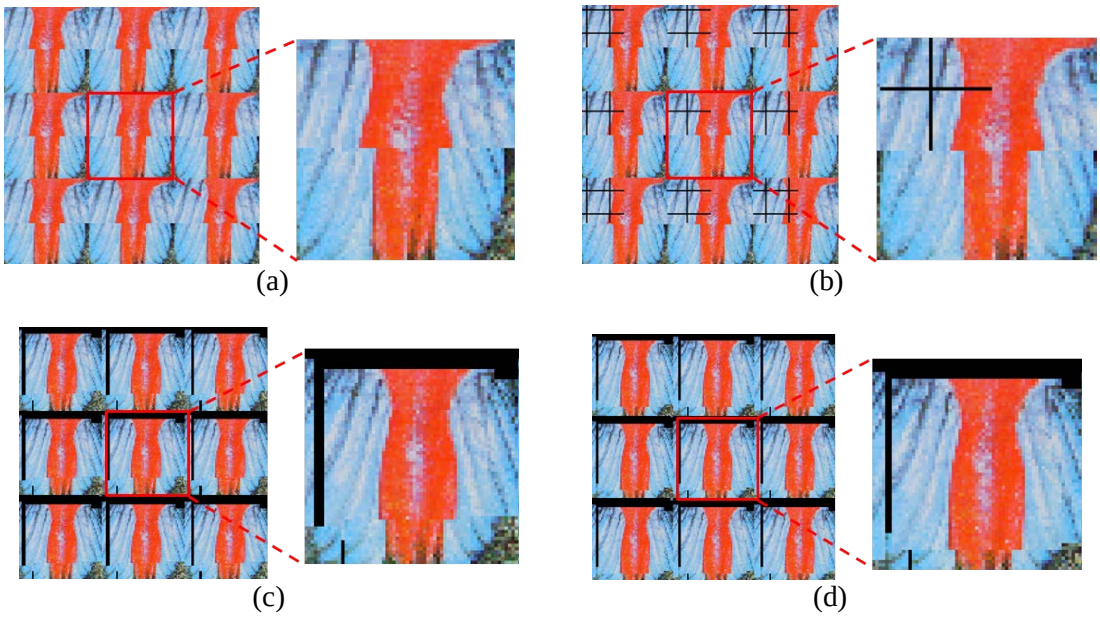


Fig. 6. CGII result using test image by point retracing rendering
(a) Backward, N_d = 4, N_p = 200 (b) Forward, N_d = 4, N_p = 200
(c) Backward, N_d = 9, N_p = 800 (d) Forward, N_d = 9, N_p = 800

5. Conclusion

Two transformations and their performance analyses between an elemental image array and a RGBD image have been proposed for three-dimensional computational integral imaging systems. In this paper, the forward and backward mapping methods have been introduced and analyzed theoretically and experimentally. Our analysis can determine which mapping method is more effective for a given condition. Moreover, in the forward mapping method, the hole-free condition has been defined so as to discuss the visual quality. Experimental results indicated that the theoretical analyses are confirmed by the experiments of this paper. Therefore, we expect that our analyses can be effectively applied to an efficient generation of

an elemental image array in the areas such as computer-generated integral imaging, especially for transmitting and displaying 3D data efficiently.

Acknowledgements

I would like to give special thanks to Yungkyung Lee for helping me. In addition, this work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

References

[1] G. Lippmann, "La photographic intergrale," *Computes-Rendus. Acad. Sci*, vol. 146, pp. 446-451, 1908.
[2] Y. Igarashi, H. Murata, and M. Ueda, "3D display system using a computer generated integral photography," *Jpn. J. Appl. Phys.*, vol. 17, no. 9, pp. 1683–1684, Jan. 1978. Article (CrossRef Link)
[3] N. Dodgson, "Autostereoscopic 3D displays," *IEEE Computer*, vol. 38, no. 8, pp. 31-36, 2005. Article (CrossRef Link)
[4] N. S. Holliman, N. A. Dodgson, G. E. Favalora, and L. Pockett, "Three-dimensional displays: a review and applications analysis," *IEEE Trans. Broad.*, vol. 57, no. 2, pp. 362-371, Jun. 2011. Article (CrossRef Link)
[5] A. Stern and B. Javidi, "Three-dimensional image sensing, visualization, and processing using integral imaging," *Proc. IEEE*, vol. 94, no. 3, pp. 591-607, Mar. 2006. Article (CrossRef Link)
[6] S. Park, J. Yeom, Y. Jeong, N. Chen, J.-Y. Hong, and B. Lee, "Recent issues on integral imaging and its applications," *Journal of Information Display*, vol. 15, no. 1, pp.37-46, 2014. Article (CrossRef Link)
[7] J.-S. Jang and B. Javidi, "Three-dimensional synthetic aperture integral imaging," *Optics Letters*, vol. 27, no. 13, pp. 1144-1146, Jul. 2002. Article (CrossRef Link)
[8] H. Yoo, "Artifact analysis and image enhancement in three dimensional computational integral imaging using smooth windowing technique," *OSA Optics Letters*, vol. 36, no. 12, pp. 2107-2109 Jun. 2011. Article (CrossRef Link)
[9] H. Yoo, "Depth extraction for 3D objects via windowing technique in computational integral imaging with a lenslet array," *Elsevier Optics and Lasers in Engineering*, vol. 51, no. 7, pp. 912-915, Jul. 2013. Article (CrossRef Link)
[10] H. Yoo, D.-H. Shin, and M. Cho, "Improved depth extraction method of 3D objects using computational integral imaging reconstruction based on multiple windowing techniques," *Elsevier Optics and Lasers in Engineering*, vol. 66, no. 3, pp. 105-111, Mar. 2015. Article (CrossRef Link)
[11] Y. Lee and H. Yoo, "Three-dimensional visualization of objects in scattering medium using integral imaging and spectral analysis," *Elsevier Optics and Lasers in Engineering*, vol. 77, no. 2, pp. 31-38, Feb. 2016. Article (CrossRef Link)
[12] Y. Piao, H. Qua, M. Zhang, and M. Cho, "Three-dimensional integral imaging display system via off-axially distributed image sensing," *Optics and Lasers in Engineering*, vol. 85, pp. 18-23, Oct. 2016. Article (CrossRef Link)
[13] C. Fehn, "A 3D-TV system based on video plus depth information," in *Proc. of 37th Asilomar Conf. on IEEE Signals, Systems and Computers*, vol. 2, pp. 1529-1533, Nov. 2003. Article (CrossRef Link)
[14] C. Fehn, "A 3D-TV approach using depth-image-based rendering (DIBR)," *Vis., Imaging, Image Process.*, pp. 482-487, 2003.
[15] L. Zhang and W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV," *IEEE Trans. Broadcasting.*, vol. 51, no. 2, pp. 191-199, June 2005. Article (CrossRef Link)

[16] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3D mapping with an RGBD camera," *IEEE Trans. Robotics*, vol. 30, no. 1, Feb. 2014. Article (CrossRef Link)
[17] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybernet*., vol. 43, no. 5, Oct. 2013.
[18] M. Zhao, C.-W. Fu, J. Cai, and T.-J. Cham, "Real-time and temporal-coherent foreground extraction with commodity RGBD camera," *IEEE Journal of selected topics in signal processing*, vol. 9, no. 3, Apr. 2015.
[19] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robot. Auto. Lett*., vol. 1, no. 2, pp.1179-1185, July 2016. Article (CrossRef Link)
[20] D.-H. Shin, S.-H.Lee, E.-S.Kim, "Optical display of true 3D object in depth-priority integral imaging using an active sensor," *Optical Comm*., vol. 275, no. 2, pp. 330-334, Jul. 2007. Article (CrossRef Link)
[21] G. Li, K.-C. Kwon, G.-H. Shin, J.-S. Jeong, K.-H. Yoo, and N. Kim, "Simplified integral imaging pickup method for real objects using a depth camera," *J. Optical Society of Korea*, vol. 16, no. 4, pp. 381-385, Dec. 2012. Article (CrossRef Link)
[22] D. Hiyama, T. Shimobaba, T. Kakue, and T. Ito, "Acceleration of color computer-generated hologram from RGBD images using color space conversion," *Elsevier Optics Communications*, vol. 340, no. 1, pp.121-125, Apr. 2015. Article (CrossRef Link)
[23] K.-C. Kwon, C. Park, M.-U. Erdeneat, J.-S. Jeong, J.-H. Choi, N. Kim, J.-H. Park, Y.-T. Lim, and K.-H. Yoo, "High speed image space parallel processing for computer-generated integral imaging system," *Optics Express*, vol. 20, no. 2, pp. 732-740, Jan. 2012. Article (CrossRef Link)
[24] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang, "Depth maps fusion for multiview stereo via matrix completion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 566-582, Sept. 2012. Article (CrossRef Link)
[25] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 3, pp. 383-396, March 2013. Article (CrossRef Link)
[26] Y. Gao, G. Cheung, T. Maugey, P. Frossard, and J. Liang, "Encoder-driven inpainting strategy in multiview video compression," *IEEE Trans. Image Process*. vol. 25, no. 1, Jan. 2016. Article (CrossRef Link)

Hoon Yoo received his BS, MS, and Ph. D degree in electronic communications engineering from Hanyang University, Seoul, Korea, in 1997, 1999, and 2003 respectively. From 2003 to 2005, he was with Samsung Electronics Co., Ltd., Korea, where he was involved in the world's first development and mass-production of the satellite DMB (digital multimedia broadcasting) and the terrestrial DMB over mobile phones. He submitted 23 domestic and international patents in the period of the development. From 2005 to 2008, as an assistant professor, he was with Dongseo University, Busan, Korea. Since 2008, as an associate professor, he has been with Sangmyung University, Seoul, Korea. In 1997, he received a Paper Award from the third Samsung Human-Tech Paper Award, which is supported by Samsung Electronics Co., Ltd. His research interests are in the area of image processing such as image interpolation, 3D integral imaging, image compression, de-interlacing, and de-mosaicking. He is also interested in real-time image processing systems of the techniques mentioned above and their implementation with low-complexity.