



Viewing Angle-Improved 3D Integral Imaging Display with Eye Tracking Sensor

Seokmin Hong¹, Donghak Shin^{1*}, Joon-Jae Lee², and Byung-Gook Lee³, *Member, KIICE*

¹Institute of Ambient Intelligence, Dongseo University, Busan 617-833, Korea

²Department of Game Mobile Contents, Keimyung University, Daegu 705-701, Korea

³Department of Visual Contents, Dongseo University, Busan 617-833, Korea

Abstract

In this paper, in order to solve the problems of a narrow viewing angle and the flip effect in a three-dimensional (3D) integral imaging display, we propose an improved system by using an eye tracking method based on the Kinect sensor. In the proposed method, we introduce two types of calibration processes. First process is to perform the calibration between two cameras within Kinect sensor to collect specific 3D information. Second process is to use a space calibration for the coordinate conversion between the Kinect sensor and the coordinate system of the display panel. Our calibration processes can provide the improved performance of estimation for 3D position of the observer's eyes and generate elemental images in real-time speed based on the estimated position. To show the usefulness of the proposed method, we implement an integral imaging display system using the eye tracking process based on our calibration processes and carry out the preliminary experiments by measuring the viewing angle and flipping effect for the reconstructed 3D images. The experimental results reveal that the proposed method extended the viewing angles and removed the flipping images compared with the conventional system.

Index Terms: Eye tracking, Integral imaging, Projective transformation, 3D space calibration

I. INTRODUCTION

Recently, integral imaging (InIm) has been considered one of the effective technologies for next-generation three-dimensional (3D) displays [1-10]. In general, the pickup part of InIm is composed of a lenslet array and a two-dimensional (2D) image sensor. Here, the optical rays coming from a 3D object are picked up by the lenslet array and recorded with the 2D image sensor as elemental images, which have their own perspective of the 3D object. On the other hand, the display part of InIm is a reverse of the pickup process. The elemental images are displayed in front of the lenslet array for the reconstruction of 3D images.

InIm has several merits and can provide both horizontal and vertical parallaxes, color images, and quasi-continuous views to observers [1-4]. However, it also has the drawbacks of a low-resolution 3D image, a narrow viewing angle, and a small depth range. Many researchers have been working towards solving these problems [5-10]. Among them, as a solution to the abovementioned limitation of the viewing angle, the computer vision technology of eye tracking has been applied to the InIm display system [5]. Here, the researchers proposed a tracking InIm system with an infrared (IR) camera and IR light-emitting diodes, which can track the viewers' exact positions. However, since the user has to wear headgear with IR diodes on his/her head to

Received 02 September 2014, Revised 29 October 2014, Accepted 01 December 2014

*Corresponding Author Donghak Shin (E-mail: shindh2@gmail.com, Tel: +82-51-320-2709)

Institute of Ambient Intelligence, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 617-833, Korea.

Open Access <http://dx.doi.org/10.6109/jicce.2014.12.4.208>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

check his/her position, this system is not suitable for practical applications. In this paper, we propose an improved InIm system with an eye tracking method based on the Kinect sensor [11, 12]. To do so, we need a tracking system to change the elemental images dynamically when the viewer's position is changed. In the proposed method, we newly introduce a 3D space calibration process for obtaining a more exact 3D position of the observer's eyes. The use of the eye tracking technology in InIm can provide a wider viewing angle for 3D image observations.

II. PROPOSED INIM DISPLAY SYSTEM WITH THE KINECT SENSOR

A. System Structure

Fig. 1 shows the principle of the InIm method. An InIm system is composed of two processes: a pickup process and a display process, as shown in Fig. 1(a) and (b), respectively. In InIm, the lenslet array is used in both processes to capture 3D objects and reconstruct 3D images. In the pickup process, the light rays coming from 3D objects pass through the lenslet array and record a set of different perspective images by using an image sensor. These recorded images are referred to as elemental images (EIs). In the reconstruction process, a lenslet array similar to that in the pickup process is used. The 3D images are formed at the location where the image was picked up by backpropagating light rays of the EIs through the lenslet array.

In general, the viewing angle ψ of the InIm system is defined by

$$\tan\left(\frac{\psi}{2}\right) = \frac{p}{2g}. \quad (1)$$

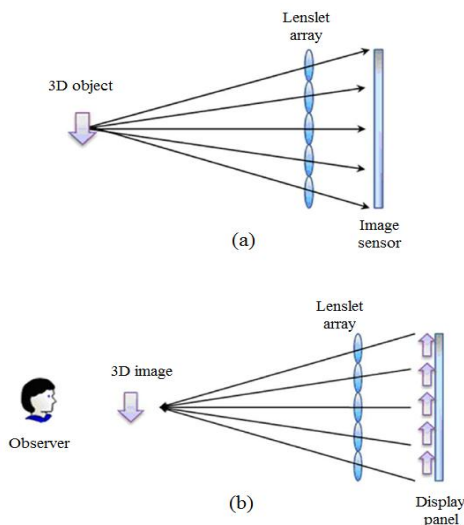


Fig. 1. Integral imaging system: (a) pickup and (b) display.

From Eq. (1), we can see that the viewing angle is dependent on the diameter of each lens and the gap between the lens array and the display panel. In general, the viewing angle is small because of the use of the lens array with a large f-number. For example, when $p = 1$ mm and $g = 3$ mm, the viewing angle becomes 18.925° .

In this study, we want to improve the viewing angle by using the eye tracking technology based on the Kinect sensor. The structure of the proposed system is shown in Fig. 2. The Kinect sensor is placed on top of the display panel to detect the eye position of the observer. In this system, the proposed tracking method is as follows: for the given InIm system, firstly, we calibrate the 3D space to calculate the exact 3D location of the observer's eyes. Secondly, we find 3D location of the observer's eyes by using the tracking technology. Thirdly, we generate elemental images to display 3D images after considering the abovementioned eye location at a real-time speed.

B. 3D Space Calibration and the Coordinate Conversion Matrix

In general, a tracked object is oriented by the image's coordinate system and displayed on the 2D coordinate display system. In the real world, however, the coordinate system is represented by the 3D space. Therefore, we need to calibrate the 3D space and perform a coordinate conversion between the image plane and the real 3D space. To calibrate between the image plane and the real 3D space, we need the position information of the observer and the target object. To solve this problem, we choose a 3D camera and use it as a tracking system.

Various 3D cameras and different types of methods have been developed to obtain 3D information in a real 3D space [11-14]. One of the widely used sensors is Kinect from

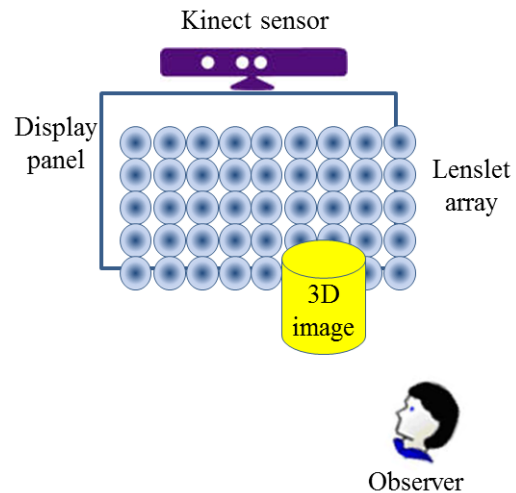


Fig. 2. System structure.



Fig. 3. Mapping the color and depth information with the given function from the Kinect software development kit (SDK).

Microsoft, which is a pattern structured IR-light type sensor. This device can provide color and depth information simultaneously [11]. However, it faces a problem due to the physical separation between the color and the depth cameras. To overcome this problem, we reference its software development kit (SDK) from Microsoft and develop a simple solution. We can use references in the function ‘NuiImageGet-ColorPixelCoordinateFrameFromDepthPixelFrameAtResolution()’ [11]. Fig. 3 shows the processing result. Thus, we obtain the color and depth information, which is simultaneously mapped to one another in real time.

Then, we calibrate the 3D space. To do so, we use the calibrator to detect feature points and regular pattern influences in order to increase accuracy [14]. In the proposed system, we use the calibrator with the same size of the monitor display, as shown in Fig. 4. This figure shows the specific information of the calibrator. Next, we extend the monitor’s coordinate system by using the calibrator. We mount Kinect on top of the monitor and move in steps of 10 cm while capturing the calibrator. The initial distance is 80 cm due to the shortest possible capturing depth distance from the Kinect sensor. Further, the maximum distance is 150 cm. This distance is chosen because it is a reliable distance to accurately find the corner points. After capturing the points at each distance, we need a parallel listing of the corner points, i.e., the calibrator’s corner points (based on the image coordinates) and the calibrator’s physical corner positions (based on the world coordinates). Fig. 5 shows the corresponding processing result obtained by using OpenGL.

Using the listing of the corner points, we can calculate the least squares through projective transformation, for various 3D feature points. For the given n point correspondences, the n vector equation is as follows:

$$\lambda_k m'_k = P m_k \text{ where } P_{i,j} (i = 1 \dots 4, j = 1 \dots 4) \text{ and } \lambda_k (k = 1 \dots n). \quad (2)$$

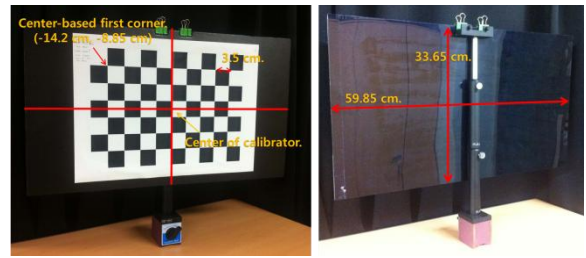


Fig. 4. Specifications of the calibrator.

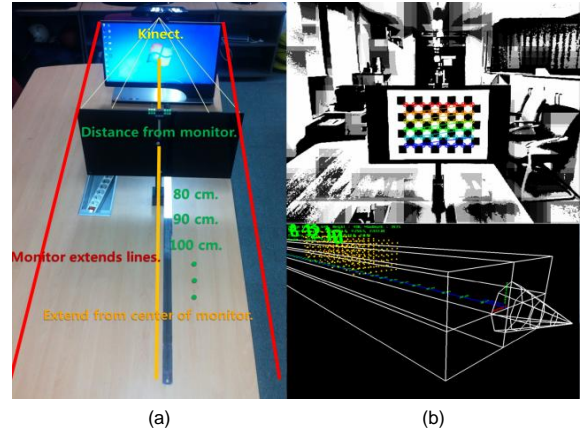


Fig. 5. (a) Extending the monitor’s coordinate system and (b) the corresponding processing result.

In this case, normally, λ_1 becomes 1 [15]. P is the projective transformation matrix. It is a 3D expression and is represented by a 4×4 matrix as follows:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}. \quad (3)$$

These equations can be simplified as follows:

$$Ax = b, \text{ Let } x = [P_{11} \dots P_{44}, \lambda_2 \dots \lambda_n] \text{ where } \lambda_1 = 1$$

$$A = \begin{bmatrix} M_1 & & & & & \\ M_2 & -M'_2 & & & & \\ M_3 & & -M'_3 & & & \\ \vdots & & & \ddots & & \\ M_n & & & & -M'_n & \end{bmatrix} \quad (4)$$

$$M_i = \begin{bmatrix} M_i^T & & & \\ & M_i^T & & \\ & & M_i^T & \\ & & & M_i^T \end{bmatrix}$$

$$b = [M_1^T, 0, \dots, 0]^T.$$

A is a $4n \times (15 + n)$ matrix, and b is a $4n$ vector. Among them, we can calculate x through the least squares solution and calculate P through Eq. (5) by using paired and listed correspondences.

$$x = (A^T A)^{-1} A^T b. \tag{5}$$

However, the difference between the columns of the data which is inside of the matrix has quite different magnitude. Thus, we need to normalize the values. When we find the least squares for various values, we prefer to normalize the values because doing so yields good results with respect to the least squares for the projective transformation matrix [16]. In our system, we set the center coordinates of the image as $(0, 0)$. If we follow equation 6, the normalization result is from $(-1.0, -1.0)$ to $(1.0, 1.0)$. Further, $xMax$ and $yMax$ denote the full resolution size of the image.

$$\begin{bmatrix} \frac{2}{xMax} & -1 \\ \frac{2}{yMax} & -1 \\ & 1 \end{bmatrix}. \tag{6}$$

By following the normalization process, we can obtain a stable and correct projective transformation matrix P . P is the coordinate conversion matrix between the Kinect and the monitor.

C. Estimating 3D Eye Position and Its Coordinate Conversion

The Kinect SDK includes a face tracking library called ‘FaceTrackLib.’ It can detect a face with 121 vertices. Furthermore, it can provide the translate, rotate, and scale factors [11]. Among them, we just need the eye position. The Kinect SDK can detect the eye’s position with 18 vertices. In fact, the Kinect SDK can provide the coordinate position of only 2D vertices from the tracked face information. Therefore, we mapped the color and the depth information processes and consequently, obtained the eye’s correct depth distance from the Kinect sensor on the basis of the mapped depth information. Fig. 6 illustrates our process to detect the 3D position of the observer’s eye.

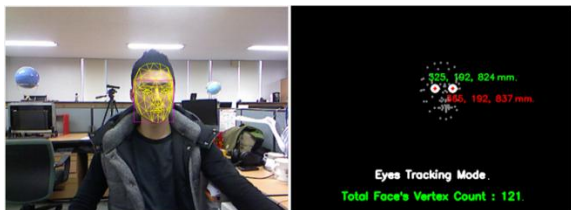


Fig. 6. Finding the 3D position of both eyes by using the face tracking information.

However, detected eye’s coordinate system is still oriented by Kinect’s own coordinate system. Therefore, we need to convert the coordinates system from the Kinect to the monitor. We already calculated the projective transformation matrix P in the previous section and the eye’s 3D coordinates system can be converted into the center coordinates of the monitor system by using matrix P . Fig. 7 shows the experimental results. Fig. 7(a) shows the scene generated by using the coordinate system of Kinect, and Fig. 7(b)–(d) illustrate the result of the conversion into the center coordinates of the monitor system. Finally, we can process the coordinate system conversion between the Kinect and the monitor.

D. Computational Generation of Elemental Integral Image for InIm Display

To display 3D images in the InIm system, we need elemental images. These images can be obtained by using either optical pickup or computational pickup. In this study, we use computational pickup, which is suitable for real-time generation of elemental images. For the sake of simplicity, we use two different images, namely a background image

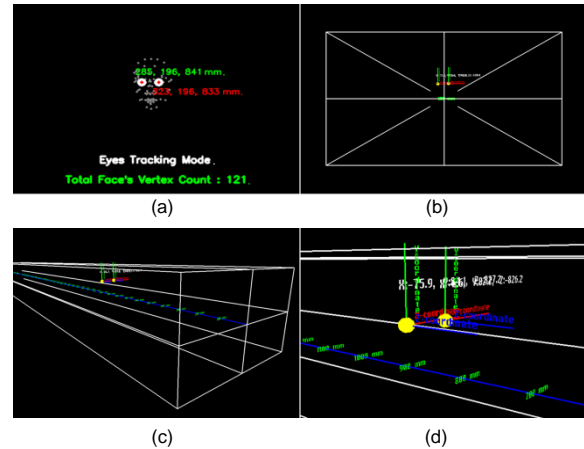


Fig. 7. Coordinate conversion results between the Kinect (a) and the monitor (b–d).



Fig. 8. Result of merging two different elemental images.

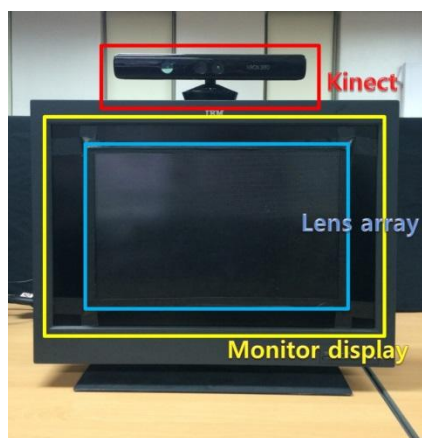


Fig. 9. Overview of the proposed implementation system.

Table 1. Specifications of the proposed implementation system

Hardware specifications	Intel i7 Octa core (8 CPUs) 3840 × 2400 UHD Kinect for Xbox Lens array
Software specifications	Windows 7 Visual Studio 2010 C++, MFC OpenCV 2.4.2 Kinect SDK 1.7
Lens-array specifications	Focal length: 8 mm Lens diameter: 1.6 mm

and a foreground image, as shown in Fig. 8. Each image is converted into the corresponding elemental images. Then, they are merged into the final elemental images and displayed through the InIm display system. Fig. 8 shows the concept of the generation process of elemental images, and the final generated elemental image for the InIm display is shown on the right side of Fig. 8.

III. EXPERIMENTS AND RESULTS

To show the usefulness of our system, we performed preliminary experiments. We used an ultra-high-definition (UHD) monitor as the display panel and placed the lens array in front of this panel. Kinect was placed at the top of the monitor. Fig. 9 shows an overview of our system environment. The specifications of this system are presented in Table 1.

For the given system, we calibrated a 3D space by using the calibrator shown in Fig. 4. Then, we built a Kinect sensor to track the observer and find the 3D position of the observer’s eyes. Based on the 3D position of the tracked

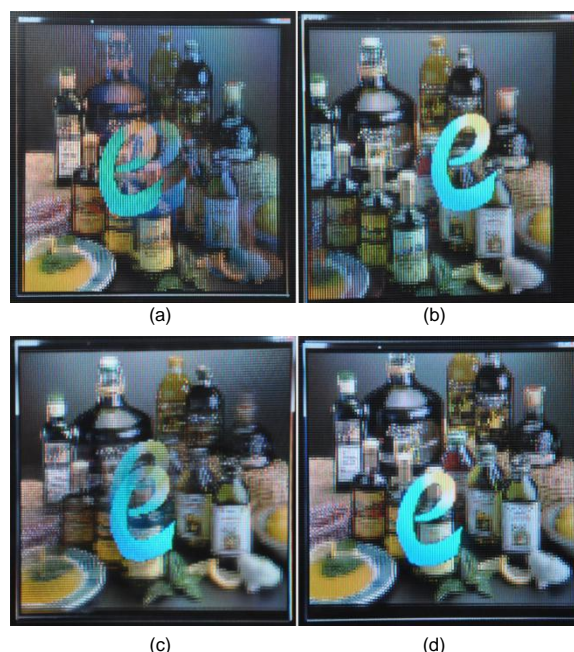


Fig. 10. Results of a comparison of the conventional method and the proposed method. (a) Left view (conventional method). (b) Left view (proposed method). (c) Top view (conventional method). (d) Top view (proposed method).

Table 2. Evaluation of the generation speed of elemental images

Resolution (pixel)	1500 × 1500		3000 × 3000	
	Generate elemental image	Merge elemental images	Generate elemental image	Merge elemental images
Genre				
Time (s)	0.06400	0.02900	0.12800	0.05800
Total (s)	0.09300		0.18600	
FPS	10.75269		5.37634	

FPS: frames per second.

eyes, elemental images with two different image planes (background and foreground images) were generated and merged at a real-time speed.

We checked the computational speed of the elemental image generation process for the real-time display system. Table 2 presents the measured results of the generation speed. The measurement process consisted of two steps. First, we measured the generation speed of each single elemental image, and then, we calculated the generation speed of merge two elemental images. We obtained generation speeds of 10 frames per second (FPS) for an image resolution size of 1500 × 1500 and 5 FPS for an image resolution size of 3000 × 3000.

Next, we experimentally measured the viewing angle of the display system. Fig. 10 shows the results of a comparison of the conventional InIm display system and the proposed method. The observer is placed at a distance of 1

m from the display panel. When the observer moves to the left by 10 cm from the center position of the monitor with parallel, the conventional method causes the flip effect, as shown in Fig. 10(a). In contrast, the proposed method results in a more enhanced viewing angle and there is no flip effect at the displayed image, as shown in Fig. 10(b). Furthermore, there is no flip effect when we move in other directions in the case of the proposed method, as shown in Fig. 10(d). From the results shown in Fig. 10, we infer that the proposed system can improve the viewing angle compared with the conventional method.

IV. CONCLUSION

The proposed system is a combination of an InIm display with a tracking system to overcome the InIm display's limitations of the viewing angle and the flipped image. We conducted two rounds of calibrations to calibrate the 3D space. Further, we demonstrated the enhancement of the viewing angle of the InIm display for a dynamic user's eye position by using a face tracking system based on Kinect. In the experimental result, we could see the 3D displayed image clearly in various positions. Further, the experimental results revealed that the implemented system effectively overcame the limitations of the conventional InIm system.

ACKNOWLEDGMENTS

This work was supported by the IT R&D program of MKE/KEIT (No. 10041682, Development of high-definition 3D image processing technologies using advanced integral imaging with improved depth range).

REFERENCES

- [1] A. Stern and B. Javidi, "Three-dimensional image sensing, visualization, and processing using integral imaging," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 591-607, 2006.
- [2] F. Okano, H. Hoshino, J. Arai, and I. Yuyama, "Real-time pickup method for a three-dimensional image based on integral photography," *Applied Optics*, vol. 36, no. 7, pp. 1598-1603, 1997.
- [3] J. S. Jang and B. Javidi, "Improved viewing resolution of three-dimensional integral imaging by use of nonstationary micro-optics," *Optics Letters*, vol. 27, no. 5, pp. 324-326, 2002.
- [4] Y. Kim, K. Hong, and B. Lee, "Recent researches based on integral imaging display method," *3D Research*, vol. 1, no. 1, pp. 17-27, 2010.
- [5] G. Park, J. H. Jung, K. Hong, Y. Kim, Y. H. Kim, S. W. Min, and B. Lee, "Multi-viewer tracking integral imaging system and its viewing zone analysis," *Optics Express*, vol. 17, no. 20, pp. 17895-17908, 2009.
- [6] M. Martinez-Corral, B. Javidi, R. Martinez-Cuenca, and G. Saavedra, "Integral imaging with improved depth of field by use of amplitude-modulated microlens arrays," *Applied Optics*, vol. 43, no. 31, pp. 5806-5813, 2004.
- [7] J. H. Park, J. Kim, Y. Kim, and B. Lee, "Resolution-enhanced three-dimension/two-dimension convertible display based on integral imaging," *Optics Express*, vol. 13, no. 6, pp. 1875-1884, 2005.
- [8] D. H. Shin, S. H. Lee, and E. S. Kim, "Optical display of true 3D objects in depth-priority integral imaging using an active sensor," *Optics Communications*, vol. 275, no. 2, pp. 330-334, 2007.
- [9] J. Y. Jang, D. Shin, B. G. Lee, and E. S. Kim, "Multi-projection integral imaging by use of a convex mirror array," *Optics Letters*, vol. 39, no. 10, pp. 2853-2856, 2014.
- [10] Y. Oh, D. Shin, B. G. Lee, S. I. Jeong, and H. J. Choi, "Resolution-enhanced integral imaging in focal mode with a time-multiplexed electrical mask array," *Optics Express*, vol. 22, no. 15, pp. 17620-17629, 2014.
- [11] Kinect for Window SDK [Internet], Available: <http://msdn.microsoft.com/en-us/library/hh855347.aspx>.
- [12] J. Kramer, N. Burrus, F. Ehtler, H. C. Daniel, and M. Parker, "Multiple kinects," in *Hacking the Kinect*. New York, NY: Apress, pp. 207-246, 2012.
- [13] Winscape [Internet], Available: <http://www.rational-craft.com/Winscape.html/>.
- [14] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly, 2008.
- [15] Z. Zhang, "Estimating projective transformation matrix (collineation, homography)," Microsoft Research, Redmond, WA, *Technical Report MSR-TR-2010-63*, 2010.
- [16] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580-593, 1997.



Seokmin Hong

was born in Busan, Republic of Korea, in 1986. He received his B.E. and M.S. in Digital and Visual Contents from Dongseo University, Busan, Korea, in 2012 and 2014, respectively. In 2012, Dongseo University honored him with the B.E. Extraordinary Award. Since 2012, he has been working with Institute of Ambient Intelligence, Dongseo University, Korea. His research interests include image processing, computer vision, and applied computer science.



Donghak Shin

was born in Busan, Republic of Korea. He received his B.S., M.S., and Ph.D. in Telecommunication and Information Engineering from Pukyong National University, Busan, Korea, in 1996, 1998, and 2001, respectively. From 2001 to 2004, he was a senior researcher with TS-Photon established by Toyohashi University of Technology, Japan. From 2005 to 2006, he was with the 3D Display Research Center (3DRC-ITRC), Kwangwoon University, Korea. He worked as a research professor at Dongseo University, Korea, from 2007 to 2010. He was a visiting scholar in the Electrical & Computer Engineering department at the University of Connecticut from 2011 to 2012. He is currently a senior researcher at the Institute of Ambient Intelligence, Dongseo University, Korea. His research interests include 3D imaging, 3D displays, optical information processing, and holography.



Joon-Jae Lee

received his B.S., M.S., and Ph.D. in Electronic Engineering from the Kyungpook National University, Daegu, South Korea, in 1986, 1990, and 1994, respectively. He worked for Kyungpook National University as a teaching assistant from September 1991 to July 1993. From March 1995 to August 2007, he was with the Computer Engineering faculty at the Dongseo University, Busan, South Korea. He is currently a full professor of the Department of Game Mobile Contents, Keimyung University. He was a visiting scholar at the Georgia Institute of Technology, Atlanta, from 1998 to 1999, funded by the Korea Science and Engineering Foundation (KOSEF). He also worked for PARMi Corporation as a research and development manager for 1 year from 2000 to 2001. His main research interests include image processing, three-dimensional computer vision, and fingerprint recognition.



Byung-Gook Lee

was born in Busan, Republic of Korea. He received his B.S. in Mathematics from Yonsei University, Korea, in 1987, and his M.S. and Ph.D. in Applied Mathematics from Korea Advanced Institute of Science and Technology (KAIST) in 1989 and 1993, respectively. He worked at the DACOM Corp. R&D Center as a senior engineer from March 1993 to February 1995. He has been working at Dongseo University, Korea, since 1995 and is currently a full professor with the Division of Computer Information Engineering. His research interests include computer graphics, computer-aided geometric design, and image processing.