

## Parallel Processing for Integral Imaging Pickup Using Multiple Threads

**Young-Hee Jang, Chan Park**

Dept. of Information and Industrial Engineering,  
Chungbuk National University, Korea

**Jae-Hyeung Park, Nam Kim**

School of Electrical and Computer Engineering,  
Chungbuk National University, Korea

**Kwan-Hee Yoo**

Dept. of Information and Industrial Engineering and  
Dept. of Computer Education,  
Chungbuk National University, Korea

### ABSTRACT

Many studies have been done on the integral imaging pickup whose objective is to get efficiently elemental images from a lens array with respect to three-dimensional (3D) objects. In the integral imaging pickup process, it is necessary to render an elemental image from each elemental lens in a lens array for 3D objects, and then to combine them into one total image. The multiple viewpoint rendering (MVR) is one of various methods for integral imaging pickup. This method, however, has the computing and rendering time problem for obtaining element images from a lot of elemental lenses. In order to solve the problems, in this paper, we propose a parallel MVR (PMVR) method to generate elemental images in a parallel through distribution of elemental lenses into multiple threads simultaneously. As a result, the computation time of integral imaging using PMVR is reduced significantly rather than a sequential approach and then we showed that the PMVR is very useful.

**Keywords:** Elemental Imaging, Multiple viewpoint rendering, Thread , Parallel.

### 1. INTRODUCTION

The most of three-dimensional (3D) visualization systems are rendering 3D objects into two-dimensional (2D) images through 3D graphics pipelines with information of a specific camera and 3D objects. To overcome 2D limitation, the stereoscopic display, which is similar to human visual mechanism, has been used to get the effect of 3D visualization. Due to the use of special glasses in the stereoscopic display, many researchers have studied about auto-stereoscopic display which does not require any special glasses for viewing 3D images. Integral imaging, which was first proposed by Lippmann [1], is one of the auto-stereoscopic 3D display techniques. Integral imaging system stores 3D information of the objects into a set of elemental images using a lens array, and then displays 3D images by integrating them through the lens array again. The system can be divided into pickup part

which captures the elemental images of the 3D object, and visualization part which integrates the elemental images into the 3D images [1]. The pickup part is performed using optical system to capture the elemental images of the real existing 3D objects. For the computer generated 3D objects, however, the pickup process can be performed computationally without real optical system. This computational pickup draws much attention recently, due to its versatility in dealing with 3D information of the computer generated objects. Several methods for the computational pickup have been proposed [2-10]. Most famous methods of them are point retracing rendering (PRR) [2], parallel group rendering (PGR) [3], viewpoint vector rendering (VVR) [4], and multi view point rendering (MVR) [5]. PRR is a simple method which draws the set of elemental images into the display image point by point, however, it is unsuitable to use the method for real-time rendering due to the heavy computation time. PGR uses the viewing characteristics of the focused mode where each elemental lens appears as a pixel to the observer. In PGR, a set of elemental images is obtained from the imaginary scenes observed in a certain direction and are named directional scenes. The number of directional scenes is the same as that of the

*This is an excellent paper selected from the papers presented at ICCV 2008.*

\* Corresponding author. E-mail : khyoo@chungbuk.ac.kr

Manuscript received Jun.09, 2009 ; accepted Oct. 25, 2009

display pixels in the elemental-lens area, and the directions correspond to the vectors from each display pixel to the center of the corresponding elemental lens. Therefore, the elemental image generation is faster and less affected by the numbers of elemental lenses and 3D object polygons. MVR is a method which applies the traditional graphics pipeline process to do a real pickup process. Specially, in MVR, each elemental image is captured by a virtual camera, which is provided by advanced computer programming tools such as OpenGL graphics library [11]. The merits of MVR are simplicity and accuracy of elemental-image generation. In MVR, the speed of image mapping is not influenced by the size of the integrated image, but is still predominantly affected by the numbers of elemental lenses and object polygons that determine the image quality of the 3D object. Owing to these limitations, MVR is not desirable for a real-time complex computer-graphics rendering system. These methods including PRR, PGR, VVR and MVR have been upgraded within aspect to both of computation time and integral imaging quality [6-10]. In this paper, we propose a novel computational pickup method, parallel multi view point rendering (PMVR), to enhance the computation time of the elemental images by using multiple threads.

Section 2 describes the rendering method of previous MVR and its problem. The design and implementation method of PMVR will be illustrated in Section 3. Section 4 shows the implementation results of the proposed method, and finally, the results of this paper are summarized and future research directions will be presented.

## 2. MVR OF INTEGRAL IMAGING

Integral imaging is the method which generates a set of elemental images with respect to 3D objects from a specific viewing camera inside any elemental lens in a lens array [4].

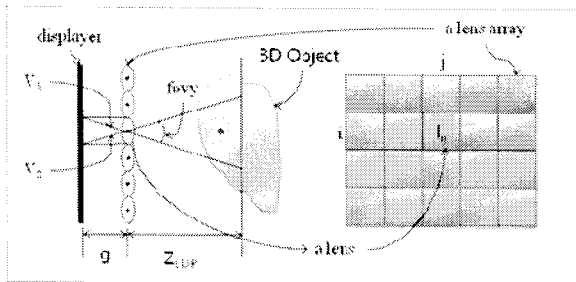


Fig. 1. The conceptual model of integral imaging

As shown in Figure 1, there are 3D objects and a lens array in integral imaging system. Lenses in the lens array are arranged in the type of square grid in the three dimensional (3D) space. Assume that there are  $N \times N$  lenses in a lens array. Let  $l_{ij}$  be a lens located  $i$ th row and  $j$ th column of the lens array. In integral imaging, there is a perspective camera which is located at the center of each lens. In general, the perspective camera has fovy, aspect ratio, near and far planes. Prior to compute the fovy information, we have to decide the location of displayer on which integral imaging will be rendered. Let  $g$

be gap between a given lens array and displayer. To obtain the  $g$ , we give two parameters,  $f$  and  $Zcdp$ . Here  $f$  means the focal length of the lens and  $Zcdp$  does the distance from lens to virtual 3D objects. In integral imaging system,  $Zcdp$  is typically designated as a value from  $2f$  to  $5f$  [1]. With these three parameters,  $f$ ,  $Zcdp$  and  $g$ , the following equation (Eqn 1) can be obtained [1]:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{Z_{CDP}} \text{-----(Eqn 1)}$$

From Equation (Eqn 1),  $g$  can be calculated as the following equation (Eqn 2).

$$g = \frac{Z_{CDP} f}{f - Z_{CDP}} \text{-----(Eqn 2)}$$

The location of displayer will be set according to the calculated  $g$  distance away from the location of a lens array as illustrated in Fig.1. A specific lens in lens array is mapped into the region on displayer by doing orthogonal projection from the lens to displayer. We can obtain two vectors  $V_1$  and  $V_2$ , which are formed by connecting the center of the lens and two boundary points in one side of the corresponding region on displayer, respectively. With these two vectors, the fovy  $\theta$  designated at camera of the lens can be calculated by using Equations equations (Eqn 3) and (Eqn 4).

$$\|V_1\| \|V_2\| \cos \theta = V_1 \cdot V_2 \text{-----(Eqn3)}$$

$$\theta = \cos^{-1} \left( \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|} \right) \text{-----(Eqn 4)}$$

As described above, we can set single virtual camera at the center of each lens in a lens array. All virtual cameras will be stored into an array of the same size as the lens array. Clearly, if a lens array has  $N \times N$  size, there are  $N \times N$  virtual cameras in the lens array. Each virtual camera stored at each element of the array will feature the same as general perspective camera which have the following parameters: position, fovy, aspect ratio, which defined as the ratio of horizontal size over vertical size of the corresponding region on displayer, and visible depth range. From the facts, generating an elemental image with respect to 3D objects is to render 3D objects from the corresponding single viewer in MVR, which can be transformed into the rendering one from the perspective camera with the computed parameters. Therefore, after we generate the single elemental image sequentially  $N^2$  times, elemental image for a lens array with  $N \times N$  size can be obtained by combining these single images. The MVR method described until now is as follows:

Step 1: Compute the parameters(position, fovy, aspect ratio, visible depth range) for all perspective virtual cameras in a  $N \times N$  lens array.

For ( $i = 0; i < N; i++$ ) {  
 For ( $j = 0; j < N; j++$ ) {  
 Step 2: specify each perspective virtual camera

corresponding to single  $(i, j)$ -element lens in the  $N \times N$  lens array with the calculated parameters.

Step 3: generating single element image,  $EI[i, j]$ , by rendering 3D objects from the perspective virtual camera.

```

}
}

```

Step 4: Obtain the final elemental image,  $FEI$ , by combining all  $EI[i, j]$ 's.

Now, consider how the method can be implemented by using general graphics library, OpenGL [11]. Since the MVR method can be directly implemented by performing rendering pipeline with respect to every perspective camera with the parameters, those are, fovy, aspect ratio, and depth range, it is very easily done by using two OpenGL functions, `glViewport` and `glPerspective`. Clearly, if  $O(RT)$  is the rendering time for given 3D objects from single camera, the total rendering time for generating the final elemental image is  $O(N^2 RT)$  time.

### 3. PARALLEL MVR OF INTEGRAL IMAGING

Due to the sequential processing of MVR, the computation time increases in proportion to the number of elemental lenses so that rendering of the final elemental images is done inefficiently. In this section, we propose a parallel multi viewpoint rendering (PMVR) method in which the rendering will be performed in a parallel simultaneously through distribution of elemental lenses into multiple threads as shown in Figure 2.

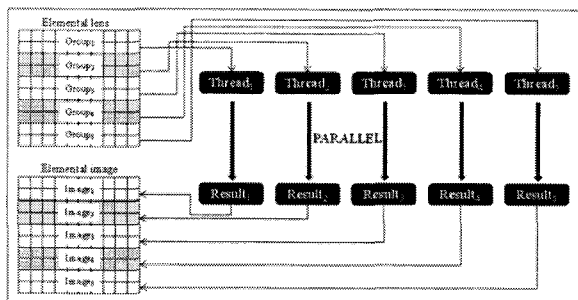


Fig. 2. The process of PMVR using threads

In general, the input data for integral imaging are given from users to generate the final elemental images. The input data include the number of elemental lenses, types of lens, pitch size of a lens, focal length,  $Z_{CDP}$ , and etc. Assume that  $N$  threads are created for a  $N \times N$  lens array. Clearly, the  $N$  lenses of them will be uniformly distributed into single thread, and then each thread will be in charge of the rendering of  $N$  elemental images. In order to do it, first of all, each thread computes the related all parameters for perspective cameras such as position, fovy, and aspect ratio to the  $N$  lenses, and then will generate the corresponding  $N$  elemental images. Figure 2 is illustrating the details. The final elemental image will be gotten by combining the elemental images created by each thread. The following method describes procedure for processing work assigned at  $i$ th

thread:

Step 1: Compute the parameters(position, fovy, aspect ratio, visible depth range) for  $N$  perspective virtual cameras in  $i$ th row of a  $N \times N$  lens array.

```

For (j = 0; j < N; j++) {

```

Step 2: specify each perspective virtual camera corresponding to single  $(i, j)$ -element lens in the  $N \times N$  lens array with the calculated parameters.

Step 3: generating single element image,  $EI[i, j]$ , by rendering 3D objects from the perspective virtual camera.

```

}

```

Step 4: Obtain the elemental image  $E[i]$  by combining  $N$   $EI[i, j]$ 's,  $j=1, \dots, N$ .

Using the above method, the final elemental images,  $FEI$ , will be obtained by combining  $N$  elemental images,  $E[i]$ ,  $i=0, \dots, N$ .

### 4. EXPERIMENTAL RESULTS

The proposed PMVR has been implemented by MS Visual Studio 2008, OpenGL, and glut on Windows Server Standard 2008 PC of Intel Core2 Quad CPU 2.4 GHz and 3.00GB RAM. In this paper, a 3D human model, which is composed of 3983 vertices, 21372 edges and 7124 polygons, is used for testing integral imaging pickup by the proposed PMVR. For given  $10 \times 10$  lens array, Figure 3 shows integral imaging generated by using PMVR with respect to the 3D human model.



Fig. 3. An integral imaging obtained from  $10 \times 10$  lens array for a 3D human model.

Figure 4 shows the rendering time of MVR method according to an increase of elemental lens. When the number of elemental lens is  $N \times N$  and the rendering time from single perspective camera is  $T$ , the rendering time is  $O(N^2 T)$  because that single view rendering is called  $N \times N$  times. Figure 5 shows the rendering time of PMVR versus the number of elemental lens when  $N$  threads are used for  $N \times N$  lens array.  $N$  threads do not perform  $N \times N$  rendering operators in a sequence, but perform the tasks simultaneously after dividing given work into proper smaller tasks. Through doing these multitasks as mentioned above, the rendering time of PMVR was enhanced rather than traditional MVR as shown in Figure 5.

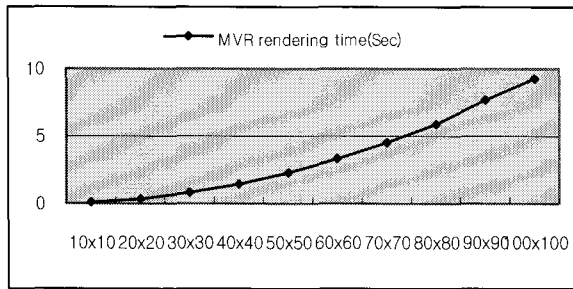


Fig. 4. The rendering time of MVR versus the number of elemental lens

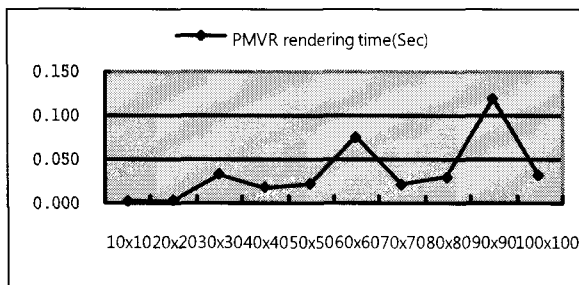


Fig. 5. The rendering time of PMVR versus the number of elemental lens

## 5. CONCLUDING REMARKS

In this paper, we consider the problem for processing integral elemental imaging in a real-time, which is very critical to visualize the 3D graphics data. The known MVR algorithm can be used to generate elemental images of high quality, however, and it has a heavy load for the rendering time according to an increase of the number of elemental lens. In order to reduce the heavy rendering time, in this paper, we propose a parallel MVR method in which generation of elemental images can be performed in parallel by distributing the rendering load into several threads. We can check that PMVR rendering time is significantly smaller than MVR rendering time.

The future directions are finding parallel algorithms which can be applied to distributed computing environments such as grids.

## ACKNOWLEDGEMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD)" (The Regional Research Universities Program/Chungbuk BIT Research-Oriented University Consortium) and by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation(KOTEF) through the Human Resources Training Project for Regional Innovation.

## REFERENCES

- [1] G. Lippmann, "La photographie integrale," *C. R Academic Science*, vol. 146, pp. 446-451, 1908.
- [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, 1887, pp.1598-1603.
- [3] Y. Igarashi, H. Murata and M. Ueda, "3D display system using a computer generated integral photography," *Japanese Journal of Applied Physics*, Vol.17, 1978, pp.1683-1684.
- [4] R. Yang, X. Huang and S. Chen, "Efficient rendering of integral images," *Proceedings of SIGGRAPH (ACM Press, New York, 2005)*, 2005.
- [5] M. Halle, "Multiple Viewpoint Rendering," *Proceedings of SIGGRAPH (ACM Press, New York, 1998)*, 1998, pp. 243-254.
- [6] Sung-Wook Min, Kyoung Shin Park, Binara Lee, Yongjoo Cho and Minsoo Hahn, "Enhanced Image Mapping Algorithm for Computer-Generated Integral Imaging System," *Japanese Journal of Applied Physics*, Vol. 45, No. 28, 2006, pp. L744-L747.
- [7] Jang-II Ser, "A study on the properties of an elemental image depending on the shape of elemental lens and the pick-up method in the integral imaging," *Journal of Telecommunication and information*, Vol. 10, 2006.
- [8] Sung-Wook Min, "Three-dimensional image processing using integral imaging method," *Optical Society of Korea summer Meeting 2005* (7.14~15, 500).
- [9] Jung-Young Son, Vladimir V. Saveljev, Jae-Soon Kim, Sung-Sik, Bahram Javidi, "Viewing zones in three-dimensional imaging systems based on lenticular, parallax-barrier, and microlens-array plates," *Applied Optics*, Vol. 43, 2004, pp. 4985-4992.
- [10] Jang-II Ser, Seung-Ho Shin, "Depth-Conversion in Integral Imaging Three-Dimensional Display by Means of Elemental Image Recombination," *Hankook Kwanghak hoeji*, Vol. 18, No. 1, 2007.
- [11] <http://www.opengl.org>, OpenGL library.



**Young-Hee Jang**

He is a M.S. learner of department of IIE(information industrial engineering) at Chungbuk National University, Korea. He received the B.S. in Computer Education from Chungbuk National University, Korea in 2007. His research interests include e-Learning, u-Learning,

Multimedia, Computer graphics.



**Chan Park**

He is a PhD learner of department of IIE(information industrial engineering) at Chungbuk National University, Korea. He received the B.S. and M.S. in Computer Education from Chungbuk National University, Korea in 2003 and 2007, respectively. His research interests

include e-Learning, u-Learning, Multimedia, Computer graphics.



**Jae-Hyeung Park**

He is currently an assistant professor in the School of Electrical & Computer Engineering, Chungbuk National University, Korea. His research area includes the autostereoscopic / super-multi-view three-dimensional display and relevant image processing. He has been

serving as a program committee member of Society for Information Display (SID) since 2007.



**Nam Kim**

He is currently a professor in the School of Electrical & Computer Engineering, Chungbuk National University, Korea. His research area includes optical information processing, fiber communications and stereoscopic display. He has been serving as a board/member

of Optical Society of Korea(OSK) since 1998.



**Kwan-Hee Yoo**

He is a professor who is working for department of Computer Education and department of IIE(information industrial engineering) at Chungbuk National University, Korea. He received the B.S. in Computer Science from Chonbuk National University, Korea in 1985, and

also received M.S., Ph.D. in computer science from KAIST (Korea Advanced Institute of Science and Technology), Korea in 1988 and 1995, respectively. His research interests include u-Learning systems, computer graphics, 3D character animation, dental/medical applications.