Light-Field Rendering In the View Interpolation Region without Dense Light-Field Reconstruction

Young Ju Jeong
Sookmyung Women’s University, Seoul, South Korea
yjeong@sookmyung.ac.kr

Abstract. Different scenes according to a user’s movement are important to provide a real three-dimensional (3D) experiment. The different scenes, called virtual viewpoint images, can be rendered by 3D structures. However, 3D structure reconstruction from camera images requires high computational complexity and computation time. A light-field rendering algorithm uses the light fields captured by the camera array as the input and selects the light rays for the virtual viewpoint image. Herein, a light-field rendering algorithm in the view interpolation region using line interpolation is introduced. The proposed algorithm functions well to generate planar viewpoint images and zoomed in/out viewpoint images.

1. Introduction

Three-dimensional (3D) displays have been developed for many decades to provide a real 3D feel to users. Users can enjoy 3D environments without using glasses, e.g., autostereoscopic 3D displays [1-3]. Three-dimensional glasses required for virtual reality and augmented reality devices have been developed to provide a vivid reality [4-5]. These wearable 3D devices are realistic because they can provide different scenes according to the user’s motion and movement [6-7].

Three-dimensional scene changes according to user’s movement can be implemented by the virtual view rendering algorithm [8-10]. Through wearable 3D devices, we can conceive 3D environments, e.g., virtual reality and augmented reality, owing to binocular vision. Binocular vision enables the user to conceive a 3D environment by showing different stereo images to the left and right eyes. Stereo images generated by the virtual view rendering algorithm provide a 3D environment, and new viewpoint stereo images for our new position can provide 3D scene changes in the 3D environment.

To generate virtual view images, two main methods can be used: structure-based rendering and image-based rendering. Structure-based rendering contains the depth information of all objects in the images. Virtual view images using structures can be generated by perspective projection [11-12]. Graphic contents are typically composed of 3D structures. However, 3D structure reconstruction from a camera-captured real scene is highly complex. To avoid these challenges in estimating the 3D structures, image-based rendering algorithm is proposed [13-14]. Light-field representation introduces new and effective parameter representations and proposes the rendering algorithm [15-17].

Herein, the virtual view rendering algorithm in the view interpolation region without dense light-field reconstruction is introduced. The proposed algorithm defines the view interpolation region. In the view interpolation region, all virtual view images can be rendered by the input images. The light-field rendering equation in the view interpolation region is proposed using light-field representation.
2. Light-field representation

Light-field representation presents a light ray using four parameters \((u, v, s, t)\) that passes through two parallel planes, \(\Pi\) and \(\Omega\), as shown in Figure 1 [15]. Using light-field representation, many cameras in the array span the light fields in the 3D space. Typically, a camera array is located on the \(\Pi\) plane. Images captured by cameras are components of light fields, and the pixel value in a camera is a light ray passing through the \(\Pi\) and \(\Omega\) planes. The position of \((u, v)\) is defined by the camera center position; \((s, t)\) is defined by the pixel location of the images. Herein, \((u, v)\) is used for the camera array images, and \((x, y)\) for the pixel location of each camera array image.

The light fields are composed of camera arrays. A camera array located in a plane captures the same scene. Figure 2 shows the camera array that captures the light field. The camera captures the light rays that pass through the camera center. Each image captured by the camera is slightly different, having horizontal and vertical motion parallax. The horizontally located cameras possess horizontal motion parallax and the vertically located cameras possess vertical motion parallax.

3. Light-field rendering

To render a virtual view image using the input camera array image, a light-field rendering algorithm for the planar view and zoomed in/out rendering is introduced.

3.1. Light-field rendering principle

A camera array captures the light field in 3D space. With a sufficient number of light rays, virtual view images are rendered by sampling the light of the input light-field images. Figure 3 shows the rendering principle. The virtual image located at the green camera can be generated by the sampling of
light-field images in the camera array. The green rays in the virtual view image can be obtained by selecting the light ray of the input camera array images. The green light rays of the virtual image, shown in Figure 3, are not sufficiently dense. If denser cameras in the array system are used, the green light rays then become sufficient. The basic principle of light-field rendering is the selection of the desired light ray among the given light rays because the cameras in the 3D space share the same light ray.

3.2. Planar view rendering
The virtual view images located on the camera array are synthesized using bilinear interpolation. This is because the light-field representation images are composed of lines, and a point in the 3D space is expressed by a line in the light-field representation image. Therefore, a higher-order interpolation algorithm does not require light-field representation images.

Figure 4 shows the input camera array images [18] and the (u, x) and (v, x) light-field representation images. The camera array is 7 × 7. The horizontal and vertical axes of the image are x and y, respectively. Four parameters, (u, v, x, y) are used, instead of (u, v, s, t) in Figure 1, for simplicity. An (s, t) plane is a typical plane and its coordinate values are global. Meanwhile, an (x, y) plane is typical but its coordinate values are not global; instead, its coordinate value depends only on each image. The plane spanned by the u and v axes is the camera array plane. Each image by the camera is defined by x and y. The (u, x) and (v, x) light-field representation images contain only lines. A line is a 3D point, and the gradient of the line represents the depth of a 3D point.

The virtual view image located on (u, v) is rendered by

\[ I_v(x, y) = (1 - \alpha)(1 - \beta)I_1(x, y) + \alpha(1 - \beta)I_2(x, y) + (1 - \alpha)\beta I_3(x, y) + \alpha\beta I_4(x, y) \] (1)

where \( I_v(x, y) \) is a light ray value of (x, y) position in the virtual view image; \( I_1, I_2, I_3, \) and \( I_4 \) are the neighboring four input images; \( \alpha \) and \( \beta \) are the distances between the neighboring images, as shown in Figure 5.

Figure 4. Input camera array images and (u, v, x, y) light-field representation images
3.3. View interpolation region

With the input camera array images, not only the virtual images on the same plane but also the virtual images out of the plane can be generated, as shown in Figure. 3. The possible view interpolation region (extra information not required) is calculated, and then the rendering algorithm is suggested.

The view interpolation region can be obtained in the virtual viewpoint image region, where the virtual view images can be completely synthesized with the input camera array light rays. The view interpolation region is defined by the field of view of the input camera array. In the 3D space, the interpolation area is where two horns are in contact with each other around the camera array plane when the input camera array is a quadrangle.

Figure 6 shows the view interpolation region when \( v \) is fixed. The interpolation region with a fixed \( v \) value can be defined by the same principle. As shown in Figure. 6(b), the most forward virtual view position can be defined by the intersecting point, i.e., the field of views of the first (the most left) and the last (the most right) camera. The bottom images of Figure. 6(b) shows the \( x,u \) light-field representation image. The horizontal line of a virtual image can be generated by the light rays of the input camera images when \( v \) is the same. Plane \((x,u)\) is demonstrated because \( v \) is fixed. The first light ray of the virtual view is the first light ray of the last camera. The last light ray of the virtual image is the first light ray of the first camera, as shown in Figure. 6(b). With the same principle, the most backward virtual image position is defined by the intersecting point of the first and the last cameras, as shown in Figure. 6(c). In this case, the first light ray of the virtual image is the first light ray of the first camera, and the last light ray of the virtual images is the last light ray of the last camera.
3.4. Zoom in/out rendering

The virtual view images in the interpolation region can be rendered by the input camera array images. In the light-field representation, a line rendering in the virtual view image is obtained by the line selection. Figure 7 shows the camera arrays and the desired virtual viewpoint location. In the 3D space, the camera array plane is defined as \( u, v \) and the depth location axis as \( z \). The virtual images with \((a, b, c)\) location for \((u, v, z)\) coordinate can be generated by line sampling in the \((u, x)\) and \((v, y)\) planes. In the \((x, u)\) light-field representation image, the line passed \( (\frac{W}{Z}, a) \) with \( \frac{cU}{WZ} \) gradient. In the \((y, v)\) light-field representation image, the line passed \( (\frac{H}{Z}, b) \) with \( \frac{cV}{HZ} \) gradient. \( W \) and \( H \) are the width and height of an image, respectively. \( U \) and \( V \) are the width and height of a camera array, respectively. Using the line in the light-field space, \( u \) and \( v \) are calculated as follows:

\[
\begin{align*}
u &= \frac{cU}{ZW} x - \frac{cU}{2Z} + a \\
v &= \frac{cV}{ZH} y - \frac{cV}{2Z} + b.
\end{align*}
\]  

The light ray of the virtual image is calculated from Eqs. (2) and (3).

\[ I(x, y) = I_{u,v}(x, y) \]  

where \( I_{u,v}(x, y) \) is the light ray of the \((x, y)\) pixel of the \((u, v)\) view point image. Typically, \( u \) and \( v \) are not integers, and \( I_{u,v}(x, y) \) can be calculated by Eq. (1).

4. Experimental results

![Figure 8](image_url)

**Figure 8.** Happy Buddha zoomed in/out rendering results (a) zoomed-in image, (b) input image, (c) zoomed-out image
The experimental results demonstrate that the virtual view images by the proposed algorithm are well defined and generated. In the experimental results, the zoomed in/out image results are discussed because the virtual view image by the zoomed in/out rendering contains the planar view light ray. Figure 8 shows the zoomed in/out image rendering result of the happy Buddha [18]. The input camera array of the happy Buddha is $7 \times 7$, as shown in Figure 4. Figure 8(b) shows an input image in the $(u,v)$ plane. Figure 8(a) shows the most forward image, i.e., the zoomed-in image, as shown in Figure 6(a). Figure 8(c) shows the most backward image, i.e., the zoomed-out image, as shown in Figure 6(c). According to the backward movement, we could observe the background occlusion region area. The background character is occluded in the zoomed-in image (Figure 8(a)); the occluded area of the character is disoccluded such that we can observe the zoomed-out image (Figure 8(c)). Figure 9 shows the zoomed in/out images of a treasure chest [19]. The input camera array of the happy Buddha is $17 \times 17$. The view interpolation region of the treasure chest is larger than that of the happy Buddha because the treasure chest has broader camera array area. According to the forward movement, the object size becomes smaller. The perspective amount of the treasure chest becomes larger according to the zoomed in motion.

5. Conclusions
Virtual viewpoint image synthesis provides a new scene according to a user’s movement. For applications of augmented reality and virtual reality, the virtual viewpoint rendering algorithm is crucial to provide a more natural 3D reality. To render virtual viewpoint images, the 3D structure-based rendering algorithm is broadly used. The light-field rendering algorithm is simple and light because it does not require a 3D structure reconstruction. Without a 3D structure, the selection of light ray in light fields captured by an input camera array can generate a virtual view image. The light-field rendering algorithm in the planar view and zoomed in/out view images using line interpolation is introduced and implemented. The virtual view images are generated in the interpolation region. For future work, complex line interpolation that uses a semi-3D structure algorithm will be studied for sparse input camera array images. Moreover, light-field images for toe-in or laterally shifted input images will be studied.

Acknowledgments
This work was supported by the Technology Innovation Program (20001700, International Standard Development of Performance Measurement Method for AR 3D HUD) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea)

References
[1] N. A. Dodgson 1996 Analysis of the viewing zone of the Cambridge autostereoscopic display Appl. Optics, 35, No. 10 pp 31–6
[2] N. A. Dodgson 2005 Autostereoscopic 3D displays Computer 38 pp 31–6
[3] T. Balogh, Z. Nagy, P. T. Kovács, and V. K. Adhikarla 2013 Natural 3D content on glasses-free light-field 3D cinema IS&T/SPIE Electron. Imaging 8648
[4] A. Wexelblat 1993 The Reality of Cooperation: Virtual Reality and CSCW Virtual Reality: Applications and Explorations. Edited by A. Wexelblat. Boston
[5] R. Azuma, "SIGGRAPH95 Course Notes: A Survey of Augmented Reality", Los Angeles, Association for Computing Machinery, 1995.

[6] H. Ishii, M. Kobayashi and K. Arita 1994 Iterative Design of Seamless Collaboration Media Communications of the ACM 37 pp 83-97.

[7] U. Neumann, S. You, Y. Cho, J. Lee and J. Park 1999 Augmented Reality Tracking in Natural Environments Mixed Reality - Merging Real and Virtual Worlds pp 101-30

[8] S. Moezzi, L. Tai, and P. Gerard. 1997 Virtual view generation for 3d digital video. IEEE Multimedia, 4 pp 18–25

[9] T. Kanade, P. Rander, and P. Narayanan 1997 Virtualized reality: Constructing virtual worlds from real scenes IEEE Multimedia 4 pp 34–47

[10] J. Konrad 1999 View reconstruction for 3-D video entertainment: issues, algorithms and applications Image Processing and its Applications

[11] C. Fehn 2004 Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV Proc. SPIE Stereoscopic Displays and Virtual Reality Syst. 5291 pp 93–104

[12] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand 2008 Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems, IEEE International Conference on Image Processing

[13] H.-Y. Shum, S.-C. Chan, and S. B. Kang 2006 Image-Based Rendering. Springer-Verlag

[14] A. M. Siu and R. W. Lau 2005 Image registration for image-based rendering IEEE Trans. Image Process. 14 pp 241–52

[15] M. Levoy and P. Hanrahan 1996 Light field rendering ACM SIGGRAPH pp 31–42

[16] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen 1996 The lumigraph ACM SIGGRAPH pp 43–54

[17] M. Levoy 2006 Light fields and computational imaging IEEE Computer 39 pp 46–55

[18] Gordon Synthetic Light Field Archive.

[19] The (New) Stanford Light Field Archive.