

Depth-of-Field Guided Rendering for Light Field Displays

Kamran Akbar, Robert Bregovic
Tampere University, Tampere, Finland

Abstract

Light field displays have a finite angular and spatial resolution, which limits the display's depth-of-field, that is, the depth range around the display screen in which the display can visualize a 3D scene with the maximum spatial resolution. This limitation causes aliasing artifacts in the parts of the scene that are outside of that range, resulting in a distorted appearance. The aliasing artifacts can be mitigated by properly blurring those parts, with blurring preferably done at the rendering stage. Though methods for rendering a single view with a correct depth of field exist, using those methods for rendering a large light field is computationally heavy. In this paper we propose a method for simultaneously rendering multiple adjacent views in a light field, with each of them having the required depth of field. By means of examples, we show that the proposed method can render a desired light field several times faster than methods for rendering a single view, without compromising on the overall rendered quality.

Introduction

Light field is a vector function that describes the light intensity in all direction at every point in space [1]. Light field displays are a category of 3D displays that aim at a 'perfect' visualization of a 3D scene by recreating the underlying light field describing the scene. To visualize a 3D scene with smooth motion parallax, a dense light field, that is, a large number of light rays needs to be reconstructed. However, current light field displays can reproduce only finite number of light rays which limits their performance. The limitations are quantified through the display's angular and spatial resolutions [2], [3], [4]. The angular resolution corresponds to the number of reconstructed views over the display's field of view and the spatial resolution is the maximum resolution of a single view [5]. The finite angular and spatial resolutions constrain the depth range around the display screen which can visualize a 3D scene with the maximum spatial resolution. This range is referred to as the display's depth of field (DoF). Consequently, the parts of the visualized 3D scene that are outside of the display's DoF will appear distorted due to aliasing [2], [5]. To mitigate the aliasing distortions, the light field needs to be preprocessed, that is, the mentioned scene parts must be removed or appropriately blurred.

The existing methods that perform blurring directly on the generated light field [2], [6], [7] require as input a densely- or moderately-sampled light field, which preparation is time consuming and memory demanding. Alternatively, there are many methods that perform blurring at the rendering stage [8]- [15] with the aim of directly rendering an image with the desired DoF. This includes approaches based on path tracing, multi-view rendering, gathering, scattering, and multi-layering. DoF rendering with path tracing is done by tracing rays in both pixel and aperture domains [8], [9]. Multi-view rendering is an approach that uses

several adjacent views for rendering DoF. This can be done by warping and merging several adjacent views [10] or sampling and integrating several images rendered over a simulated camera lens [11]. Both of those approaches are time-consuming and unsuitable for real-time applications. Gathering and scattering are post-processing techniques that use the final rendered image and the corresponding depth map to generate DoF [12], [13]. However, naive implementation of these techniques can cause color bleeding and depth discontinuity artifacts due to occlusions. Multi-layering approach tackles the problem of occlusions by storing different regions of the scene geometry in a multi-layered texture. The multi-layering approach can be combined with methods using gathering and scattering to avoid color bleeding and depth discontinuity artifacts [14], [15].

To visualize a 3D scene on a light field display, a complete light field needs to be rendered. All the existing methods render each view individually, which is computationally expensive and inefficient. In this paper, to accelerate the light field rendering, we propose a method which simultaneously renders several adjacent views. Through examples, we demonstrate that this approach is faster than the existing methods without compromising the overall quality.

The rest of the paper is structured as follows. In Section 2, we review the concepts of DoF in photography, DoF in light field displays, and overviews the multi-layer DoF rendering with tiled splatting approach. In Section 3, we describe the proposed method for fast light field rendering with a desired DoF. In Section 4, we evaluate the proposed method by means of examples. Finally, in Section 5, we present some concluding remarks.

Background

Depth of Field in Photography

In photography, the DoF is the range around the focus plane that appears sharp on the image sensor. This range is also referred to as the focused region in the captured image. The DoF depends on several factors, e.g., aperture size, focal distance, focal length of the lens, and sensor pixel size. Outside of the DoF, each point is mapped to a circle on the image sensor with a diameter greater than one pixel. This circle is referred to as the circle of confusion (CoC). The CoC diameter can be estimated by approximating the camera lens with a thin lens model. The model is illustrated in Figure 1 and is quantified with the Gaussian thin lens equation [16]:

$$\frac{1}{f} = \frac{1}{z_f} + \frac{1}{z'_f}, \quad (1)$$

where f is the focal length, z_f is the focus distance, and z'_f is the distance between the lens and the image sensor. The CoC

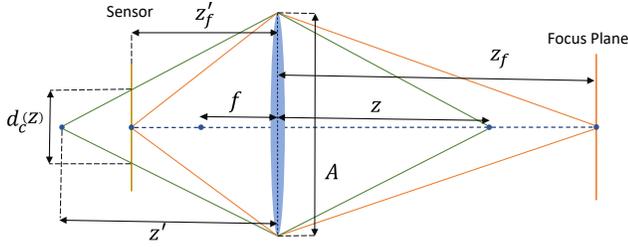


Figure 1. Thin lens model.

diameter $d_c(z)$ of a point at distance z can be estimated as [16],

$$d_c(z) = \left| \frac{Af(z - z_f)}{z(z + z_f)} \right| \cdot \frac{n_{px}}{l}, \quad (2)$$

where A is the aperture diameter, n_{px} is the sensor horizontal resolution, and l is the sensor width. An example of the CoC diameter change with distance is shown in Figure 2.

Depth of Field in Light Field Displays

A light field display, due to its finite spatial and angular resolution, can show 3D content with the maximum spatial resolution only in a limited depth range around the screen. This range is denoted as the display's DoF [2]. Due to the limited DoF, the smallest spatial feature size, $p(z)$, that a display can visualize at distance z from the viewing plane is [17],

$$p(z) = \frac{|z - z_f| \tan(\alpha_s)}{p_0}, \quad (3)$$

where α_s is the display's angular resolution, s_w is the display's screen size, and z_f is the distance of the screen plane from the viewing plane. p_0 is the pixels size at the screen plane that corresponds to the size of the smallest feature that the display can visualize and can be evaluated as

$$p_0 = \frac{s_w}{n_{px}}. \quad (4)$$

The parameter n_{px} corresponds to the display's maximum horizontal spatial resolution. In this paper we assume that n_{px} is also equal to the camera sensor horizontal resolution as discussed in [6]. The display/viewing parameters are illustrated in Figure 3 and the change in the smallest spatial feature size $p(z)$ reproduced by the display, as given by Equation 3, is illustrated in Figure 2. Though, so far we only referred to the horizontal direction, same equations apply also in the vertical direction.

Multi-Layer Depth of Field Rendering with Tiled Splatting

To render a light field with a given DoF in real-time, we will build upon the multi-layer DoF rendering with tiled splatting approach that renders a single image with a desired DoF [15]. As illustrated in Figure 5, this approach contains five stages: generation, reduction, tiling, sorting, and accumulation.

In the generation stage the scene geometry is stored into multiple layers with the first layer containing the initial projection of the scene onto the camera sensor and every other layer containing

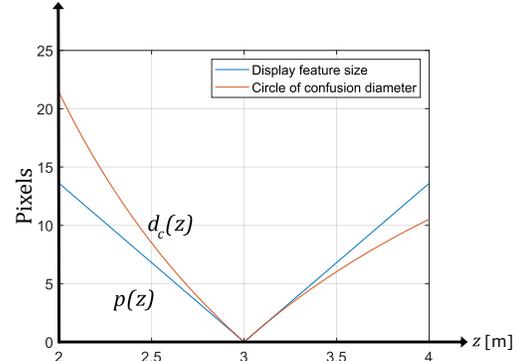


Figure 2. Change of CoC diameter $d_c(z)$ and display's feature size $p(z)$ with distance for $f = 61.7$ mm, $A = 48$ mm, $z_f = 3$ m, $\alpha_s = 0.95^\circ$, and $p_0 = 1.2$ mm.

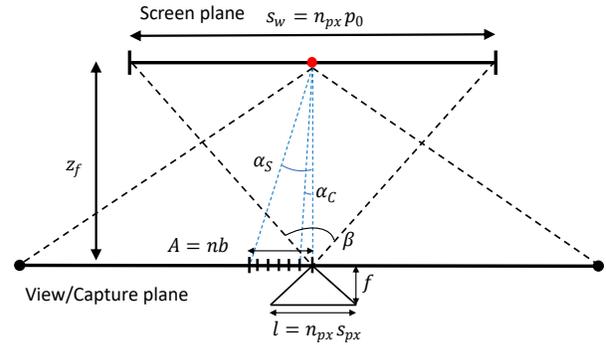


Figure 3. Simplified light field display model and corresponding view setup.



Figure 4. Sponza scene [18] rendered into multiple layers.

the scene region occluded by the previous layer. Figure 4 illustrates the result of multi-layering. In the reduction stage, adjacent pixels inside the multi-layered image with similar color, depth, and distance from the focus plane are merged by averaging their color and depth values. In the tiling stage, the image plane is divided into several region denoted as tiles. For each pixel inside each tile the CoC radius $\frac{d_c(z)}{2}$ is calculated and its overlap is checked with the neighbouring tiles. In the case of overlapping, the pixel's color, depth, and location are copied into the overlapping tiles. In the sorting stage, all pixels in each tile are sorted based on their depth value.

In the accumulation stage, the image with the desired DoF is generated. Throughout this stage, for each pixel location $(u, v)_i$,

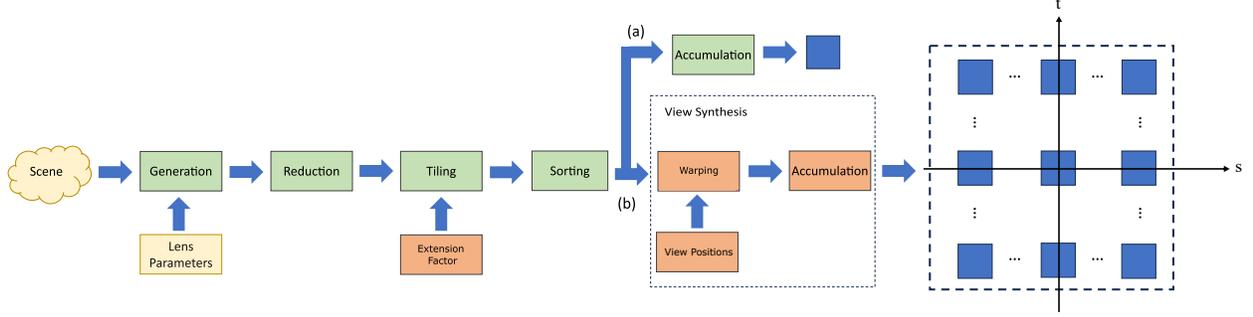


Figure 5. Pipeline of multi-layer DoF rendering with tiled splatting. Path (a) corresponds to the original pipeline, and path (b) illustrates the modifications for rendering several adjacent views simultaneously. The yellow, green and orange blocks correspond to the input, original, and modified stages of the pipeline, respectively. The blue blocks illustrate the synthesized views.

the corresponding tile is found. Next, for each pixel residing in the tile, its CoC radius $\frac{d_c(z_j)}{2}$ and the distance $\delta_{i,j}$ to the pixel location $(u,v)_i$ is computed. The color contribution of all pixels inside the tile with CoC radius more than the computed distance $\delta_{i,j}$ is evaluated as,

$$\Psi_i = \sum_{j=1}^n \left(\alpha_j \Psi_j \prod_{k=1}^{j-1} (1 - \alpha_k) \right), \quad (5)$$

where α_j and Ψ_j refers to the contributing pixel j opacity and color, respectively. The pixel opacity is computed based on CoC radius. More detailed explanation of this approach is given in [15].

Methodology

In this paper, to render a light field for a given light field display, we first calculate the required camera parameters which will provide the correct DoF in each view. Second, we propose a rendering pipeline to render several adjacent views simultaneously.

Selection of Camera Parameters

For rendering a light field with a DoF as required by a light field display, the camera parameters such as lens aperture and field of view need to be computed. This requires establishing a relation between the camera and display parameters. Following the discussion in [6], for a densely sampled light field (DSLFL), the aperture filtering method computes the aperture diameter as (see Figure 3 for illustration),

$$A = \left[\frac{\alpha_s}{\alpha_c} \right] b, \quad (6)$$

where α_s and $\alpha_c = \tan^{-1} \left(\frac{b}{z_f} \right)$ are the angular sampling rate of the utilized light field display and the DSLFL, respectively. Moreover, b is the distance between adjacent views in the DSLFL at the viewing plane formulated as,

$$b = \min \left(\frac{z_f z_{\min}}{f(z_f - z_{\min})}, \frac{z_f z_{\max}}{f(z_f - z_{\max})} \right), \quad (7)$$

where z_{\min} and z_{\max} corresponds to the minimum and maximum distance of the 3D scene from the viewer plane. After computing the aperture A , the CoC diameter $d_c(z)$ is determined by Equation 2.

The camera's field of view can be computed based on the display's screen width s_w and the viewing distance from the screen z_f as shown in Figure 3,

$$\beta = 2 \arctan \left(\frac{s_w}{2z_f} \right). \quad (8)$$

By using the computed camera aperture and field of view, the rendered views will have a DoF matching the light field display specifications.

View Synthesis Pipeline

The pipeline of our proposed method is illustrated in Figure 5. It is based on the single-view multi-layer DoF rendering with tiled splatting pipeline [15] - green stages in Figure 5. To render several views simultaneously, we modified the tiling and accumulation stages - brown stages in Figure 5. In this way, the computationally intensive parts of the pipeline are calculated only once for several views, which makes the overall rendering of the light field faster.

To synthesize several views simultaneously, first a central view is rendered. Then, the accumulation stage is repeated for each rendered view in the horizontal and vertical direction. For each rendered view, pixel locations existing in each tile are shifted according to the distance between the new view and the central view, which is denoted as warping. The shifting amount $[\Delta_u(s), \Delta_v(t)]^T$ for pixels location (u,v) and view position index (s,t) is equal to the disparity between a rendered and the central view, which is computed through recentering with respect to the focus distance z_f and pixel's depth at the central view $z_{u,v}(0,0)$ as,

$$\Delta_{u,v}(s,t) = \begin{bmatrix} \Delta_u(s) \\ \Delta_v(t) \end{bmatrix} = f b_{s,t} \left(\frac{1}{z_{u,v}(0,0)} - \frac{1}{z_f} \right), \quad (9)$$

where $b_{s,t}$ is the distance between the rendered view and the central view that can be calculated as,

$$b_{s,t} = b' \begin{bmatrix} \left(s - \frac{N_x - 1}{2} \right) \\ \left(t - \frac{N_y - 1}{2} \right) \end{bmatrix}, \quad (10)$$

for $0 \leq s < N_x, 0 \leq t < N_y$, where (N_x, N_y) correspond to the total number of rendered views in the horizontal and vertical directions, and b' is the distance between two adjacent views.

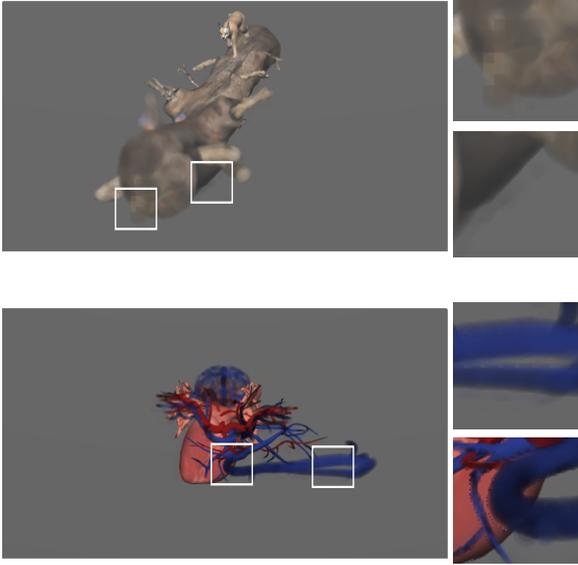


Figure 6. The pixelation and color bleeding artifacts due to missing pixels in the warping stage.

Due to the shift in the pixels' location during the warping stage, in comparison to original pipeline, each tile must store additional pixels for rendering adjacent views. Otherwise, artifacts as shown in Figure 6 will appear. To select the additional required pixels, in the tiling stage, the pixels' overlap with the neighbouring tiles is checked with an extended CoC radius $\lambda \frac{d_c(z)}{2}$. The extension factor λ is calculated as,

$$\lambda = \frac{\Delta_{u,v}^{max} + \frac{d_c^{max}}{2}}{\frac{d_c^{max}}{2}}, \quad (11)$$

where $\Delta_{u,v}^{max}$ is the l_2 -norm of the shift for the maximum desired baseline $b_{s,t}^{max}$ and d_c^{max} is the maximum CoC diameter in the scene.

To speed up light field rendering, adjacent view synthesis is performed with three different patterns, as illustrated in Figure 7 (a). The integration of each synthesis pattern into a light field is shown in Figure 7 (b), (c), and (d). The row pattern synthesizes three views at a time in horizontal or vertical direction and can be easily integrated into a light field. The star pattern generates five views, both horizontally and vertically, but the integration into a light field is challenging and some views, as visualized red in Figure 7 (d), cannot be rendered with the star pattern. Hence, they need to be rendered separately. Based on Equation 11, the extension factor λ is equal 2 for the row and star synthesis pattern. The grid pattern generates nine views in both horizontal and vertical directions and is easily integratable into a light field. However, corner views are outside of the aperture computed in Equation 6. Therefore, the extension factor is increased to $\lambda = 2\sqrt{2}$.

Evaluations

To evaluate the proposed method, three different scenes, vessels, trunk, and elephant, are used [6]. Examples of the rendered scenes are shown in Figure 8. The evaluation is done by considering rendering time and quality.

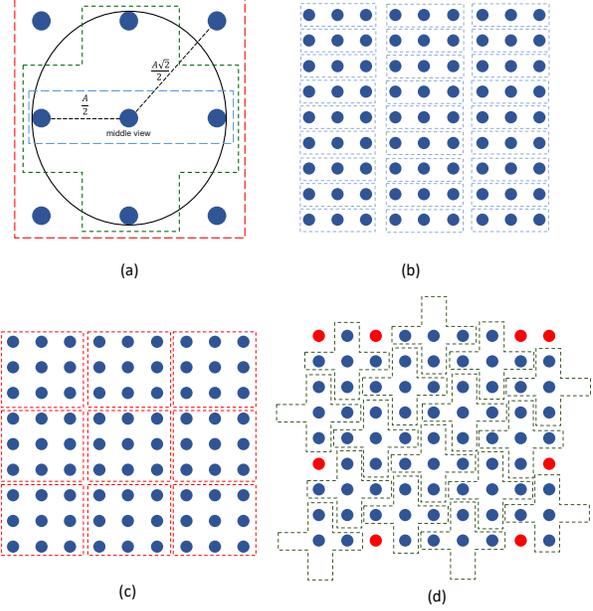


Figure 7. (a) Different view synthesis patterns: row (blue), star (green), and grid (red). In the grid pattern, views in the corner are outside of the aperture area. The integration of row (b), grid (c), and star (d) patterns into a light field. To render a light field using the star synthesis pattern, views marked as red need to be rendered individually.

To analyze the efficiency of the proposed method based on the rendering time, we first estimate the required time to render a single view, as well as several adjacent views using the row, star, and grid pattern. The resolution of each view is 1280x720, and the utilized GPU to render the light field is NVIDIA Quadro T1000. The extension factor λ is set to 1 for the single view, 2 for the row and star pattern, and $2\sqrt{2}$ for the grid pattern. The rendering times are given in Table 1. As seen in the table, using the grid pattern one can render nine views around two times faster than using the single view rendering method. Second, assuming several light fields with different sizes, starting from 3x3 to 21x21, we compute the required time to render each light field using the single view and the proposed method. As illustrated in Figure 9, all three proposed synthesis patterns accelerate the light field rendering. From all three, the largest reduction in rendering time is obtained by using the grid pattern. Using the star pattern increases the rendering time in comparison to the grid pattern since in the integration of the star pattern in the light field, some views need to be rendered individually. However, the star pattern occupies less memory due to using a smaller λ . The row pattern is the slowest but it is most suitable for generating horizontal parallax only [3] light fields. Times given in Table 1 and Figure 9 correspond to the required time for rendering views in the light field and do not consider the amount of time needed for post-processing procedures, e.g. recentering.

To analyze the rendering quality, we render a 6x6 light field with the aperture filtering method as ground truth [6], the multi-layer DoF rendering with tiled splatting for rendering single views [15], and the proposed approach using row, star, and grid pattern. The comparison is done with full-reference quality assessment metrics, that is, SSIM, PSNR, and MAE. The comparison

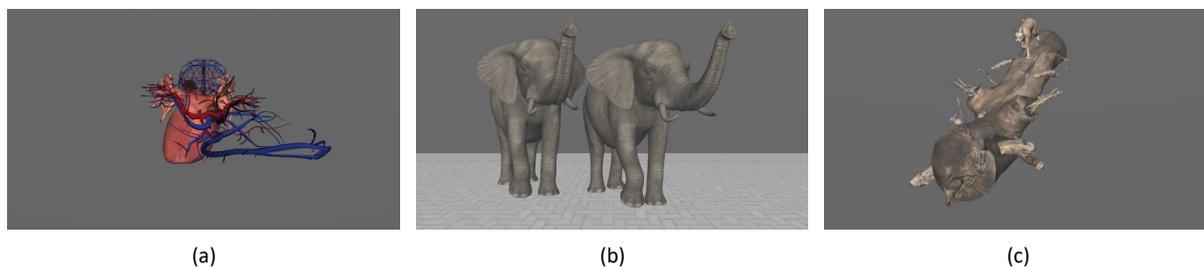


Figure 8. Vessels (a), elephants (b), and trunk (c) scenes rendered using pinhole camera.

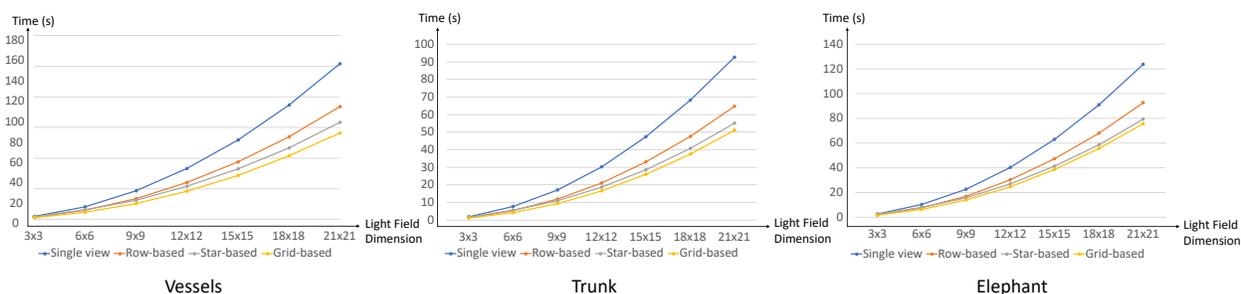


Figure 9. Rendering time comparison for proposed synthesis patterns across light fields of different sizes.

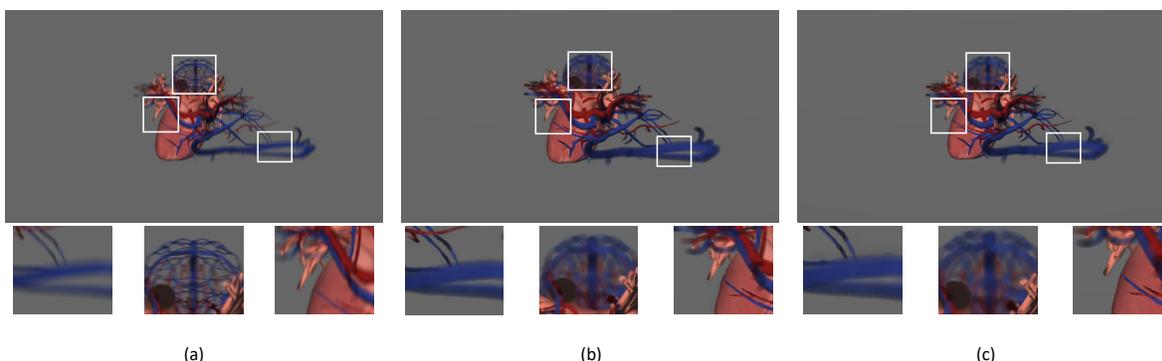


Figure 10. Vessels rendered using (a) aperture filtering, (b) single view DoF rendering, and (c) proposed method with grid pattern.

is done image by image and then averaged over the whole light field. As shown in Table 2, using our proposed method with any of the synthesis patterns produce a light field of a similar quality as the one that renders each view separately. This demonstrates that our method accelerates the light field rendering without compromising the quality. Figure 10 compares the rendering result of aperture filtering, single-view DoF rendering, and view rendered with the grid pattern. Figure 11 illustrates views from trunk and elephants scene rendered with the grid pattern.

Conclusion

In this paper we propose a method that simultaneously renders several adjacent views of a light field with each of them having the desired DoF. The proposed method significantly accelerates the light field rendering, resulting in a reduced rendering time. The achieved quality of the rendered images is similar to the one achieved by rendering every view separately. Although our proposed method accelerates the light field rendering, the method is still not real-time. Further improvements are needed, particularly in the case of large light fields.

Patterns	λ	Vessels	Trunk	Elephants
Single (1 view)	1	0.23	0.21	0.28
Row (3 views)	2	0.5	0.44	0.63
Star (5 views)		0.67	0.58	0.84
Grid (9 views)	$2\sqrt{2}$	1.15	1.04	1.54

Table 1. Rendering times in seconds for different scenes and synthesis patterns.

References

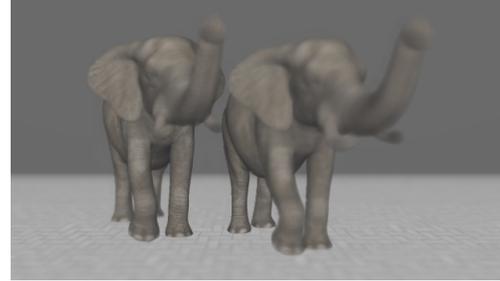
- [1] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," in Computational Models of Visual Processing, M. Landy and J. A. Movshon, Eds. 1991.
- [2] M. Zwicker, W. Matusik, F. Durand, and H. Pfister, "Antialiasing for Automultiscopic 3D Displays," in Symposium on Rendering, T. Akenine-Moeller and W. Heidrich, Eds., 2006, pp. 73-82.
- [3] R. Bregović, P. T. Kovács, and A. Gotchev, "Optimization of light field display-camera configuration based on display properties in spectral domain," Opt. Express, vol. 24, no. 4,

		Vessels		Elephants		Trunk	
		proposed	SV	proposed	SV	proposed	SV
SSIM	Row	0.968	0.968	0.928	0.928	0.976	0.976
	Star	0.967		0.928		0.976	
	Grid	0.968		0.928		0.976	
PSNR	Row	31.20	31.20	31.54	31.82	33.79	33.82
	Star	31.00		31.58		33.78	
	Grid	31.18		31.58		32.76	
MAE	Row	1.65	1.65	3.29	3.19	1.95	1.94
	Star	1.69		3.28		1.96	
	Grid	1.65		3.28		1.95	

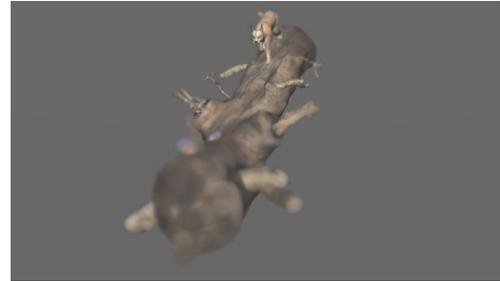
Table 2: The performance comparison of DoF rendering for our proposed method vs the single view (SV) [15] method, by using a full-reference evaluation. The size of the rendered light field is 6x6 and the reference views are rendered with the aperture filtering method introduced in [6].

pp. 3067-3088, 2016.

- [4] P. T. Kovács, R. Bregović, A. Boev, A. Barsi and A. Gotchev, "Quantifying Spatial and Angular Resolution of Light-Field 3-D Displays," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1213-1222, Oct. 2017.
- [5] K. Akbar and R. Bregovic, "Single-Image Evaluation of Angular and Spatial Resolution for Projection-Based Light Field Displays," 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP), Poitiers, France, 2023, pp. 1-6.
- [6] K. Akbar and R. Bregovic, "Antialiasing Filtering for Projection-Based Light Field Displays," 2023 International Symposium on Image and Signal Processing and Analysis (ISPA), Rome, Italy, 2023, pp. 1-6.
- [7] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000)*, Kurt Akeley, Ed., July 23-28, 2000, New Orleans, Louisiana, USA, pp. 297-306, Computer Graphics Annual Conference Series. New York, NY: Association for Computing Machinery.
- [8] R. L. Cook, T. Porter, and L. Carpenter, "Distributed ray tracing," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 137-145, Jul. 1984.
- [9] C. Soler, K. Subr, F. Durand, N. Holzschuch, and F. Sillion, "Fourier depth of field," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 12 pages, Apr. 2009.
- [10] X. Yu, R. Wang, and J. Yu, "Real-time Depth of Field Rendering via Dynamic Light Field Generation and Filtering," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 2099-2107, Nov.-Dec. 2010.
- [11] P. Haerberli and K. Akeley, "The accumulation buffer: hardware support for high-quality rendering," in *Proceedings of the 17th Annual Conference on Computer Graphics and In-*



(a)



(b)

Figure 11. Trunk and elephants scenes rendered with the proposed method.

teractive Techniques, 1990.

- [12] J. D. Mulder and R. van Liere, "Fast perception-based depth of field rendering," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '00)*, New York, NY, USA, 2000, pp. 129-133.
- [13] M. Potmesil and I. Chakravarty, "Synthetic Image Generation with a Lens and Aperture Camera Model," *IEEE Trans. Vis. Comput. Graph.*, vol. 1, no. 2, pp. 85-108, Apr. 1982.
- [14] K. Selgrad, C. Reintges, D. Penk, P. Wagner, and M. Stamminger, "Real-time depth of field using multi-layer filtering," in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games (i3D '15)*, New York, NY, USA: Association for Computing Machinery, 2015, pp. 121-127.
- [15] L. Franke, N. Hofmann, M. Stamminger, and K. Selgrad, "Multi-Layer Depth of Field Rendering with Tiled Splatting," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 17 pages, Jul. 2018.
- [16] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.
- [17] T. Balogh, P. T. Kovacs, Z. Dobranyi, A. Barsi, Z. Megyesi, Z. Gaal, and G. Balogh, "The Holovizio System - New Opportunity Offered by 3D Displays," in *Proceedings of TMCE*, 2008, p. 11.
- [18] Github (2013) [jimmiebergmann Sponza]. <https://github.com/jimmiebergmann/Sponza?tab=readme-ov-file>. [Accessed 2024].