

# CSLF: Cube Surface Light Field and Its Sampling, Compression, Real-Time Rendering

Xiaofei Ai , Yigang Wang , Simin Kou

Hangzhou Dianzi University, School of Computer Science, Hangzhou, China

## Abstract

Light field is gaining both research and commercial interests since it has the potential to produce view-dependent and photo-realistic effects for virtual and augmented reality. In this paper, we further explore the light field and presents a novel parameterization that permits 1) effectively sampling the light field of an object with unknown geometry, 2) efficiently compressing and 3) real-time rendering from arbitrary viewpoints. A novel, key element in our parameterization is that we use the intersections of the light rays and a general cube surface to parameterize the four-dimensional light field, constructing the cube surface light field (CSLF). We resolve the huge data amount problem in CSLF by uniformly decimating the viewpoint space to form a set of key views which are then converted into a pseudo video sequence and compressed using the high efficiency video coding encoder. To render the CSLF, we employ a ray casting approach and draw a polygonal mesh, enabling real-time generating arbitrary views from the outside of the cube surface. We build the CSLF datasets and extensively evaluate our parameterization from the sampling, compression and rendering. Results show that the cube surface parameterization can simultaneously achieve the above three characteristics, indicating the potentiality in practical virtual and augmented reality.

## CCS Concepts

• **Computing methodologies** → Image-based rendering; Ray tracing; Image compression;

## 1. Introduction

Light field has emerged as a solution for capturing scenes with photo-realism while allowing realistic changes in viewpoints. Current light field is known as a representation for all possible light rays with various intensities in all directions. This brings a problem, i.e., how to mathematically parameterize those rays so as to efficiently sample, compress and render the light field.

Fortunately, a number of light field parameterization methods have been proposed, such as two-plane parameterization (2PP) [LH96, GGSC96], two-sphere parameterization (2SP) [CLF98], sphere-plane parameterization (SPP) [IPL97] and surface parameterization [WAA\*00, CWZ\*18]. The considering specific visual task usually motivates the proposal of different light field parameterizations. However, these parameterization methods can not simultaneously satisfy 1) effectively sampling the light field of an object with unknown geometry, 2) efficiently compressing and 3) real-time rendering from arbitrary viewpoints, which are crucial for view-dependent rendering in virtual and augmented reality applications.

We propose a novel parameterization that supports the above three characteristics simultaneously. The key insight of our method is the cube surface light field (CSLF) that parameterizes the four-dimensional (4D) light field by the intersections of the rays and a general cube surface. Different with the 2PP method, CSLF al-

lows the parameterization for light rays that are parallel to the two planes. In contrast to surface parameterization, CSLF does not require the estimation of scene's geometry. Using a general cube surface instead of the spherical surface in 2SP and SPP, it benefits the sampling for both the viewpoint space and image space, as well as the compression and rendering implementation, making it practical in virtual and augmented reality applications. In summary, the main contributions of our work include the following:

- A novel parameterization method that permits to efficiently sample, compress and render the light field.
- A Monte Carlo path tracing based sampling method that generates multi-resolution CSLF datasets.
- A tailored compression approach that reduces the data amount of CSLF by 1000:1.
- Real-time CSLF rendering without exact scene's geometry, which supports generating the view-dependent and photo-realistic effects from arbitrary viewpoints outside the cube surface.

## 2. Related Work

There are various light field parameterization methods for different visual tasks and detailed surveys can be seen in [WMJ\*17]. Here, we focus on those most related work, including light field parameterization, compression and its subsequently rendering.

## 2.1. Light Field Parameterization

Light field is a high-dimensional representation of the rays in free space. To incorporate such a light field into a computational framework, Levoy and Hanrahan [LH96] proposed the 4D light field by assuming that light rays do not attenuate during the propagation in free space. In this case, a light ray can be parameterized by the corresponding intersections with two parallel planes, which is convenient for acquiring a light field with an array of cameras [WSLH01, WJV\*05] or micro-lenses [NLB\*05]. A special case of 2PP is the Lumigraph [GGSC96] that enables a representation for object or scene using six 2PPs. These methods cannot parameterize those rays parallel to the two planes [CTCS00, BFV12], resulting a limited viewing volume.

Similar to the Lumigraph, spherical parameterizations [CLF98, IPL97, DDB\*15] have been proposed to represent the light field of a scene with finite size, such that a unit sphere can encapsulate the whole scene. In this case, light rays are parameterized using the intersections with two spheres, i.e., positional sphere and directional sphere. Recently, Overbeck et al. [OEE\*18] presented a spherical light field system that allows capturing spherical light field of the outside environment. The captured light field provides data required to generate novel views located within the recorded spherical volume. Alternative spherical parameterization is SPP [IPL97] that defines the light rays using the intersections with a sphere and a plane. The spherical light field is hard for uniform discretization since the sphere is more irregular than flat surface. The relation between each spherical light field sub-view includes both translation and rotation that indicates less coherence, making its compression more difficult.

In addition, Wood et al. [WAA\*00] proposed the surface light field (SLF) that defines the light rays on a base mesh and the lumispheres. Different with the above parameterizations, it requires the estimation of base mesh and the accuracy of the approximate scene's geometry has a deep impact on the quality of reconstructed views.

## 2.2. Compression

Raw light field data using the above parameterizations can be very huge, e.g., hundreds of gigabytes for a static scene, making it difficult for storing and transmitting over the limited bandwidth. Numerical methods are commonly employed to compress the huge amount of data including vector quantization (VQ) [LH96, WAA\*00], wavelet transforms [CZRG06], non-negative matrix factorization [CBCG02], principal component analysis [WAA\*00, CBCG02], and other compression methods utilizing scene's geometry [OEE\*18, WAA\*00, CBCG02]. References [LH96, GGSC96] designed a compression scheme that combines VQ of two-dimensional (2D) slices or 4D tiles with gzip entropy encoding for a total compression ratio of 120:1. This schemes allow random access and fast decompression, so that real-time rendering becomes feasible. Recently, image compression techniques, e.g., JPEG-Pleno [SAT\*18, EFPS16], have become computationally-affordable and light field compression has also been inspired by video coding methods [OEE\*18, BFO\*20, LZM18].

## 2.3. Rendering

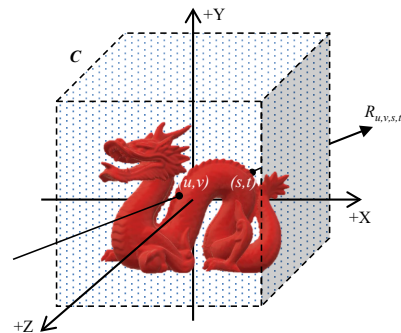
Light field rendering without geometry or depth information can be implemented by image warping and ray-space interpolation [LH96], in which each ray corresponding to a target screen pixel is mapped to nearby sampled rays. It has been shown that the use of a quadratic linear kernel is beneficial in terms of computational efficiency and quality due to the lack of band-limited property of light field [LSS04].

For more sophisticated light fields combining geometry information such as SLF, computer graphic methods have been exploited to accelerate the rendering. The geometry information can either be implicit that relies on positional correspondences or explicit in form of depth along known light ray or three-dimensional (3D) coordinates. Using the approximate geometry, the Lumigraph used a depth corrected rendering method. Buehler et al. [BBM\*01] created a proxy to warp multiple images into a novel view and blended them with specific weights. More recently, Overbeck et al. [OEE\*18] projected the light field images onto a view-dependent geometry and blended the results using a disk-based reconstruction basis.

## 3. Methodology

We aim to represent the light field of an object or scene, i.e., outside-in looking light field with unknown exact geometry. To this end, we propose the CSLF that uses a cube to encapsulate the whole scene and parameterizes the light rays by the intersections on the cube surface. The detailed parameterization, sampling, compression and rendering of CSLF are described throughout this section.

### 3.1. Parameterization



**Figure 1:** Cube surface parameterization in which light ray's origin is defined by the first intersection and the direction is defined by the second intersection.

We begin with a cube which centroid is set to the origin of 3D coordinate system. This cube surface can be defined by a 2D function on each face and can be written as the equation below:

$$C : \sum_{i=1}^6 \Pi_i(x, y) \subset \mathbb{R}^2 \quad (1)$$

where  $(x, y)$  refers to arbitrary point's coordinates on the  $i$  cube face that is parallel to one of the coordinates, such as  $\pm X$ ,  $\pm Y$  or  $\pm Z$  plane. Using this cube surface, light ray's origin is defined by the first intersection  $(x, y) = (u, v)$  on the cube surface and the direction is defined by the second intersection  $(x, y) = (s, t)$ , as shown in Fig. 1. The CSLF is thus defined by the  $RGB$  value of light rays in all directions, written as the following equation:

$$L : \sum_{m=1}^6 \left( \Pi_m(u, v) \times \sum_{n=1}^6 \Pi_n(s, t) \right) \longrightarrow RGB \quad (2)$$

where  $m$  and  $n$  refer to the index of the cube face and  $m \neq n$  that is because the ray cannot intersect with the object when its origin and direction are co-planar, i.e., on a common cube face. At present, each light ray is defined by a six tuples  $(m, u, v, n, s, t)$  in that  $m$  and  $n$  are both enumerations from 1 to 6. We further simplify the tuples by using the cube texture coordinates  $(u, v)$  and  $(s, t)$ , resulting the final representation of CSLF:

$$L : I(u, v) \times I(s, t) \longrightarrow RGB \quad (3)$$

The CSLF can be viewed as  $u, v$  array of images  $I(s, t)$  or as  $s, t$  array of images  $I(u, v)$ . Each image in the array represents the rays starting from a point on one cube face to other five cube faces, which can be formatted by a normal cube map.

### 3.2. Sampling

Since CSLF is continuous, it requires discretization in each of the  $u, v, s, t$  dimensions to map such a function (Eq. 3) into a computational framework and efficiently sample the CSLF. We denote the  $u, v$  dimensions as viewpoint space and the  $s, t$  dimensions as image space. The discretization can be different between these two spaces or even all of the four dimensions. Without loss of generality, the cube can be defined by two vertices:  $P_{max} = (x_{max}, y_{max}, z_{max})$  and  $P_{min} = (x_{min}, y_{min}, z_{min})$ . Therefore, the length for a given cube  $P_{min}, P_{max}$  is  $E = x_{max} - x_{min}$ . To uniformly sample the CSLF, we choose to discretize the  $u, v$  dimensions into  $6 \times M^2$  samples and discretize the  $s, t$  dimensions into  $6 \times N^2$  samples. Each sample  $(u, v)$  is indexed with  $(i, j)$  and is located at  $(u_i, v_j)$  that can be computed by subdividing  $E$  into  $M$  and  $N$  parts respectively. Similarly, a sample  $(s, t)$  is indexed with  $(p, q)$  and is located at  $(s_p, t_q)$ . A 2D sampling point pair is thus indexed with  $(i, j, p, q)$  and its value, i.e., the  $RGB$  at  $(i, j, p, q)$  is referred to as  $c_{i,j,p,q}$ . Therefore, the discrete CSLF is as the following linear sum:

$$L(u, v, s, t) = \sum_{i=0}^M \sum_{j=0}^M \sum_{p=0}^N \sum_{q=0}^N c_{i,j,p,q}(u, v, s, t) \quad (4)$$

One of advantages using the discretization is the convenience for sampling the light field of an object. We sample the CSLF directly for synthetic scenes based on the Monte Carlo path tracing [DBB18]. Firstly, by modifying the path tracing algorithm into a multi-viewpoint tracing, the ray's origin and direction uniformly distribute on all the sampling points on a virtual cube surface which

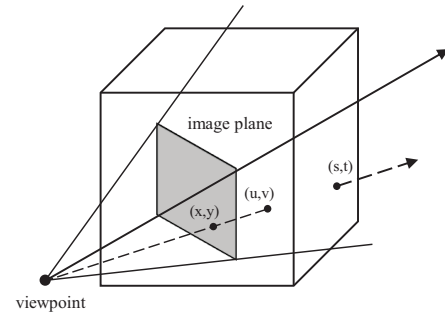
is defined by adding an offset to its bounding box. This ensures that the object can be wrapped by the cube and does not intersect with cube surface. We finally perform the path tracing at each sampling point to generate the CSLF. For each sampling point in viewpoint space, the light field is represented by a cube map that is stored as PNG format.

### 3.3. Compression

For an object with unknown geometry and material, the CSLF can be very large and requires a large amount of storage. For example, it requires  $6 \times 64 \times 64 \times 512 \times 512 \times 6 \times 24 = 108$  GB of storage to store the entire CSLF discretized by  $M = 64, N = 512$  using 8 bits per color channel (24 bits per pixel). Fortunately, there is a large of redundancies in each light field image as well as coherence between each light ray sample.

In our cube surface parameterization, cube maps are used to represent the light field sub-views. However, it is unnecessary to store all the views when  $(u, v)$  and  $(s, t)$  are co-planar. Therefore, each cube map degenerates into five small images for each of the cube faces. Secondly, all of light field sub-views on a common cube face show highly similar, indicating that we can apply a transform codec to the CSLF, such as a wavelet transform. Here, we uniformly decimate the viewpoint space to form a set of key views. This is aimed at decreasing the number of images which go to the compression engine. The key views are converted further into a pseudo video sequence and compressed using high-efficiency video coding (HEVC) encoder.

### 3.4. Rendering



**Figure 2:** The relation between CSLF and a pixel in arbitrary viewpoint.

Light field rendering is to synthesize novel view for a desired viewpoint, which can be achieved by solely looking up and blending reference light rays. One of advantages of our cube surface parameterization is the convenient rendering from arbitrary viewpoints. Figure 2 shows the relation between the cube surface parameterization and a pixel in an arbitrary view. For a desired pixel, what we need to do is to calculate the two intersections  $(u, v)$  and  $(s, t)$  of the ray started from the viewpoint to the pixel and the cube surface. This can be implemented by various ray tracing algorithms or path tracing algorithms [HLRSR09]. Meanwhile, it is convenient for computing the intersections of ray and triangle mesh, and the

rendering can be in real time on most of consumer computer since the simplicity of cube surface. Algorithm 1 illustrates the rendering of a novel view from an arbitrary viewpoint.

---

**Algorithm 1** CSLF Rendering using Ray Casting on Cube Surface
 

---

**Input:** CSLF  $L$ 
**Output:** Frame buffer  $I$ 

```

1: Initialize the frame buffer to black
2: Polygonal cube setup  $C$ 
3: Viewing transformation
4: for all pixels  $p \in I$  do
5:   Generate ray  $r_i$  from the viewpoint  $v$  to a specific pixel  $p_i$ 
6:   Intersection test with  $r_i$  and  $C$ 
7:   if  $r_i \cap C$  then
8:     Convert the first intersection  $v_i$  to  $(u, v)$ 
9:     Convert the second intersection  $v_o$  to  $(s, t)$ 
10:     $p_i = L(u, v, s, t)$ 
11:   else
12:     Set the  $p_i$  to black
13:   end if
14: end for
15: return Frame buffer  $I$ 

```

---

## 4. Results and Evaluation

### 4.1. Datasets

Since we use a novel parameterization, the current light field datasets [Dan18, LYJ\*17] cannot be used directly to evaluate our method. According to Sec. 3.2, we sample the CSLF for two synthetic scenes: one is the Stanford dragon with specular reflectance and the other is the Cornell box with global illumination. All presented results are generated using a desktop PC with an Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz, 16GB RAM and Nvidia Quadro K1200 GPU.

**Table 1:** The statistics of sampling the CSLF by different viewpoint resolutions

Viewpoint Resoluition	M = 16	M = 32	M = 40
Total Images	1536	6144	9600
Sampling Time (s)	9216	36864	57600

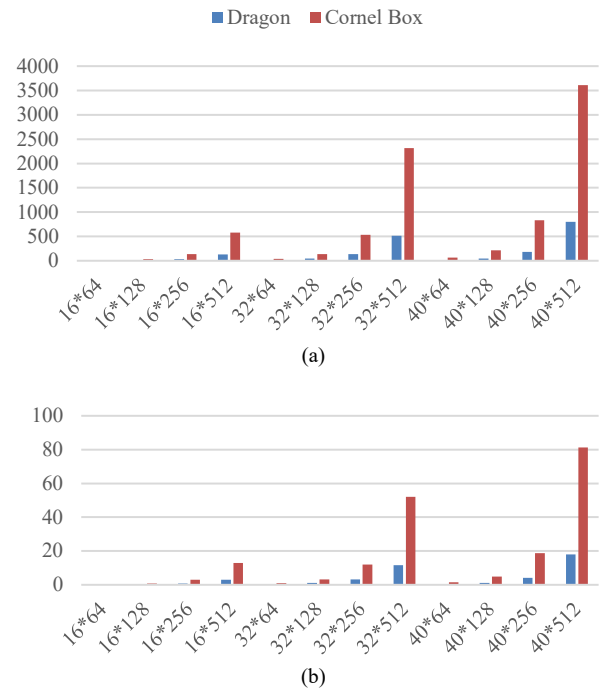
### 4.2. Sampling

To evaluate our method, the CSLFs are sampled with different resolutions both in viewpoint space and image space. Since  $6 \times M \times M \times N \times N \times 6 \times 24$  bits must be much less than the maximum memory and  $512 \times 512$  pixels are usually enough for a single view, we set it to both the upper limit of view resolution and image resolution. When the image resolution reaches at  $512 \times 512$ , the viewpoint resolution is limited by  $60 \times 60$  to make the total storage available and without considering the memory occupied by other programs. Therefore, we sample the CSLF using the viewpoint resolutions  $M = 16, 32, 40$  and image resolutions  $N = 64, 128, 256, 512$ . Since the path tracing is implemented in

GPU and performed progressively, the image resolution and scene's complexity have little impact on the time cost. We report the total number of sampled images and the time cost for different viewpoint resolutions in Table 1.

### 4.3. Compression

The sampled datasets are as input of the compression. Figure 3 shows the data amount for the uncompressed and compressed CSLF datasets sampled by different viewpoint resolutions and image resolutions. The compression ratio can reach to 1000:1 using the HEVC encoder while preserving the visual quality, which is because massive redundancy exists in the CSLF and the general cube surface parameterization makes the it convenient to compress using existing advanced compression methods.

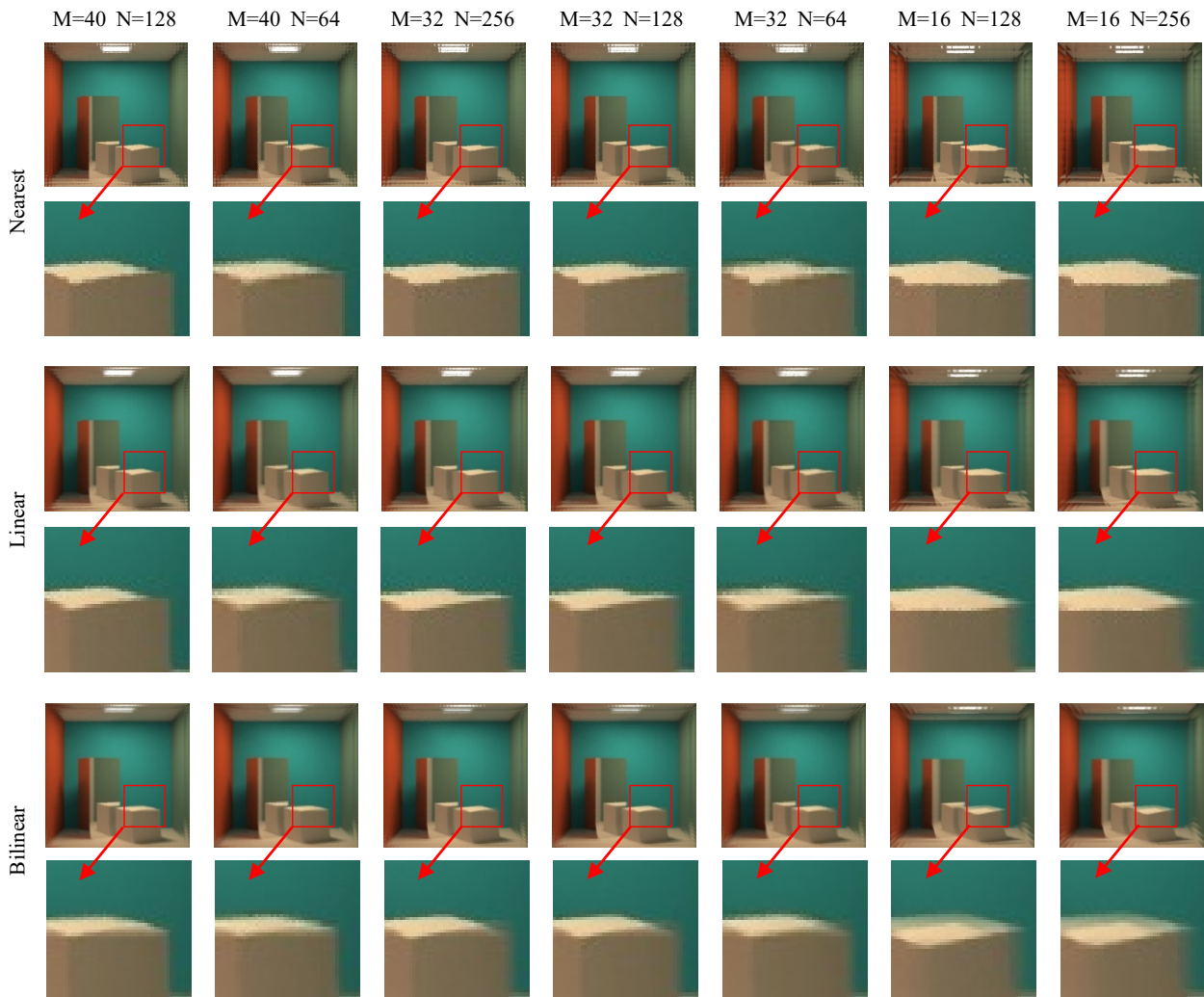


**Figure 3:** The memory cost (MB) for uncompressed CSLF (a) and compressed CSLF (b).

### 4.4. Rendering Quality

To evaluate the render quality, we generate several views with a resolution of  $512 \times 512$  from the Cornell box datasets with seven different resolutions, including  $M \times N = 40 \times 128, 40 \times 64, 32 \times 256, 32 \times 128, 32 \times 64, 16 \times 256, 16 \times 128$ . Figure 4 illustrates the results generated by using three interpolation methods in our rendering algorithm, including the nearest method, linear interpolation and bilinear interpolation.

We find that the visual quality varies from different resolutions. Good results without obvious ghosting artifacts can be generated by using the resolutions of  $40 \times 40 \times 128 \times 128$  and  $40 \times 40 \times 64 \times 64$ . Given a specific viewpoint resolution, the resolution in image space



**Figure 4:** Visualization for novel views rendered from Cornell box datasets with the different resolutions.

has little impact on the final results. It indicates that a lower image resolution is enough for CSLF rendering. Given a specific image resolution, higher visual quality can be achieved by a larger resolution in viewpoint space, indicating larger viewpoint resolution can be used for improving the quality of CSLF rendering. Moreover, different interpolation methods also have an impact on the rendering results. As can be seen from each column of Fig. 4, the highest quality is generated by bilinear interpolation, indicating that a good interpolation requires less resolution for providing comparative visual quality in the CSLF.

#### 4.5. Limitations and Future Work

Although our parameterization supports the novel views rendering without exact geometry, it is a pure image-based method thus requiring a dense sampling for antialiasing rendering. Despite the compression has reached a certain level, such as 40:1-1000:1, it still requires much memory to generate high-quality novel views.

In the future, we would like to further exploit the redundancy in CSLF and develop an algorithm for optimally compressing, perhaps using the deep learning techniques similar to those used in image compression.

For the CSLF in real scenes, our method may require a large number of images captured by hand-held cameras or special camera rigs. Moreover, the 4D CSLF does not support dynamic scenes inherently. A potential solution is to add the time dimension to our CSLF.

In addition, CSLF rendering is real-time in GPU, independent with the complexity of scenes or light conditions, indicating that our method can be used for the acceleration for time-consuming global illumination methods. However, we only evaluate seven configurations of subdivision and three different methods for ray's interpolation by which the visual quality is limited. Based on the cube surface parameterization, we would like to further develop



algorithms for the high-quality rendering with moderate light field resolution.

## 5. Conclusions

This paper has proposed a novel parameterization that represents the light field of an object or scene by a general cube surface. Based on the cube surface parameterization, a Monte Carlo path tracing based sampling method has been proposed to generate multi-resolution CSLF datasets. A tailored compression method has also been presented and the data amount of CSLF can be reduced to 1/1000. Moreover, we have presented the subsequently real-time CSLF rendering method that supports generating the view-dependent and photo-realistic effects from arbitrary viewpoints outside the cube surface. Our method has been evaluated through various experiments and comparisons. Results have demonstrated that our cube surface parameterization can simultaneously achieve the three characteristics 1) effectively sampling the light field of an object with unknown geometry, 2) efficiently compressing and 3) real-time rendering from arbitrary viewpoints.

## References

- [BBM\*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. In *Computer Graphics, SIGGRAPH 2001 Proceedings* (2001), pp. 425–432. doi:10.1145/383259.383309. 2
- [BFO\*20] BROXTON M., FLYNN J., OVERBECK R., ERICKSON D., HEDMAN P., DUVAL M., DOUGARIAN J., BUSCH J., WHALEN M., DEBEVEC P.: Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 1–15. doi:10.1145/3386569.3392485. 2
- [BFV12] BAGNATO L., FROSSARD P., VANDERGHEYNST P.: Plenoptic spherical sampling. In *2012 19th IEEE International Conference on Image Processing* (2012), IEEE, pp. 357–360. doi:10.1109/ICIP.2012.6466869. 2
- [CBCG02] CHEN W.-C., BOUGUET J.-Y., CHU M. H., GRZESZCZUK R.: Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 447–456. doi:10.1145/566654.566601. 2
- [CLF98] CAMAHORT E., LERIOS A., FUSSELL D.: Uniformly sampled light fields. In *Eurographics Workshop on Rendering Techniques* (1998), Springer, pp. 117–130. doi:10.1007/978-3-7091-6453-2\_11. 1, 2
- [CTCS00] CHAI J.-X., TONG X., CHAN S.-C., SHUM H.-Y.: Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 307–318. doi:10.1145/344779.344932. 2
- [CWZ\*18] CHEN A., WU M., ZHANG Y., LI N., LU J., GAO S., YU J.: Deep surface light fields. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 1 (2018), 1–17. doi:10.1145/3203192. 1
- [CZRG06] CHANG C.-L., ZHU X., RAMANATHAN P., GIROD B.: Light field compression using disparity-compensated lifting and shape adaptation. *IEEE transactions on image processing* 15, 4 (2006), 793–806. doi:10.1109/TIP.2005.863954. 2
- [Dan18] DANSEREAU D. G.: Stanford light field archives, 2018. <http://lightfields.stanford.edu>. Accessed July 15, 2021. 4
- [DDB18] DUTRÉ P., BEKAERT P., BALA K.: *Advanced Global Illumination*, 2nd edition ed. A. K. Peters Ltd./CRC Press, October 2018. doi:10.1201/9781315365473. 3
- [DDB\*15] DEBEVEC P., DOWNING G., BOLAS M., PENG H.-Y., URBACH J.: Spherical light field environment capture for virtual reality using a motorized pan/tilt head and offset camera. In *ACM SIGGRAPH 2015 Posters*. 2015, pp. 1–1. doi:10.1145/2787626.2787648. 2
- [EFPS16] EBRAHIMI T., FOESSEL S., PEREIRA F., SCHELKENS P.: JPEG Pleno: Toward an efficient representation of visual reality. *IEEE MultiMedia* 23, 4 (2016), 14–20. doi:10.1109/MMUL.2016.64. 2
- [GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 43–54. doi:10.1145/237170.237200. 1, 2
- [HLRS09] HADWIGER M., LJUNG P., REZK SALAMA C., ROPINSKI T.: Advanced illumination techniques for GPU volume raycasting. *ACM SIGGRAPH Courses Program* (August 2009), 1–166. doi:10.1145/1508044.1508045. 3
- [IPL97] IHM I., PARK S., LEE R. K.: Rendering of spherical light fields. In *Proceedings The Fifth Pacific Conference on Computer Graphics and Applications* (1997), IEEE, pp. 59–68. doi:10.1109/PCCGA.1997.626172. 1, 2
- [LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 31–42. doi:10.1145/237170.237199. 1, 2
- [LSS04] LIN Z., SHUM H.-Y., SHUM H.: A geometric analysis of light field rendering. *International Journal of Computer Vision* (July 2004), 121–138. doi:10.1023/B:VIST.0000015916.91741.27. 2
- [LYJ\*17] LI N., YE J., JI Y., LING H., YU J.: Saliency detection on light field. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1605–1616. doi:10.1109/TPAMI.2016.2610425. 4
- [LZM18] LIU Y., ZHU C., MAO M.: Light field image compression based on quality aware pseudo-temporal sequence. *Electronics Letters* 54 (02 2018). doi:10.1049/el.2017.4560. 2
- [NLB\*05] NG R., LEVOY M., BRÉDIF M., DUVAL G., HOROWITZ M., HANRAHAN P.: *Light field photography with a hand-held plenoptic camera*. PhD thesis, Stanford University, 2005. 2
- [OEE\*18] OVERBECK R. S., ERICKSON D., EVANGELAKOS D., PHARR M., DEBEVEC P.: A system for acquiring, processing, and rendering panoramic light field stills for virtual reality. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15. doi:10.1145/3272127.3275031. 2
- [SAT\*18] SCHELKENS P., ALPASLAN Z., TABUS I., EBRAHIMI T., OH K.-J., PINHEIRO A., CHEN Z., PEREIRA F.: JPEG Pleno: a standard framework for representing and signaling plenoptic modalities. In *Optical Engineering + Applications* (09 2018), vol. 10752 of *SPIE Conference Proceedings*. doi:10.1117/12.2323404. 2
- [WAA\*00] WOOD D. N., AZUMA D. I., ALDINGER K., CURLESS B., DUCHAMP T., SALESIN D. H., STUETZLE W.: Surface light fields for 3D photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 287–296. doi:10.1145/344779.344925. 1, 2
- [WJV\*05] WILBURN B., JOSHI N., VAISH V., TALVALA E.-V., ANTUNEZ E., BARTH A., ADAMS A., HOROWITZ M., LEVOY M.: High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers*. 2005, pp. 765–776. doi:10.1145/1073204.1073259. 2
- [WJM\*17] WU G., MASIA B., JARABO A., ZHANG Y., WANG L., DAI Q., CHAI T., LIU Y.: Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing* 11, 7 (2017), 926–954. doi:10.1109/JSTSP.2017.2747126. 1
- [WSLH01] WILBURN B. S., SMULSKI M., LEE H.-H. K., HOROWITZ M. A.: Light field video camera. In *Media Processors 2002* (2001), vol. 4674, International Society for Optics and Photonics, pp. 29–36. doi:10.1117/12.451074. 2