# Compression and Real-Time Rendering of Inward Looking Spherical Light Fields

Saghi Hajisharif[1†] and Ehsan Miandji[2] and Gabriel Baravadish [1] and Per Larsson [1] and Jonas Unger[1]

[1] Linköping University, Sweden [2] INRIA, Rennes, France

**Figure 1:** *Left: schematic of our* 360° *inward-looking light field capturing system that consists of a precision guided rotating arm with multiple cameras. Right: a few samples of the synthesized light field images from different data sets and their corresponding depth maps.*

**Abstract**

*Photorealistic rendering is an essential tool for immersive virtual reality. In this regard, the data structure of choice is typically light fields since they contain multidimensional information about the captured environment that can provide motion parallax and view-dependent information such as highlights. There are various ways to acquire light fields depending on the nature of the scene, limitations on the capturing setup, and the application at hand. Our focus in this paper is on full-parallax imaging of large-scale static objects for photorealistic real-time rendering. To this end, we introduce and simulate a new design for capturing inward-looking spherical light fields, and propose a system for efficient compression and real-time rendering of such data using consumer-level hardware suitable for virtual reality applications.*

**CCS Concepts**

• ***Computer graphics*** → *Image-based rendering; Computational photography; Image compression;*

## 1. Introduction

Producing high-quality content for Virtual Reality (VR) applications is a challenging task. The human visual system is highly capable of distinguishing real content from virtual ones. Visual discrepancies and delays in rendering create uncomfortable symptoms such as headaches, dizziness, and fatigue. In recent years, light field imaging has been very successful in capturing high dimensional data from real environments with complex geometry, lighting, shadows, and reflectance at low cost and high quality. In light field imaging, all rays from different directions and locations are recorded, creating a stereoscopic parallax leading to a realistic experience in various applications such as virtual museums, product visualization, visual effect industry, and computer games.

In this paper, we present a design, using simulation, for a capturing system to acquire a full inward-looking spherical light field of a scene providing 6-DoF viewing. We have employed a learning-based compression technique [MHU19, HML*19] for encoding light field data sets that, besides providing a real-time rendering of large amount of data, has shown to be very efficient in maintaining the visual quality. The light-weight reconstruction algorithm allows for high frame rates suitable for various VR applications.

## 2. Related Works

Light field capturing and rendering was first introduced as a system comprising of multiple cameras capturing a scene from various vantage points in order to render the aforementioned scene with or without geometry information [LH96, GGSC96]. Light fields are high-dimensional data that can be parametrized in various ways. For instance, considering a 2-plane parameterization, a light field is a 5-dimensional (5D) function $L(x,y,\theta,\phi,c)$, representing the spatial domain $(x,y)$, the angular domain $(\theta,\phi)$, and the spectral do-

† Corresponding author: email: firstname.lastname@liu.se



DOI: 10.2312/egs.20201007

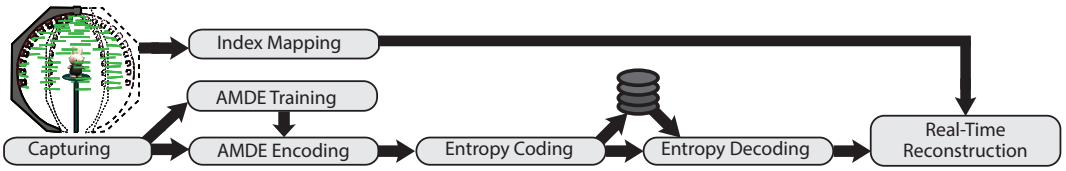


**Figure 2:** *Our proposed pipeline for acquisition, compression and real-time rendering of spherical light fields.*

main $c$. Given the large memory footprint of light fields, capturing and storage of such data pose various challenges.

**Capturing** There exists many approaches for capturing light fields, from camera arrays [WJV*] to gantry light fields [LH96, GGSC96, KZP*13]. For capturing high-quality light fields, a large number of cameras are required, which is very costly and bulky, with excessive bandwidth requirements. Hand-held hight field cameras [NLB*05, Ray19] were designed to capture dense light field with micro-baseline by trading spatial resolution for angular resolution. Furthermore, due to their small disparity between neighboring views, it is not suitable for VR applications. To capture 360° of the environment, Overbeck et al. [OEE*18] introduce a system for capturing outward-looking light fields of the environment. Recently, Mildenhall et al. [MSOC*19] proposed a user-guided capturing system for inward-looking light fields, but it has limitations on capturing light fields with high spatial resolution.

**Compression** There are various methods for compression of light fields like analytical basis functions such as discrete cosine transform used in JPEG or wavelets and Fourier bases. Unsupervised learning methods such as KSVD [AEB06], on the other hand, derive dictionary models directly from the data set. Miandji et al. [MHU19] have proposed a novel learning-based method suitable for high dimensional data that provides a high compression ratio compared to the previous work, as well as supporting fast GPU-based encoding [BMU19].

**Contributions** In this paper, we propose a capturing design that solves the problem of multi-camera systems for capturing a 360° inward-looking light field data set using only a few cameras. Furthermore, we employ the *aggregated multidimensional dictionary ensemble* (AMDE) algorithm similar to Miandji et al. [MHU19] for training a multidimensional dictionary to encode the acquired light field data. We further compress the data by quantizing the sparse coefficients obtained from AMDE using a clustering algorithm, followed by the entropy coding of the quantized values.

## 3. Capturing System

The capturing setup consists of multiple cameras mounted equidistantly on a circular arm, looking inward and rotating around an axis, as shown in Figure 1. The cameras can be placed in any desirable configuration to cover the full outgoing radiance of the scene, either densely with a smaller field of view or placed sparsely with a larger field of view. In our experimental simulations, thirteen cameras are used to cover the hemisphere centered at the object to reduce the baseline between the cameras. By rotating the arm around the object, in a controlled manner, the light field of the scene is captured. In this paper, we set the rotation angle to 1°, meaning that the angular resolution of the data set is $13 \times 360°$. This results in hundreds of Gigabytes of data, which will, later on, be compressed using a learning-based compression algorithm in order to utilize the highly insufficient GPU memory for real-time rendering.

### 3.1. Index Mapping

To find the nearest cameras in real-time reconstruction, an index map is created by sampling the azimuth $\theta$ and elevation $\phi$ directions from the calibration data. Extrinsic camera matrices are used to estimate the center of the capturing device by fitting a sphere to the camera positions. For each sample on the sphere, four nearest camera indices are stored in an index map for fast access to the data during the real-time rendering.

### 3.2. Sparse Approximation

Displaying the light field data set in real-time requires an efficient compression algorithm with random access to the data set such that only a small portion of the light field is reconstructed (i.e. decoded) at a given time. In this paper, we use an unsupervised learning algorithm to train an Aggregate Multidimensional Dictionary Ensemble (AMDE) [MHU19] for sparse representation of the light field data set. A single dictionary in AMDE consists of multiple orthonormal matrices (one for each data dimension). For efficient sparse representation, multiple dictionaries are trained in practice. Once an AMDE is constructed given a training set, it can be used to represent any light field data set with reasonably similar image statistics as the training set. Advantages of AMDE include the small memory footprint for the dictionaries, as well as producing highly sparse representations; together, these properties lead to a significant compression ratio in the orders of 100:1 or more depending on the structure of the input data, as shown in Section 5.

#### 3.2.1. Training

AMDEs are constructed by training a set of dictionaries on a training set that consist of $N_l$ small patches (or data points), denoted $\{\mathcal{L}^{(i)}\}_{i=1}^{N_l}$, extracted from a set of light fields. The testing set, i.e. the light field data set that we intend to compress, is denoted by $\{\mathcal{T}^{(i)}\}_{i=1}^{N_t}$ with $N_t$ data points. A data point in this context refers to as a small patch that spans all dimensions of the light field. Since the light field data set acquired with our proposed design is very sparse in the elevation angle, the elevation angle is not included in the data point due to the lack of coherence between neighboring angles. Hence, the light field is compressed for each azimuthal ring independently. With a slight abuse of notation, we denote the size of each data point as $x \times y \times \theta \times c$, where $(x \times y)$ is the resolution of the data point in the spatial domain, $\theta$ is the azimuthal resolution, and $c$ is the number of color channels. Moreover, we assume that a reasonably accurate depth information is available. We include the depth information in the last dimension; i.e. $c = 4$, where the dimensions are $R$, $G$, $B$, and $D$, where $D$ is the depth.

As the first step, a pre-clustering algorithm is used on the data points to group them based on their sparsity and reconstruction error into $C$ pre-clusters. This step is necessary to reduce the effect of noise in the data set and training time while improving the sparsity of the representation [HML*19]. For each pre-cluster, a Multidimensional Dictionary Ensemble (MDE) is trained, $\{\mathbf{U}^{(1,k)},\dots,\mathbf{U}^{(4,k)}\}_{k=1}^{K}$, where $K$ is the number of dictionaries. Each data point can then be represented as:

$$\mathcal{L}^{(i)} = \mathcal{S}^{(i)} \times_1 \mathbf{U}^{(1,k)} \cdots \times_4 \mathbf{U}^{(4,k)} = \mathcal{S}^{(i)} \bigtimes_{j=1}^{4} \mathbf{U}^{(j,k)}, \tag{1}$$

where $\mathcal{S}^{(i)}$ is a tensor of sparse coefficients and $\left\|\mathcal{S}^{(i,k)}\right\|_0 \leq \tau_l$, where $\tau_l$ is a user-defined sparsity parameter for training. To train each MDE satisfying (1), we solve the following optimization problem

$$\min_{\mathbf{U}^{(j,k)},\mathcal{S}^{(i,k)},\mathbf{M}_{i,k}} \sum_{i=1}^{N_l}\sum_{k=1}^{K} \mathbf{M}_{i,k} \left\| \mathcal{L}^{(i)} - \mathcal{S}^{(i,k)} \bigtimes_{j=1}^{4} \mathbf{U}^{(j,k)} \right\|_F^2, \tag{2}$$

where, $\mathbf{M} \in \mathbb{R}^{N_l \times K}$ is a clustering matrix associating each data point to a dictionary in the ensemble. The trained MDEs for all the $C$ pre-clusters are aggregated to form the AMDE:

$$\Psi = \bigcup_{i=1}^{C} \left\{ \mathbf{U}^{(1,k,i)},\dots,\mathbf{U}^{(4,k,i)} \right\}_{k=1}^{K} = \left\{ \mathbf{U}^{(1,k)},\dots,\mathbf{U}^{(4,k)} \right\}_{k=1}^{CK} \tag{3}$$

### 3.2.2. Encoding

To encode the light field data set, each data point $\{\mathcal{T}^{(i)}\}_{i=1}^{N_t}$, is projected onto all dictionaries of AMDE as follows

$$\mathcal{S}^{(i,k)} = \mathcal{T}^{(i)} \bigtimes_{j=1}^{4} \left(\mathbf{U}^{(j,k)}\right)^T, \quad \forall k \in \{1,\dots,CK\}. \tag{4}$$

Smallest elements of $\mathcal{S}^{(i,k)}$ are nullified until a threshold for representation error is achieved. Additionally, for data points that are not sparse, a sparsity upper-bound is used to ensure sparsity, for more details see [MHU19]. Finally, once the $i$th data point, $\mathcal{T}^{(i)}$, is projected onto all dictionaries in AMDE, the index of the dictionary that produces the most sparse coefficients and the least reconstruction error is stored as the membership index $\mathbf{m}_i$, where $\mathbf{m} \in \mathbb{R}^{N_t}$.

The light field coefficients that are obtained from (4) can be further compressed by quantization and entropy coding. Initially, the nonzero elements of the sparse tensor $\mathcal{S}^{(i,k)}$ are quantized using Fisher-Jenks classification algorithm [Fis58] which classifies features of a 1D vector using natural breaks in data values by minimizing sum of the squares of the deviations from the class means. The number of cluster centroids is user-defined. Note that for the sparse tensor, we need to store the location of the nonzero coefficients too. The quantized nonzero coefficients (8-bits per coefficient), together with their corresponding locations (32-bits per location, i.e. 8-bits per dimension), are then encoded by the Huffman algorithm. The entropy coded coefficients together with Huffman dictionary are stored on the disk and decoded once the data is loaded to the memory.

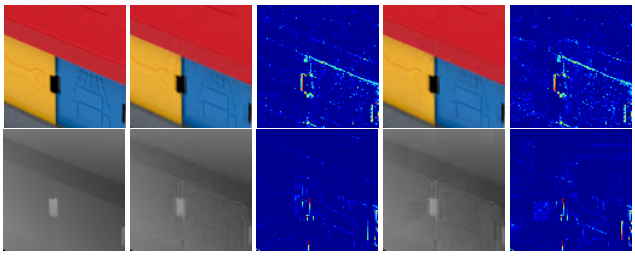


**Figure 3:** *Quantization of sparse coefficient and its effect on the reconstruction quality for the TOY data set.*

## 4. Real-time Reconstruction

One of the key features of AMDE is random local access to each element of the Light field data set. As a result, for a given viewpoint, a single pixel can be reconstructed using a light-weight, GPU-friendly algorithm. The sparse coefficients of all acquired views, $\mathcal{S}^{(i)}$, together with their corresponding dictionaries, $\{\mathbf{U}^{(1,1)},\dots,\mathbf{U}^{(4,CK)}\}$ and their membership matrix $M$ are uploaded to the GPU as textures where each element of a data point $\mathcal{T}^{(i)}$, denoted $\mathcal{T}^{(i)}_{x_1,x_2,x_3,x_4}$, is reconstructed in a shader program as a simple multiplication:

$$\mathcal{T}^{(i)}_{x_1,x_2,x_3,x_4} = \sum_{j=1}^{\tau_i} \mathcal{S}^{(i)}_{l_1^j,l_2^j,l_3^j,l_4^j} \mathbf{U}^{(1,\mathbf{m}_i)}_{x_1,l_1^j} \mathbf{U}^{(2,\mathbf{m}_i)}_{x_2,l_2^j} \mathbf{U}^{(3,\mathbf{m}_i)}_{x_3,l_3^j} \mathbf{U}^{(4,\mathbf{m}_i)}_{x_4,l_4^j}, \tag{5}$$

where $\tau_i$ is the number of nonzero elements in $\mathcal{S}^{(i)}$ and $(l_1^j,l_2^j,l_3^j,l_4^j)$ is the corresponding locations for nonzero elements.

The index map and calibration data are also uploaded to the GPU. A proxy geometry, e.g. a quad, is placed at the center of the bounding sphere of the sampled viewpoints to assist the rendering. During rendering, from the virtual camera position, we calculate a ray-sphere intersection to find the closest intersection point with the bounding sphere of the cameras. The intersection point is converted to spherical coordinate to look up the closest four cameras in the index map as explained in Section 3.1. Each point on the proxy geometry is projected onto each of the four closest cameras, and then projected back to the 3D world using the depth map of each view that is reconstructed in real-time using Equation (5). Subsequently, the final pixel is reconstructed by interpolation between the four points obtained from the four closest cameras.

## 5. Experiments

To test our proposed framework, we simulated the capturing design shown in Figure 1 using Maya and Mental Ray, as explained in Section 3. The calibration and depth maps are generated through the simulation process. The compressed data, together with its dictionaries and corresponding indices, are loaded and transformed into a suitable format to fit on OpenGL textures. As the light field is compressed separately for each ring, we have 13 compressed files corresponding to 13 cameras. The data points for this experiment have a fixed size of $12 \times 16 \times 6 \times 4$ corresponding to $x \times y \times \theta \times c$. We created diverse data sets with different material properties from diffuse to specular and translucent. Table 1 shows the result of our compression for data sets: *POT, STOOL, VASE, TOY,* and *LAMP*. The compression ratio using only the AMDE method varies between 1022:1 for the *POT* with the image resolution of $4000 \times 3000$

| | *Dataset* | *POT* | *STOOL* | *VASE* | *LAMP* | *TOY* |
|---|---|---|---|---|---|---|
| | Resolution | $4000 \times 3000$ | $800 \times 600$ | $800 \times 600$ | $800 \times 600$ | $800 \times 600$ |
| AMDE | Comp. Ratio | 1022:1 | 58:1 | 129:1 | 59:1 | 148:1 |
| | PSNR | 53.11dB | 39.72dB | 47.23dB | 35.57dB | 44.71dB |
| AMDE + Quant. + Huffman | Comp. Ratio | 3054:1 | 130:1 | 294:1 | 128:1 | 329:1 |
| | PSNR | 52.41dB | 39.18dB | 45.96dB | 35.32dB | 44.15dB |

**Table 1:** *The comparison of AMDE results with and without quantization and entropy coding of the sparse coefficients using the Huffman coding. The data sets include depth information.*

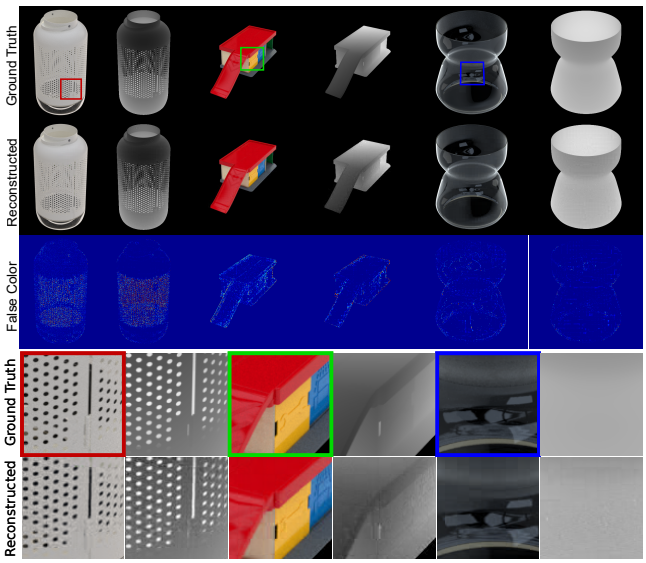


**Figure 4:** *Reconstruction quality after after applying AMDE and sparse coefficient compression of LAMP, TOY, and VASE data sets.*

to 58:1 for *STOOL* data set with a resolution of $800 \times 600$, showing that the compression efficiency is dependent on the content of the scene and image resolution. The upper-bound for the number of the coefficients is fixed to 128 for all scenes. The real-time reconstruction achieves 180 frame per second using a GeForce GTX 1080 Ti graphics card, see the supplementary video. Figure 4 illustrates the reconstruction of the compressed light fields with depth map for *LAMP, TOY* and *VASE* data sets. The reconstruction error, as shown in this figure, is insignificant, which shows the effectiveness of the reconstruction algorithm. The effect of Huffman coding and Fisher-Jenks clustering of sparse coefficients with 128 cluster centers is shown quantitatively in Table1 and visually for the *TOY* data set in Figure3. As Table 1 shows the entropy coding improves the compression ratio by a factor of 3 while preserving the reconstruction quality for most of the scenes. The average source entropy is $H = 4.01$. By exploiting the sparse structure of $\mathcal{S}^{(i,k)}$ we achieved an entropy of $E = 4.06$, which is close to the source.

## 6. Conclusion and Future Work

We presented a complete system from capturing to the rendering of 360° inward-looking spherical light fields. The presented compression technique enables random access to memory, which is suitable for real-time rendering of high-resolution data sets. We proposed a design for capturing high-quality light field data sets in a controlled environment. In the under-sampled regions along the elevation, we used a depth-based view synthesis to enhance the resolution. Our quantization and entropy coding of the sparse coefficients improved the compression ratio. The combination of high compression ratio and high-quality real-time reconstruction makes our system suitable for displaying light fields in HMDs or light field displays.

## 7. Acknowledgments

This work was supported by the strategic research environment ELLIIT, Vinnova through grant 2017-03728: Surgeon's View, and Wallenberg Autonomous Systems and Software Program (WASP).

## References

[AEB06] AHARON M., ELAD M., BRUCKSTEIN A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing 54*, 11 (2006). 2

[BMU19] BARAVDISH G., MIANDJI E., UNGER J.: GPU Accelerated Sparse Representation of Light Fields. In *VISAPP'2019* (Feb 2019). 2

[Fis58] FISHER W. D.: On grouping for maximum homogeneity. *Journal of the American Statistical Association 53*, 284 (1958), 789–798. 3

[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). 1, 2

[HML*19] HAJISHARIF S., MIANDJI E., LARSSON P., TRAN K., UNGER J.: Light field video compression and real time rendering. *Computer Graphics Forum 38*, 7 (2019), 265–276. 1, 3

[KZP*13] KIM C., ZIMMER H., PRITCH Y., SORKINE-HORNUNG A., GROSS M.: Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph. 32*, 4 (July 2013), 73:1–73:12. 2

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), SIGGRAPH '96, ACM, pp. 31–42. 1, 2

[MHU19] MIANDJI E., HAJISHARIF S., UNGER J.: A unified framework for compression and compressed sensing of light fields and light field videos. *ACM Trans. Graph. 38*, 3 (May 2019), 23:1–23:18. 1, 2, 3

[MSOC*19] MILDENHALL B., SRINIVASAN P. P., ORTIZ-CAYON R., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph. 38*, 4 (July 2019), 29:1–29:14. 2

[NLB*05] NG R., LEVOY M., BRÃL'DIF M., DUVAL G., HOROWITZ M., HANRAHAN P.: Light field photography with a hand-held plenopic camera. *Technical Report CTSR 2005-02 CTSR* (01 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 Trans. Graph. 37*, 6 (Dec. 2018), 197:1–197:15. 2

[Ray19] RAYTRIX: 3d light field camera technology, 2019. 2

[WJV*] 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*, SIGGRAPH '05, ACM, pp. 765–776. 2