Light Field Compression via Compact Neural Scene Representation

Jinglei Shi, Christine Guillemot

LIGHT FIELD COMPRESSION VIA COMPACT NEURAL SCENE REPRESENTATION

Jinglei Shi[*] Nankai University

Christine Guillemot INRIA Rennes - Bretagne Atlantique

ABSTRACT

In this paper, we propose a novel light field compression method based on a low rank-constrained neural scene representation. While most existing methods directly compress the light field views, our method first learns a Multi-Layer Perceptron (MLP)-based Neural Radiance Field (NeRF) from the input views. To be able to efficiently compress the NeRF scene representation, the weights of the MLP are optimized under a low-rank constraint using the Alternating Direction Method of Multipliers (ADMM) optimization method. The weights of NeRF are then decomposed into Tensor Train (TT) components which allow us to distill original NeRF network into a slimmer one. The slim NeRF is then refined using a quantization-aware training procedure. Experimental results show that this low rank-constrained NeRF-based light field compression method can achieve better rate-distortion than reference methods, while keeping the free-viewpoint reconstruction capability.

Index Terms— Light Field, Compression, Low Rank, Network Distillation

1. INTRODUCTION

In recent decades, light fields have emerged as promising technology in the field of computational photography. They record the position and direction of light rays in the 3D space after only one single exposure, thus enabling applications such as digital refocusing [1], depth estimation [2] and view synthesis [3] etc. However, the spatio-angular information captured by light fields also means large volumes of data and consequently huge information redundancy, with strong implications on storage, transmission and display systems. Many solutions have been proposed to address this problem which can be roughly categorized as Pseudo Video Sequence (PVS)-based methods, transform-based methods and reconstruction-based ones.

Light field views can be reordered as a pseudo video sequence (PVS), and encoded using video compression standards such as HEVC or its variants [4]. The authors in [5] propose to follow raster or spiral-like scan orders to generate the PVS for compression. State-of-the-art learning-based video compression methods [6, 7, 8] are also applicable to compress light fields. However, due to the different coding tools used for the keyframe coded in the I mode, and for the P and B frames, and due to the rate control used to have high quality reference frames (I to predict all the other frames, P to predict the B frames), the quality varies from one frame to the other. Transformations have also been applied to light fields to reduce their redundancy. A 4D-DCT transform-based solution, known as the Multidimensional Light field Encoder (MuLE) [9], has been adopted in the 4D transform mode of JPEG Pleno. The authors in [10] propose a geometry-aware graph-transform based light field compression method. New models such as homography-based low rank models and mixture of experts have also been investigated in [11] and [12] respectively. Some transform-based methods like [11] work well on narrow-baseline data, but performance quickly drops when data has large baseline with less correlation. Predictive coding solutions have also been designed based on view synthesis methods. A sparse set of light field views is first encoded and used to synthesize (predict) the remaining views using e.g., Depth Image Based Rendering (DIBR), as in [13], transform-assisted [14], or learning-based view synthesis [15, 16] approaches. The performance of view synthesis-based predictive coding solutions depends on the performance of DIBR techniques used for view synthesis, and on the choice of the reference views used for synthesis. Though the aforementioned categories of methods follow different design philosophies, they all directly compress the light field views. NeRF [17], as an implicit scene representation learned from a set of input views, is much less impacted by the camera baseline during training, and does not require prior disparity estimation (e.g. using DIBR techniques), it naturally gets rid of the limitations of the previous methods. Several variants of NeRF have been proposed to improve its performance, to accelerate its training or test efficiency [18], to reduce the number of input views [19], or for deterring aliasing [20].

In this paper, we adopt a simplified NeRF having only one MLP to represent a light field, whose weights are optimized under a low-rank constraint using the ADMM optimization method. These low-rank weights of LR-NeRF are then decomposed into TT components to initialize a slimmer network D(istilled)LR-NeRF, scene information is thus distilled and represented with fewer parameters. These parameters

[*]Most of work has been done when the first author was in INRIA Rennes, funded by the DeepCIM project in the context of the French ANR programme on Artificial Intelligence. Corresponding to jinglei.shi@nankai.edu.cn

are finally quantized with an optimized codebook to obtain a compact neural scene representation of light field, namely Q(uantized)DLR-NeRF. The performance of our proposed QDLR-NeRF has been assessed against HEVC and JPEG Pleno compression standards, and also three recent learning-based video compression methods [6, 7, 8]. Experimental results show that our method achieves a good rate-distortion performance on tested light fields. Moreover, our scene representation-based method, when compared with other methods, can generate coherent views and is not limited by the baseline and depth priors. It is also capable of retrieving views in any viewpoint at decoder side.

2. METHOD

2.1. Overall workflow

The workflow of our method is demonstrated in Fig. 1. Light field information is first learned and stored in NeRF, optimized with respect to the target rank r in ADMM method. The optimized low-rank weights are decomposed to transfer scene information from LR-NeRF to DLR-NeRF. The weights of DLR-NeRF are finally quantized to obtain compact QDLR-NeRF.

2.2. NeRF initialization from light field

NeRF network takes 5D coordinates (x, y, z, θ, ϕ) of light rays as inputs to predict a color and density (RGB, σ), like formulated as follows:

$$RGB, \sigma = F_\Theta(x, y, z, \theta, \phi), \tag{1}$$

during this procedure, scene information is recorded by model parameters $\Theta = \{W_i, b_i\}$, with W_i and b_i are weights and biases of the MLP.

Compared with the original NeRF presented in [17], two modifications are made to adjust to the compression context: a) Instead of using two MLPs, we adopt only one MLP to sample rays, which halves the parameter number with little performance degradation (<0.2dB). b). Since COLMAP package [21] fails to estimated camera pose for narrow-baseline light fields, we simplify the camera pose of structured light fields into focal length f and offset Δ (proportional to the baseline), and set f and Δ as trainable parameters like in [22]. The learning process of NeRF can be described as:

$$\{W_i, b_i, f, \Delta\} = argmin_{\{W_i, b_i, f, \Delta\}} ||c' - c_{gt}||_2^2, \tag{2}$$

where c' and c_{gt} are respectively rendered and ground-truth pixel values.

2.3. Rank optimization in ADMM

To reduce the bitrate needed to encode the NeRF MLP, we can apply TT decomposition on $W_i \in \mathbb{R}^{n_1 \times n_2}, \forall i$ to further decrease the parameter number:

$$W_i = Q_i^1 \cdot Q_i^2, \tag{3}$$

where $Q_i^1 \in \mathbb{R}^{1 \times n_1 \times r}$ and $Q_i^2 \in \mathbb{R}^{r \times n_2 \times 1}$ are the TT components, and r is the target rank. Let's note that other tensor or matrix decomposition methods are totally applicable as well. However, decomposing the full-rank W_i into low-rank matrices Q_i^1 and Q_i^2 would lead to information loss. Similar to [23], we optimize NeRF under a rank constraint, and iteratively update its weights W_i using the ADMM method. The problem is first formulated as:

$$W_i = argmin_{W_i} ||c' - c_{gt}||_2^2, \tag{4}$$

$$s.t.\ rank(W_i) < r, \tag{5}$$

By introducing an indicator function $g(\cdot)$:

$$g(W_i) = \begin{cases} 0, & rank(W_i) < r \\ +\infty, & otherwise. \end{cases} \tag{6}$$

and an auxiliary variable Z, the above problem is reformulated as:

$$argmin_{\{W_i, Z_i\}} ||c' - c_{gt}||_2^2 + g(Z_i) \tag{7}$$

$$s.t.\ W_i = Z_i \tag{8}$$

The augmented Lagrangian in the scaled dual form of the above optimization problem is defined as:

$$\mathcal{L}(W_i, Z_i, U_i) = l(W_i) + g(Z_i) + \frac{\rho}{2}||W_i - Z_i + U_i||_F^2 + \frac{\rho}{2}||U_i||_F^2, \tag{9}$$

with ρ being a positive penalty parameter, U_i being the dual multiplier, and $l = ||c' - c_{gt}||_2^2$ being the rendering error. Such an optimization problem can be solved in an alternative manner as:

$$W_i^{t+1} = argmin_{\{W_i\}} \mathcal{L}(W_i^t, Z_i^t, U_i^t), \tag{10}$$

$$Z_i^{t+1} = argmin_{\{Z_i\}} \mathcal{L}(W_i^{t+1}, Z_i^t, U_i^t), \tag{11}$$

$$U_i^{t+1} = U_i^t + W_i^{t+1} - Z_i^{t+1} \tag{12}$$

where the updates of W_i in Eq. 10 can be achieved by gradient descent, and the updates of Z_i, according to [24], can be solved as:

$$Z_i^{t+1} = \Pi_r(W_i^{t+1} + U_i^t), \tag{13}$$

with Π_r being the operation that decomposes the matrix into the TT format truncated to the desired rank r. This step of rank optimization in ADMM forces the network weights to be low-rank without large performance degradation for TT decomposition in next stage.

2.4. Network Distillation

The optimization in terms of tensor rank allows decomposing W_i^r into TT components with a small approximation error. We can thus use TT components $Q_i^1 \in \mathbb{R}^{1 \times n_1 \times r}$ and $Q_i^2 \in \mathbb{R}^{r \times n_2 \times 1}$ to initialize a smaller (distilled) network after neglecting the first dimension of Q_i^1 and the last dimension of Q_i^2. Let us note that a layer having weight W_i in LR-NeRF

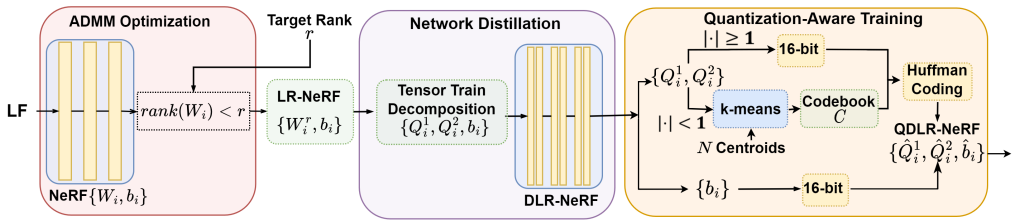

Fig. 1. Overall workflow of our proposed QDLR-NeRF method.

will be decomposed into two layers with weights Q_i^1 and Q_i^2 in DLR-NeRF only if the parameter number is reduced after decomposition.

Network distillation is crucial in our workflow. One may further reduce the model size by quantizing Q_i^1 and Q_i^2, and retrieve $\hat{W}_i^r$ from quantized $\hat{Q}_i^1$ and $\hat{Q}_i^2$ when rendering views. However, the difference between $\hat{W}_i^r$ and W_i^r caused by quantization error makes the workflow unstable and leads to rendering artifacts. The distillation operation allows us to quantize Q_i^1 and Q_i^2 without reconstructing $\hat{W}_i^r$ during rendering, keeping the workflow stable and avoiding rendering artifacts. This step of distillation reduces the number of parameters via the TT decomposition and the initialization of DLR-NeRF. It will also facilitate the quantization in next stage.

2.5. Rate-constrained Quantization

As explained in Sec. 2.4, DLR-NeRF can be further compacted after being quantized. We found that only 0.5% of the parameters in $\{Q_i^1, Q_i^2\}$ are outside the interval $[-1, 1]$, and these parameters are essential for good rendering quality and should not be coarsely quantized, we thus adopt different quantization strategies for values inside and outside $[-1, 1]$.

The values outside $[-1, 1]$ are quantized using a uniform 16-bit quantizer to keep enough precision. While the values within $[-1, 1]$ that occupy most of the proportion are quantized using a non-uniform optimized quantizer. The codebook C of such a quantizer is obtained by applying k-means to cluster parameters into N centroids. Different from the work in [25], where the authors learn layer-wise codebooks, we instead optimize one single codebook for all layers, and adopt a quantization-aware training strategy to reduce errors caused by quantization. More precisely, after quantizing $\{Q_i^1, Q_i^2\}$ of a certain layer $i = n$, we continue to train the network and update $\{Q_i^1, Q_i^2\} \forall i > n$ and $\{b_i\} \forall i$. Once all layers are quantized, we update the learned codebook and start a new iteration. Though more iterations can produce better global codebook, we found in the experiments that one iteration is good enough. During the training process, the biases b_i can compensate in some degree the quantization error of $\{Q_i^1, Q_i^2\}$, we thus prefer to precisely quantize b_i with a 16-bit uniform quantizer.

The quantized matrices $\{Q_i^1, Q_i^2\}$, are then encoded using Huffman coding to further reduce the bitrate. Considering that separately coding the two sets of $\{Q_i^1, Q_i^2\}$ (inside and outside $[-1, 1]$) will require one extra bit as indication flag, we directly code the union of the two sets at the cost of a small bitrate increase to obtain $\hat{Q}_i^1$ and $\hat{Q}_i^2$.

The quantized version of DLR-NeRF, referred to as QDLR-NeRF, is a compact model that represents the input light field. One can transmit the quantized matrices $\hat{Q}_i^1$, $\hat{Q}_i^2$, $\hat{b}_i$ (16 bits), the camera parameters f (32 bits), Δ (32 bits) and the codebook (32 bits) to reconstruct the light field. The quantization finally reduces the number of bits needed for each parameter.

3. EXPERIMENTS

3.1. Training details

We adopt a 8-layer MLP with 256 channels as NeRF architecture. The sampling rate of light ray is set to 128, camera pose parameters f and Δ are initialized to $0.01W$ (W is the image width) and 10, the penalty factor ρ is set to 10. During the training and rank optimization steps of NeRF, learning rate is set to 5×10^{-4} with an exponential decay rate 0.1, each step is trained for 3×10^5 iterations. After initialization of DLR-NeRF, we further finetune it for 2×10^5 iterations for better performance. In the final quantization step, the codebook C is obtained after 20 iterations of clustering in the k-means algorithm. After quantizing each layer, we train the subsequent layers for 3×10^4 iterations. The entire workflow is implemented using the Pytorch learning framework and has been trained using one GPU GeForce RTX 2080Ti with 11GB memory.

3.2. Settings

Both the target rank r and the centroids number N can control the model size. To test the performance of our method at different bitrates, we first optimized the model using different rank values $r = \{150, 90, 70, 40\}$ with a fixed $N = 256$. However, we found that the reconstructed light fields are prone to artifacts when using a rank r inferior to 40. We therefore decrease the centroid number $N = \{256, 64, 32\}$ after setting $r = 40$ to further compress the model. The method has been tested using both real-world and synthetic light fields. We took four Illum-captured scenes *Bikes, Danger, StonePillarsOutside(SPO), FountainVincent2(FV2)* from the EPFL dataset [26] as real-world test data, where each light field has a resolution of $432 \times 624 \times 9 \times 9$. For synthetic data, we took four scenes *boxes, sideboard, cotton, dino* from the HCI dataset [27], where each light field has a resolution of

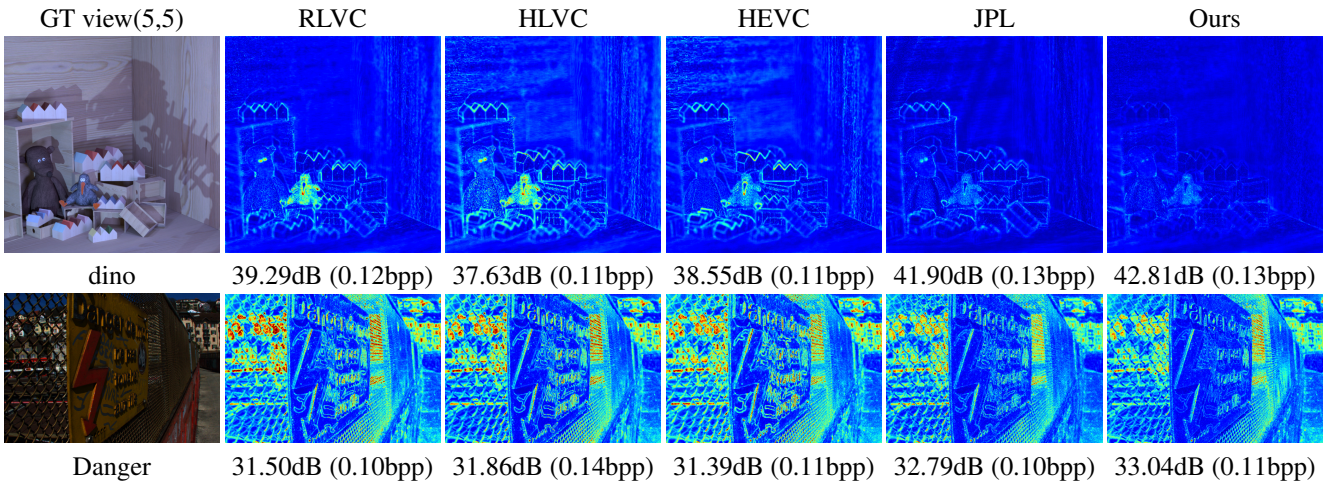

Fig. 2. Averaged error maps of decompressed light fields using different methods, along with the PSNR and bitrate values.

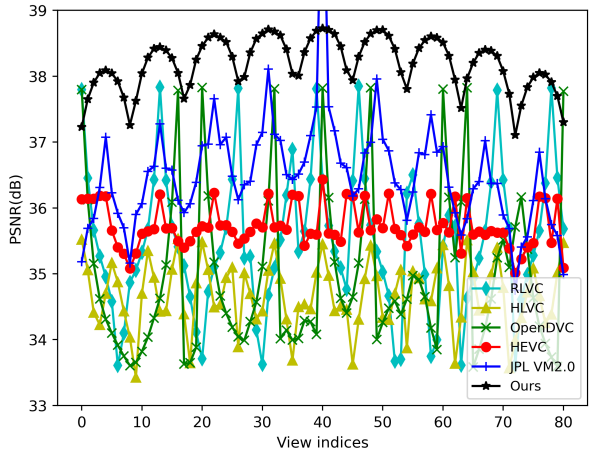

Fig. 3. PSNR variation across views for the scene '*boxes*'.

$512 \times 512 \times 9 \times 9$. We use estimated camera poses in our test in spite of the availability of ground truth camera pose for synthetic data.

3.3. Performance

We compare our method with two typical non-learning-based compression standards HEVC [28] and JPEG Pleno [29], and with three learning-based methods RLVC [8], HLVC [7] and OpenDVC[6]. We assess the BD-PSNR gains using the Bjontegaard measure, and taking the results with HEVC as a reference. Table 1 shows that our method outperforms other compared methods with a large margin. Although Lytro-captured light fields contain artifacts such as vignetting and blurriness caused by hardware limitations, which prevents our NeRF-based method from retrieving a precise scene, our method can still yield good results.

Let us note that some compared methods may select certain views as anchors to compress the other views, leading to a high variation of quality across views (lower view coherence), as shown in Fig. 3. Therefore we show in Fig. 2 the error maps averaged across ALL views under similar bitrates. And by observing Fig. 2 and Fig. 3, we found that our method can produce better decompressed views with both higher inter-view coherence and lower errors. More detailed explanations and further results can be found in [30].

Table 1. BD-PSNR gains with respect to HEVC baseline (covering bpp range 0-0.2). The best results are in bold.

LFs	JPEG Pleno	RLVC	HLVC	OpenDVC	Ours
boxes	0.65	-0.41	-0.79	-1.46	**2.24**
sideboard	1.77	-0.52	-1.75	-2.12	**3.88**
cotton	**0.98**	-2.95	-3.44	-3.18	0.34
dino	2.56	-0.18	-1.07	-1.28	**3.33**
Bikes	0.54	0.19	-0.22	-0.70	**0.67**
Danger	1.17	0.25	-0.05	-0.78	**1.58**
SPO	0.40	0.15	-0.24	-0.60	**0.49**
FV2	0.36	0.25	-0.15	-0.57	**0.49**
Average	1.05	-0.40	-0.96	-1.34	**1.63**

Table 2. PSNR and model size in percentage after each step. PSNR is averaged on 8 tested light fields.

Metrics	NeRF-org	LR-NeRF	DLR-NeRF	QDLR-NeRF
PSNR	41.98dB	38.83dB	38.72dB	38.18dB
Size	100%	50%	16.7%	3.3%

3.4. Contributions of each step

To give insights on the contributions of each functional block, we show in Table 2 the contributions of each step in terms of PSNR and model size. By applying each operation, our method can yield a very compact neural scene representation which has a model size divided by 33, compared with the original NeRF, and with an acceptable quality loss.

4. CONCLUSION

In this paper, we have proposed a novel light field compression method based on compact neural scene representation. Results show that our method can achieve very high compression ratio while keeping better view coherence and reconstruction quality when compared with other methods.

5. REFERENCES

[1] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report (CSTR)*, 2(11):1–11, 2005.

[2] J. Shi, X. Jiang, and C. Guillemot. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Trans. Image Process. (TIP)*, 28(12):5867–5880, Dec 2019.

[3] J. Shi, X. Jiang, and C. Guillemot. Learning fused pixel and feature-based view reconstructions for light fields. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2555–2564, 2020.

[4] C. Conti, P. Nunes, and L. Soares. HEVC-based light field image coding with bi-predicted self-similarity compensation. In *IEEE Int. Conf. on Multimedia Expo. Workshops (ICMEW)*, pages 1–4, 2016.

[5] F. Dai, J. Zhang, Y. Ma, and Y. Zhang. Lenselet image compression scheme based on subaperture images streaming. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 4733–4737, 2015.

[6] R. Yang, L. Van Gool, and R. Timofte. OpenDVC: An open source implementation of the DVC video compression method. *CoRR*, abs/2006.15862, 2020.

[7] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[8] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 15(2):388–401, 2021.

[9] MB. de Carvalho, MP. Pereira, G. Alves, E. da Silva, C. Pagliari, F. Pereira, and V. Testoni. A 4D DCT-based lenslet light field codec. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 435–439, 2018.

[10] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot. Geometry-aware graph transforms for light field compact representation. *IEEE Trans. Image Process. (TIP)*, 29:602–616, 2020.

[11] X. Jiang, M. Le Pendu, R. Farrugia, and C. Guillemot. Light field compression with homography-based low-rank approximation. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 11(7):1132–1145, Oct. 2017.

[12] R. Verhack, T. Sikora, G. Van Wallendael, and P. Lambert. Steered mixture-of-experts for light field images and video: Representation and coding. *IEEE Trans. Multimedia (TMM)*, 22(3):579–593, 2020.

[13] P. Astola and I. Tabus. WaSP: Hierarchical warping, merging, and sparse prediction for light field image compression. In *Eur. Workshop on Vis. Info. Process. (EUVIP)*, pages 1–6, 2018.

[14] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson. Shearlet transform based prediction scheme for light field compression. In *Data Compression Conf. (DCC)*, pages 396–396, 2018.

[15] D. Liu, X. Huang, W. Zhan, L. Ai, X. Zheng, and S. Cheng. View synthesis-based light field image compression using a generative adversarial network. *Information Sciences*, 545:118–131, 2021.

[16] J. Shi, X. Jiang, and C. Guillemot. Deep residual architecture using pixel and feature cues for view synthesis and temporal interpolation. *IEEE Trans. on Comput. Imaging (TCI)*, 8:246–259, 2022.

[17] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. on Computer Vision (ECCV)*, pages 405–421, 2020.

[18] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. (TOG)*, 2022.

[19] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2021.

[20] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis. (ICCV)*, pages 5855–5864, 2021.

[21] J. Schonberger and J. Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4104–4113, 2016.

[22] Z. Wang, S. Wu, W. Xie, M. Chen, and V. Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[23] M. Yin, Y. Sui, S. Liao, and B. Yuan. Towards efficient tensor decomposition-based dnn model compression with optimization framework. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10674–10683, 2021.

[24] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc., 2011.

[25] A. Fan, P. Stock, B. Graham, E. Grave, R. Gribonval, H. Jegou, and A. Joulin. Training with quantization noise for extreme model compression. *Int. Conf. Learn. Represent. (ICLR)*, 2020.

[26] M. Rerabek and T. Ebrahimi. New light field image dataset. In *Int. Conf. on Quality of Multimedia Experience (QoMEX)*, number EPFL-CONF-218363, 2016.

[27] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conf. on Computer Vision (ACCV)*, pages 19–34, 2016.

[28] ISO/IEC JTC 1/SC29. High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 2: High Efficiency Video Coding. ISO/IEC 23008-2:2017. Technical report, 2017.

[29] Jpeg pleno. `https://jpeg.org/jpegpleno/`.

[30] J. Shi and C. Guillemot. Distilled low rank neural radiance field with quantization for light field compression. *arXiv preprint arXiv:2208.00164*, 2022.