IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 6, 2022, accepted April 29, 2022, date of publication May 23, 2022, date of current version June 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3177601

A Hierarchical Approach for Lossy Light Field Compression With Multiple Bit Rates Based on Tucker Decomposition via Random Sketching

JOSHITHA RAVISHANKAR AND MANSI SHARMA
Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India
Corresponding author: Mansi Sharma (mansisharma@ee.iitm.ac.in)

This work was supported in part by the Ministry of Education, Government of India, under Project SB21221952EEIITM008675.

ABSTRACT Recently, there has been extensive progress in developing autostereoscopic platforms for display purposes to present real-world 3D scenes. Light fields are the best emerging choice for computational multi-view autostereoscopic displays since they provide an optimized solution to support direction-dependent outputs simultaneously without sacrificing the resolution. We present a novel light field representation, coding and streaming scheme that efficiently handles large tensor data. Intrinsic redundancies in light field subsets are eliminated through low-rank representation using Tucker decomposition with tensor sketching for various ranks and sketch dimension parameters, making it ideal for streaming and transmission. Apart from removing spatial redundancies, the approximated light field is used to construct a Fourier disparity layers representation to further exploit other non-linear, temporal, intra and inter-view correlations present among the approximated sub-aperture images. Four scanning or view prediction patterns are utilized and the subsets in each pattern hierarchically construct the FDL representation and synthesize subsequent views. Iterative refinement and encoding with HEVC are followed by the final light field reconstruction. The complete end-to-end processing pipeline can flexibly work for multiple bitrates and is adaptable for a variety of multi-view autostereoscopic platforms. The compression performance of the proposed scheme is analyzed on real light fields. We achieved substantial bitrate savings compared to state-of-the-art codecs, while maintaining good reconstruction quality.

INDEX TERMS Light field coding, Fourier disparity layers, HEVC, low-rank tensor decomposition, streaming, tensor sketching.

I. INTRODUCTION

In recent years, light fields [1], [2] have been dominating the market of 3D media and entertainment, with an increased focus on autostereoscopic 3D displays and imaging systems. There exist several technologies for glasses-free 3D viewing [3]–[5]. The popular ones include parallax barriers-based [6]–[8], lenticular 3D displays [9], [10], and stacked layered displays [4], [11]–[16]. Compared to conventional lenticular and parallax-based designs, the layered display can support multiple viewing angles simultaneously without sacrificing the resolution of each view direction, a desirable property for glasses-free 3D viewing. They provide a more realistic viewing experience by presenting different perspectives of the scene and create depth perception and natural motion parallax within a field of view of the display [14], [17]–[21]. In general, autostereoscopic displays are categorized as multi-view displays, and multi-view displays that can accommodate numerous viewing directions are considered light field displays.

The primary challenge to support computational displays is to develop efficient representation, coding and processing schemes for light fields that provide a wide field of view and continuous motion parallax with accurate depth perception from different viewing directions. The displayed images must be direction-dependent and consistent with the target 3D scene appearance. Simultaneously reproducing multi-view images or a light field from many directions without sacrificing the resolution is critical for 3D displays. However, maintaining the practical feasibility of processing large volume of light field data is difficult. The storage and transmission of light fields need to be addressed. It is critical to develop efficient coding and representation solutions for light fields suitable for display and streaming applications. Compressing

The associate editor coordinating the review of this manuscript and approving it for publication was Zihuai Lin.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

light field images efficiently for storage and transmission is a challenge. The raw image captured by a light field camera is huge. For instance, Lytro Illum, the consumer light field camera, produces raw data at 7728 × 5368 resolution and 10 bits-per-pixel (bpp) precision, namely 51,854,880 bytes per shot/frame [22].

While operating on light fields, the representation must also be compact and generalized to provide flexibility in realizing a range of bitrates for compression. Regarding 3D display adaption, the light field representation and coding must be scalable and appropriate for different viewing conditions. The format must be invariant to the display type and allow the depth impression to be easily changed to best meet viewers' preferences for visual comfort. Researchers have undertaken several approaches to compress the light field data. Takahashi *et al.* [4], [11], [13]–[15] have used an analytical framework based on non-negative tensor factorization to approximate the light field for 3D viewing. The analytical solution requires significant computational resources, particularly for the multiplicative update rules used in their implementation [4]. This signal processing approach supports a limited number of viewpoints that can be displayed. The solution is unconstrained and artifacts may be observed between target viewing positions [4]. Also, the analytical approach for light field optimization increases the cost and complexity with multiple layers and directional backlighting. Thus, the constrained non-linear least squares problem involves significant computational resources and requires refined optimization with perceptual error metrics for further reduction in complexity.

Other existing light field coding approaches are designed either to work directly on the raw lenslet image [23]–[27] or operate on individual sub-aperture images (SAIs). Present algorithms either need explicit geometry information [28], [29], scene content information [30], disparity information [31], or are epipolar plane image-based and multiplane image-based [32]–[34], view prediction based learning schemes [35]–[38] or methods considering light field data as a pseudo video sequence [22], [39]–[41]. The approaches that arrange SAIs as pseudo-temporal sequences are constrained to a one-dimensional coding structure. Non-linear correlation in horizontal and vertical directions among adjacent SAIs are not carefully considered in such coding models [22], [39]–[41]. The geometry-based or learning-based methods usually demand accurate mathematical models for parameter estimation, otherwise suffer from the primary limitation of low quality reconstruction. Moreover, these compression approaches are not explicitly designed for computational multi-view displays. They are designed to train a specific system architecture and support only certain bitrates. An elaborate analysis of related light field coding schemes and their shortcomings has been provided in section II.

Our goal is to develop a compact light field representation, coding and streaming scheme that can provide extended functionality and backward compatibility with existing transmission and broadcasting standards for glasses-free light field displays. In this article, a novel hierarchical scheme that can efficiently handle light field data is presented. The proposed scheme takes multiple light field view subsets in different scanning orders; Circular-2 (C_2), Circular-4 (C_4), Hierarchical-2 (H_2) and Hierarchical-4 (H_4), as the input in the very first block (Fig 1). Low-rank representation using Tucker decomposition [42] eliminates intrinsic redundancies present in view subsets for various ranks. Incorporating tensor sketching for multiple sketch dimension parameters enables the Tucker decomposition to work with large tensors easily in just one single pass and performs with good accuracy and reduced memory requirement. This makes our scheme ideal for streaming and transmission. In the next block, the approximated light field is further analyzed by sampling it in the depth dimension by decomposing the scene as a discrete sum of Fourier Disparity layers (FDL) [43]. This exploits additional intra-view, inter-view, and non-linear redundancies among adjacent views in both horizontal and vertical directions of sub-aperture images (SAIs) and allows scalable light field coding. The subsets in each scanning pattern hierarchically construct the FDL representation and synthesize subsequent views. Iterative refinement and encoding with high efficiency video coding (HEVC) are followed by the final light field reconstruction in the last block.

The complete processing pipeline operates as a single integrated system and can flexibly work for a range of multiple bitrates by varying the ranks and sketch dimension parameters. Our implementation has reduced memory requirements because the entire tensor light field data need not be stored in full. Also, since the Fourier transform of real signals is symmetric, only half the frequencies are computed in the FDL representation. Compression performance of the proposed scheme was analyzed on real light fields in the four chosen patterns and substantial bitrate savings compared to state-of-the-art codecs are achieved, while maintaining good reconstruction quality. We accomplish superior results in terms of the size of the encoded bitstream file as well. There are also significant PSNR gains obtained in our hybrid model compared to all anchor codecs.

Additionally, there are more implications of the proposed hybrid representation and coding model which make it quite versatile. It is a scheme that can permit view interpolation or extrapolation and generalized rendering by shifting approximated SAIs in the depth dimension instead of two angular dimensions. The combined Tucker decomposition with sketching and FDL representation allows for streaming, and filters the noise and corrects color inconsistencies between reconstructed views [43]. Our scheme does not require any explicit transmission of complete disparity maps unlike JPEG Pleno [44]–[47]. Hence, significant overheads are saved in the proposed model for low bitrate coding scenarios. The proposed hybrid representation and coding scheme is also suitable for GPU parallelization since it is built upon simple linear algebra operations carried out independently in the spatial and frequency domains. On the whole, our model leverages compact representation of light field view subsets

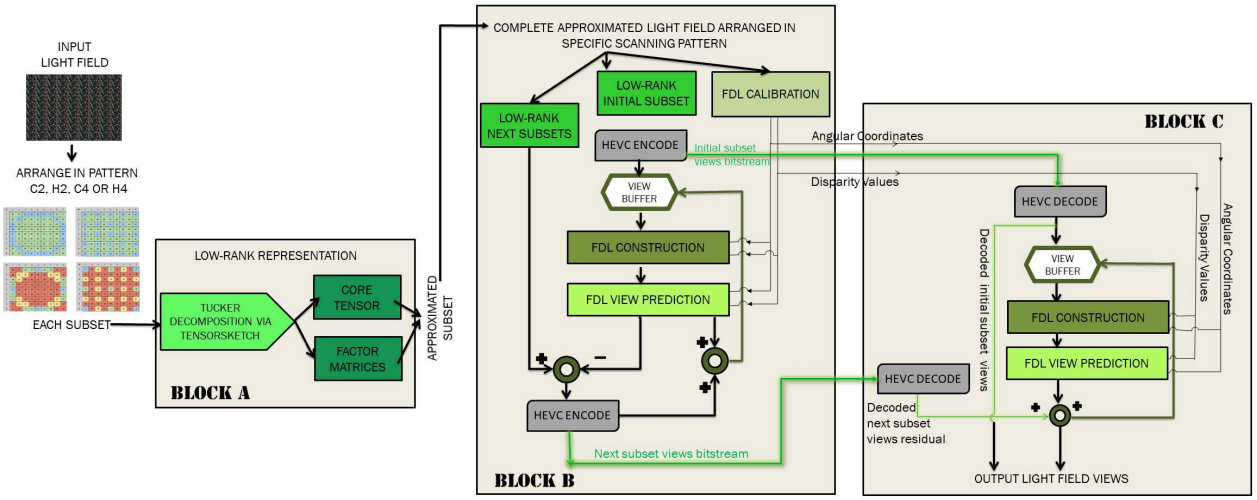
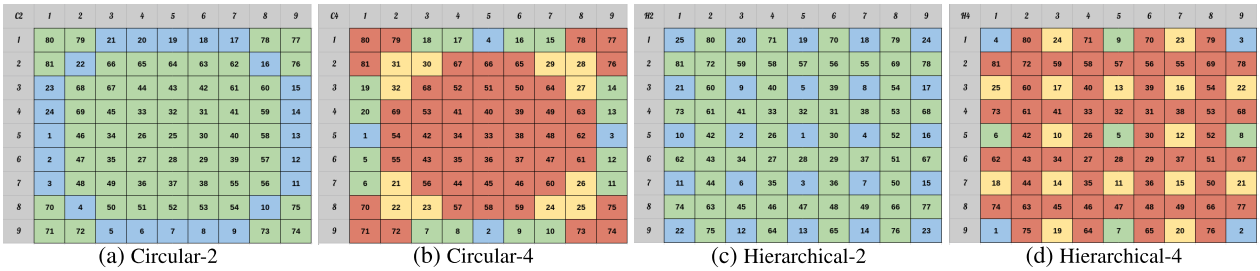


FIGURE 1. The complete pipeline of the proposed scheme with three primary components, BLOCK A, B and C. Light field view subsets in different scanning orders are taken as inputs in BLOCK A and are low-rank approximated using Tucker decomposition via tensor sketching for various ranks and sketch dimension parameters. Intrinsic redundancies are removed in a single pass with a reduced memory requirement. In BLOCK B, the approximated light field views are used to construct Fourier disparity layer (FDL) representation iteratively. Additional intra-view, inter-view, and non-linear redundancies in both horizontal and vertical directions of light field structure are removed in the frequency domain. Finally, the decoding and reconstruction of the light field is performed in BLOCK C.



(a) Circular-2 (b) Circular-4 (c) Hierarchical-2 (d) Hierarchical-4

FIGURE 2. The light field view prediction orders C_2, C_4, H_2 & H_4. Views in blue, green, yellow & orange form the first, second, third and fourth subset respectively.

in the spatial domain making it suitable for streaming and further processing low-rank approximated light field in the FDL-based frequency domain that results in improved coding efficiency.

The rest of this article is organized into four major sections. Section II describes various existing light field compression approaches and their shortcomings. The elaborate description of the proposed scheme is presented in Section III. We have elaborated our experiments specifying the implementation, results, and analysis in Section IV. In Section V, we give the conclusion with comprehensive findings of our proposed scheme and implications of future work.

II. RELATED WORK

The majority of existing light field coding approaches are not directly applicable for multi-layered displays. Current coding techniques are commonly classified into two categories; lenslet-based or approaches based on perspective sub-aperture images (SAIs). The direct compression on the raw light field lenslet image captured by plenoptic camera exploits the spatial redundancy between the microimages [23]–[27]. This approach is usually based on the existing image/video coding standards, such as JPEG [44]–[47] or high efficiency video coding (HEVC) [48].

Li *et al.* [23] incorporated a full inter-microlenses prediction scheme in HEVC intra prediction to remove the redundancy in lenslet images. The raw light field was partitioned into tiles in [24], followed by pseudo-temporal sequencing and HEVC temporal predictive coding. Li *et al.* [25] introduced inter and bi-prediction capability within the HEVC intra prediction module based on references taken from already encoded parts of the image. Liu *et al.* [27] proposed a method to classify the HEVC prediction units (PU) into three different categories based on texture homogeneity. In general, the unique light field structural characteristics make it harder to achieve good prediction accuracy in the lenslet image regions with complex textures. Moreover, there are numerous microimages in the raw light field image which need careful handling. They are of low resolution and require customized reshaping before being fed into an HEVC encoder. Ideally, the lenslet-based compression solutions need to transmit camera parameters for further processing, and thus the coding burden on the compressed data stream increases. These drawbacks encouraged researchers to extract the SAIs from raw plenoptic image and explore compression possibilities on these views.

Various schemes branch out under perspective SAIs based compression of light fields; content-based compression [30], disparity-based [28], [29], [31], geometry-based [32]–[34], pseudo-sequence-based [22], [39]–[41], [49], view synthesis based [35], [36], [50], [51], and learning-based compression methods [37], [38], [52]–[57]. Hu *et al.* [30] proposed a two-layer compression architecture that encoded the high and low-frequency components of the sub-aperture views differently and reconstructed the final light field using a graph neural network. Their method discarded a portion of viewpoint data during the encoding process, which could not be recovered from remaining viewpoints later. Moreover, the scheme's reconstruction algorithms needed improvement with a better mathematical model for parameter estimation.

Under disparity-based coding, Zhao and Chen [29] proposed a sparse sampling method with a linear approximation prior to utilize the similarity between the sub-aperture views. However, the linear prior was not accurate enough to describe the correlation between the viewpoints. A homography-based low-rank approximation method called HLRA was introduced by Jiang *et al.* [28]. They aligned sub-aperture views using global homography or multiple homographies and low-rank approximated them followed by encoding with HEVC. This approach depends on how much the disparities across views varied and did not optimally reduce the low-rank approximation error for light fields with large baselines. Dib *et al.* [31] recently proposed a novel parametric disparity estimation method to support the low-rank approximation using super rays to efficiently expose redundancy across the different views. Their scheme performed well for medium bitrates, but it required additional information that added to the coding burden.

The light field geometry-based compression method by Vagharshakyan *et al.* [32] utilizes sparse representation of epipolar-plane images in the shearlet transform domain. Ahmad *et al.* [33] categorized light field views into key and decimated views. The key views are compressed using the multi-view extension of HEVC, and decimated views are predicted from the compressed key views and a residual bitstream. Chen *et al.* [34] used essential views to predict a global multiplane representation which further predicted all views. In predictive compression methods, the light field sub-aperture views are reordered as pseudo sequences. Liu *et al.* [22] compressed the central view first as the I-frame (intra frame) followed by the remaining views as P-frames in a symmetric, 2D hierarchical order. Another proposal by Li *et al.* [49] involved division of all light field views into four quadrants and a hierarchical coding structure within each quadrant. Ahmad *et al.* [39] utilized the multi-view extension of HEVC (MV-HEVC) to compress the light field in the form of a multi-view sequence. The extension of their work [40] highlighted the usage of hierarchical levels where views belonging to higher levels are assigned with better quality and predicted lower-level views. In another SAI-based technique, synthesized virtual reference frames were generated from Adaptive Separable Convolution Network [41]. Such frames were considered as extra reference candidates in a hierarchical coding structure for MV-HEVC to further exploit intrinsic similarities in light field images.

Senoh *et al.* [50] estimated depth maps from few reference views and encoded them along with their depth information. The remaining views were then synthesized from the decoded views and depth maps. A block-basis estimation of views from translated reference views was proposed in [36], where residuals of the estimated views were also transmitted to the decoder along with the rest of the view estimation parameters. The Multi-view Video plus Depth (MVD) structure was adopted for depth image-based rendering to synthesize the intermediate SAIs in [51]. However, the light field is heavily cropped in this scheme, and only the central 7×7 or 9×9 views are used. A pair of steps to generate noise-refined depth maps for selected perspective views was elaborated in [35]. This scheme synthesized unselected views from the selected encoded views using depth image-based rendering. It required a better reconstruction solution to improve the overall performance.

Bakir *et al.* [54] presented a deep-learning-based scheme on the decoder side to improve the reconstruction quality of SAIs. Similarly, Zhao *et al.* [55] encoded only sparsely sampled SAIs, while the remaining SAIs were synthesized using a convolutional neural network with the decoded sampled SAIs as priors. However, these methods require large-scale and diverse training samples, and high quality of reconstructed views is only obtained if at least half the SAIs are taken as reference. Schiopu and Munteanu [37] proposed a novel network that synthesized the entire light field image as an array of synthesized macro pixels in one step. In [56], light field videos are compressed framewise by identifying a region of interest (ROI), a complex non-ROI, and a smooth non-ROI. A generative adversarial network-based (GAN) unsampled SAI generation is proposed in [57] and Liu *et al.* [38] also adopt a GAN framework to boost the light field compression. They use an image group-based sampling method to reduce more sub-aperture image redundancy and maintain the reconstruction quality.

The existing light field coding or compression techniques are not adaptable for a variety of autostereoscopic 3D displays based on multiple views. These schemes usually train a system (or network) to support only specific bitrates during the compression. Such coding solutions do not have scalable signal representation and do not flexibly stream light field contents at various bitrates. Moreover, there is a need for suitable algorithms that have extended functionality, such as N-view generation and backward compatibility with existing transmission and broadcasting standards for present-day 2D or 3D displays.

In our previous work Ravishankar *et al.* [58], multiplicative layers from complete 13×13 light fields were learnt using a convolutional neural network. Spatial and temporal correlations present among SAIs were taken into account and the hidden low-rank structure of the multiplicative layers was

analyzed on a Krylov subspace. Factorization derived from Block Krylov singular value decomposition (BK-SVD) was performed, followed by encoding with HEVC. On the other hand, the current scheme implements an entirely different approach, including a varied mathematical formulation. The proposed light field representation, coding and streaming scheme handles 9×9 light fields divided into view subsets based on four scanning patterns. The image subsets are taken as tensors and low-rank approximated using Tucker decomposition [42] for various ranks. Incorporation of tensor sketching for multiple sketch dimension parameters enables the Tucker decomposition to work with large tensors easily in just one single pass and it performs with good accuracy and reduced memory requirements. Once the intrinsic redundancies present in the view subsets are removed by Tucker decomposition with sketching, the approximated light field is analyzed by sampling it in the depth dimension and decomposing the scene as a discrete sum of Fourier Disparity layers [43]. This exploits additional intra-view, inter-view, and non-linear redundancies among adjacent views in both horizontal and vertical directions and further allows scalable light field coding. Further, signal representation in the proposed scheme is more flexible to support extended functionality such as light field view interpolation or extrapolation and is appropriate for more general rendering by shifting approximated SAIs in the depth dimension instead of two angular dimensions. Thus, the proposed formulation differs from our previous work [58].

In our other approach [59] and its extended version [60], we handled multiplicative layers of view subsets of the light field and low-rank approximated them with BK-SVD for different ranks. The multiplicative layers were optimized using trained CNNs and the scheme in general was more suited for multi-layered displays. Mathematically, our proposed scheme again varies from [59], [60] in terms of the formulation and implications of using Tucker decomposition via sketching. It provides streaming functionality and the implementation by itself has reduced memory requirements because the entire tensor light field data need not be stored in full. Overall, our previous coding and representation approaches in [58]–[60] are more adaptable to layered displays. Such display consists of a few layers stacked with small intervals with pixels carrying out light ray operations [4], [11]–[16]. The display appearance could vary over the observed viewpoints after the light rays pass through different combinations of pixels. We optimized multiplicative layer patterns using convolutional neural networks and low-rank approximated them to display a 3D scene. These multiplicative layers can be physically implemented using stacked light attenuating or liquid crystal display panels and a backlight [4]. Thus, our previous schemes were suited for stacked layered designs that could be applied to light-field projections [61], [62], head-mounted displays [63], [64], and table-top displays [65]. The proposed scheme in this paper can support more general representation, streaming, coding and rendering for multi-view/light field display applications. Our algorithm works at the level of the sub-aperture image in both spatial and frequency domains. The scheme can achieve coding gains and the representation is flexible to provide several extended functionalities. Hence, it can be adaptable for a variety of multi-view autostereoscopic displays. It is also possible to adapt the proposed scheme for stacked layered-based displays by performing operations considering multiplicative layers instead of sub-aperture images of view subsets.

In [66], we modeled coded-aperture cameras using convolutional autoencoders and obtained optimal acquired images for processing dynamic light field content. The target was to efficiently encode a light field of a moving scene or a 3D scene changing over time. We demonstrated the reduction of a dynamic light field (with five camera viewpoints, containing 5×5 SAIs each) into single acquired images per time frame by producing the best aperture patterns using a deep neural network. The acquired images were then approximated using a hybrid Tucker-TS-Vector Quantization (HTTSVQ) algorithm. Thus, we simulated a coded aperture camera for acquisition of dynamic light field. In contrast, in the present formulation, the input SAIs are extracted from lenslet images captured by light field cameras. Also, in the proposed scheme, hierarchical scanning patterns of light field subsets are used and redundancies are removed using Tucker-TS algorithm which is different from HTTSVQ presented in [66]. Additionally, the proposed scheme operates in both the spatial and frequency domain to support various applications. The two schemes are not directly comparable since we deal with static light fields in the present article but handle dynamic light field content in [66]. Furthermore, the proposed light field representation scheme has extended functionality to enable general rendering for any given viewpoint, aperture size and depth of focus. Table 1 gives the comparative overview showing the key differences between our previous approaches [58]–[60] and the proposed scheme.

III. PROPOSED SCHEME

The complete workflow of our proposed representation and coding scheme with three main blocks is illustrated in Fig. 1. An input light field is taken in a scanning orders Circular-2 (C_2), Circular-4 (C_4), Hierarchical-2 (H_2) or Hierarchical-4 (H_4). In BLOCK A, the intrinsic redundancies present in the view subsets of the scanning patterns are effectively removed during low-rank representation using Tucker decomposition with tensor sketching for various ranks and sketch dimension parameters [42]. This enables suitable streaming and transmission capabilities of the light field as well. Next in BLOCK B, the entire approximated light field is divided into view subsets based on the same scanning pattern. These approximated subsets are used to construct Fourier disparity layer (FDL) representation of light fields [43]. There exist non-linear correlations between neighboring sub-aperture views in both horizontal and vertical directions in the light field structure. We target these redundancies between adjacent light field views by processing in the Fourier domain as specified by different scanning or predication orders. The

TABLE 1. Comparative table depicting the differences between the proposed scheme and our previous approaches.

Proposed Scheme	Ravishankar et al. [58]	Ravishankar et al. [59, 60]
• 9x9 input light field views are divided into view subsets based on chosen hierarchical scanning orders. • Subsets of views undergo Tucker decomposition via tensor sketching for efficient coding and streaming of light field considering different ranks and sketch dimension parameters. This removes intrinsic redundancies and outputs an approximated light field. • View subsets taken as tensor data are efficiently processed in a single pass itself, without the need to store in full. • Intra-view, inter-view and other non-linear redundancies in the approximated light field are removed while processing in the Fourier domain using FDL • The proposed coding scheme provides streaming capability and allows scalable light field representation that is flexible to support multiple bitrates and extended functionality for more general rendering of apparent viewpoints, aperture size and depth of focus.	• 13x13 SAIs of a complete light field simultaneously given as input • Does not follow a hierarchical pattern of input views in the form of subsets and thus can not capture non-linearity in the light field structure. • CNNs are trained to optimize multiplicative layers that can be implemented using LCDs in the multi-layered displays. • Low-rank approximation using BKSVD is performed for varied ranks to remove spatial redundancies in multiplicative layers. • Algorithm needs to store all information at a time for applying BKSVD algorithm, hence can not handle streamed data and does not support scalable light field coding. • The light field is reconstructed using approximated multiplicative layers but the signal processing is not flexible to support extended functionality such as rendering and N-view synthesis.	• 9x9 input SAIs which are then divided into view subsets based on hierarchical patterns • Different CNNs are required to be trained to produce optimized multiplicative layers for each view subset • BKSVD is performed on the multiplicative layers of each subset • BKSVD is not a one-pass algorithm and can not handle streamed data • Approximated view subsets are then processed in the Fourier domain • The low-rank approximation step using BK-SVD is completely different and does not involve any tensor sketching to reduce memory usage of the decomposition.

light field is then iteratively reconstructed by the FDL representation in a hierarchical fashion.

Angular coordinates and disparity values of each view of the low-rank approximated light field (at different ranks and quantization parameters) are found through FDL calibration and directly transmitted to the decoder (BLOCK C) as metadata [67]. The initial approximated light field subset is used in construction of the FDL representation, which is further employed to synthesize the subsequent subsets of views. The correlations in prediction residue are removed, and a more accurate FDL representation is constructed from previously encoded subsets. Thus, we iteratively refine the FDL representation in BLOCK B until all the approximated light field views are encoded. The decoding scheme is depicted in BLOCK C. Here, angular coordinates and disparity values of the low-rank approximated light field, along with the encoded bitstreams of approximated subsets are utilized for the final light field reconstruction. Each block of the proposed compression scheme is explained in the following sections.

A. LOW-RANK TUCKER DECOMPOSITION VIA TENSOR SKETCHING

Algorithms that adopt randomized methods have been popularly used to perform tensor decompositions. Sketching techniques and various randomized methods to compute Higher-order singular value decomposition of large tensor data have been developed in [68]–[74]. The Tucker decomposition is one such technique to decompose tensors into core and factor matrices. The standard approach to perform Tucker decomposition is using aternating least-squares (ALS) or the higher-order orthogonal iteration (HOOI). The TUCKER-ALS [75], Memory Efficient Tucker (MET) [71] and HOOI version of MACH algorithm in [70] run out of memory during decomposition of large tensors and are not one-pass algorithms. Current sketching techniques involved in the decomposition have design matrices as Kronecker products of factor matrices which result in large computations. These Kronecker products are substantially huge to be formed and stored in the RAM, which led to development of *TensorSketch* [76], [77].

Based on the streaming properties of *TensorSketch* [78], Malik and Becker [42] have developed single-pass algorithms that are highly efficient for low-rank decomposition. These methods can compress large tensor data without the need to store it in full by streaming the tensor elements. The algorithms only require a single pass of the data and perform with good accuracy and reduced memory requirements. Motivated by this work, we have proposed our light field coding, transmission and streaming scheme that can handle multiple bitrates of data. The standard Tucker decomposition of a general tensor $\mathcal{X}$ is defined as

$$\begin{aligned}\mathcal{X} &= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \\ &=: \left[\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\right], \end{aligned} \tag{1}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is called the core tensor, and every $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ is known as a factor matrix. By assuming that the factor matrices have orthonormal columns, $\mathcal{X}$ in (1) is a rank-$(R_1, R_2, \ldots, R_N)$ tensor.

Given a light field [1], [2], we take each view subset as a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, an array of dimension N. Decomposing each data tensor $\mathcal{Y}$ using Tucker decomposition can be formulated as

$$\underset{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}}{\arg\min} \left\| \mathcal{Y} - \left[\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\right] \right\| \tag{2}$$

Alternating least-squares (ALS) is an standard approach to solve the optimization problem in (2). It can be rewritten appropriately and solved by repeating the following steps until convergence:

1. For $n = 1, \ldots, N$, update

$$\mathbf{A}^{(n)} = \underset{\mathbf{A} \in \mathbb{R}^{I_n \times R_n}}{\arg\min} \left\| \left(\bigotimes_{i=N, i \neq n}^{1} \mathbf{A}^{(i)} \right) \mathbf{G}_{(n)}^{\top} \mathbf{A}^{\top} - \mathbf{Y}_{(n)}^{\top} \right\|_F^2 \quad (3)$$

2. Update

$$\mathcal{G} = \underset{\mathcal{Z} \in \mathbb{R}^{R_1 \times \cdots \times R_N}}{\arg\min} \left\| \left(\bigotimes_{i=N}^{1} \mathbf{A}^{(i)} \right) \mathbf{z}_{(:)} - \mathbf{y}_{(:)} \right\|_2^2 \quad (4)$$

Here, $\bigotimes$ indicates the Kronecker products operation and $\mathbf{z}_{(:)}$ and $\mathbf{y}_{(:)}$ denote the vectorization of $\mathcal{Z}$ and $\mathcal{Y}$ respectively.

While incorporating random sketching in the Tucker decomposition equations above, it is crucial to consider the large computation due to Kronecker products involved in the algorithm. *TensorSketch* [78] is a restricted or specialized version of *CountSketch* [79], and can be used to efficiently approximate the solutions of large overdetermined least-squares problems involving such Kronecker products. By incorporating random sketching using *TensorSketch* in carrying out Tucker decomposition of large tensors, streamed data can be handled in a single pass [42]. We briefly introduce both the above mentioned sketching techniques in the following subsections.

1) COUNTSKETCH

Consider the overdetermined least squares problem

$$\mathbf{x}^* \overset{def}{=} \underset{\mathbf{x} \in \mathbb{R}^R}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \quad (5)$$

where, $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{y} \in \mathbb{R}^I$ and $I \gg R$. *CountSketch* [42], [79] enables the reduction of this problem size by solving

$$\mathbf{x}' \overset{def}{=} \underset{\mathbf{x} \in \mathbb{R}^R}{\arg\min} \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{y}\|_2 \quad (6)$$

instead. Here $\mathbf{S}$ is a subspace embedding matrix such that $\mathbf{S} : \mathbb{R}^I \to \mathbb{R}^J$ with $J \ll I$, and $\mathbf{S} = \mathbf{P}\mathbf{D}$, where

- $\mathbf{P} \in \mathbb{R}^{J \times I}$ has $p_{h(i),i} = 1$ and all other entries $= 0$;
- $h : [I] \to [J]$ is a random map such that $(\forall i \in [I])(\forall j \in [J])\mathbb{P}(h(i) = j) = 1/J$; and
- $\mathbf{D} \in \mathbb{R}^{I \times I}$ is a diagonal matrix with entries $+1$ or -1, with equal probability.

When the sketch dimension J is sufficiently large, the sketch problem (6) satisfies

$$\left\|\mathbf{A}\mathbf{x}' - \mathbf{y}\right\|_2 \leq (1 + \epsilon) \left\|\mathbf{A}\mathbf{x}^* - \mathbf{y}\right\|_2 \quad (7)$$

with high probability for a fixed $\epsilon > 0$. The *CountSketch* operator $\mathbf{S}$ can be applied to matrix $\mathbf{A}$ in only $\mathcal{O}(\text{nnz}(\mathbf{A}))$ and avoids storing $\mathbf{S}$ as a full matrix.

2) TensorSketch

TensorSketch [42], [78] is a randomized method which reduces the cost and memory usage of computations in alternating least-squares approaches. It is a restricted version of *CountSketch* that operates on matrix $\mathbf{A} \in \mathbb{R}^{L \times M}$, $L \gg M$, and $\mathbf{y} \in \mathbb{R}^L$. The linear map $\mathbf{T} : \mathbb{R}^L \to \mathbb{R}^J$ with $J \ll L$ reduces the size of the problem (5) to

$$\tilde{\mathbf{x}} \overset{def}{=} \underset{\mathbf{x} \in \mathbb{R}^M}{\arg\min} \|\mathbf{T}\mathbf{A}\mathbf{x} - \mathbf{T}\mathbf{y}\|_2 . \quad (8)$$

If J is sufficiently large and $\epsilon > 0$, with high probability $\left\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\right\|_2 \leq (1 + \epsilon) min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}^* - \mathbf{y}\|_2$.

Suppose matrix $\mathbf{A} \in \mathbb{R}^{I \times R}$ is of the form $A = A^{(N)} \otimes A^{(N-1)} \otimes \cdots \otimes A^{(1)}$ with each $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $I_n > R_n$ such that $I \overset{def}{=} \prod_n I_n$ and $R \overset{def}{=} \prod_n R_n$. *TensorSketch* allows sketching of $\mathbf{A}$ as $\mathbf{T}\mathbf{A}$ without ever having to form the full matrix, by sketching each factor matrix $\mathbf{A}^{(n)}$ individually. The corresponding sketch of $\mathbf{A}$ is then computed efficiently using the fast Fourier transform (FFT) as

$$\begin{aligned} \mathbf{T}\mathbf{A} &= \mathbf{T} \bigotimes_{n=N}^{1} \mathbf{A}^{(n)} \\ &= \text{FFT}^{-1} \left(\left(\bigodot_{n=N}^{1} \left(\text{FFT} \left(\mathbf{S}^{(n)} \mathbf{A}^{(n)} \right) \right)^{\top} \right)^{\top} \right) \end{aligned} \quad (9)$$

where, $S^{(n)} \in \mathbb{R}^{J \times I}$ is an independent *CountSketch* operator for each factor matrix $A^{(n)}$.

TensorSketch can be applied on the least square problems in (3) and (4) to solve the resulting smaller problems efficiently. The core and factor matrices are first randomly initialized considering each element as i.i.d in U$(-1, 1)$ and then orthogonalized. Since the design matrix in (4) has more rows than in (3), two *CountSketch* operators $\mathbf{S}_1^{(n)} \in \mathbb{R}^{J_1 \times I_n}$ and $\mathbf{S}_2^{(n)} \in \mathbb{R}^{J_2 \times I_n}$ are then needed to construct the *TensorSketch* operators $\mathbf{T}^{(n)} \in \mathbb{R}^{J_1 \times \prod_{i \neq n} I_i}$, for $n \in [N]$, and $\mathbf{T}^{(N+1)} \in \mathbb{R}^{J_2 \times \prod_i I_i}$ respectively. J_1 and J_2 are the two target dimensions such that $J_2 > J_1$. For practical implementation, *CountSketch* operator sets were chosen with different target sketch dimensions $J_1 = KR^{(N-1)}$ and $J_2 = KR^N$, for sketch dimension parameter $K > 4$ and rank R as suggested by Malik and Becker [42]. Explicitly, there is no need to store the two sketch operators as matrices in the implementation. Instead, only the function h and diagonal of $\mathbf{D}$ as defined previously is generated and stored for each *CountSketch* operator.

By applying *TensorSketch* operators on the Kronecker product matrices involved in (3) and (4), the equations are now modified to:

1. For $n = 1, \ldots, N$, update

$$\mathbf{A}^{(n)} = \underset{\mathbf{A}}{\arg\min} \left\| \left(\mathbf{T}^{(n)} \bigotimes_{i=N, i \neq n}^{1} \mathbf{A}^{(i)} \right) \mathbf{G}_{(n)}^{\top} \mathbf{A}^{\top} - \mathbf{T}^{(n)} \mathbf{Y}_{(n)}^{\top} \right\|_F^2 \quad (10)$$

2. Update

$$\mathcal{G} = \underset{\mathcal{Z}}{\arg\min} \left\| \left((\mathbf{T}^{(N+1)} \bigotimes_{i=N}^{1} \mathbf{A}^{(i)} \right) \mathbf{z}_{(:)} - \mathbf{T}^{(N+1)} \mathbf{y}_{(:)} \right\|_2^2 \quad (11)$$

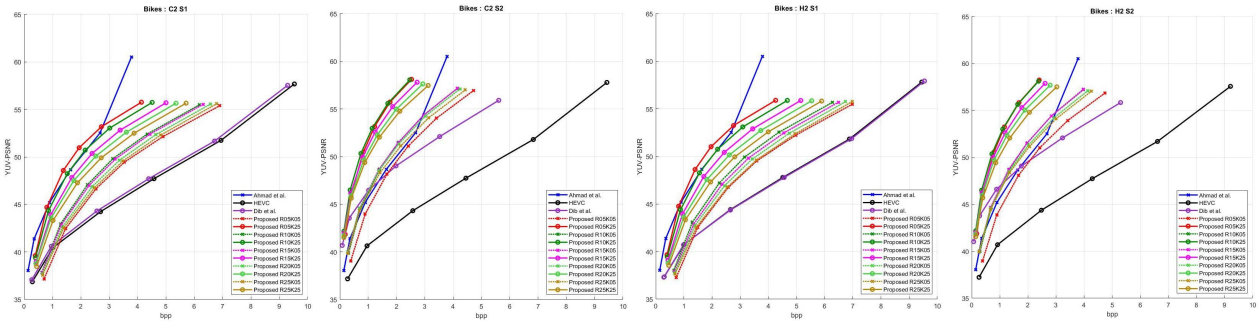


FIGURE 3. BPP vs YUV-PSNR curves for the proposed compression scheme, Ahmad *et al.* [40], Dib *et al.* [67] and HEVC [48] for *Bikes* in the C_2 and H_2 patterns.

The computations of $\mathbf{T}^{(n)}\mathbf{Y}_{(n)}^{\top}$ and $\mathbf{T}^{(N+1)}y_{(:)}$ can be performed only once implicitly without forming any full sketching matrices as they do not change throughout the algorithm. At every step, the factor matrices are orthogonalized and core matrix is updated. Once the problem converges, the view subset is decomposed to its core tensor $\mathcal{G}$ and factor matrices $\mathbf{A}^{(n)}, n \in [N]$. To obtain the low-rank approximated light field view subset, *n-mode tensor-times-matrix (TTM) product* of the core and factor matrices was performed as

$$\hat{\mathcal{Y}} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}. \tag{12}$$

Thus, at the end of BLOCK A (Figure. 1) of the proposed scheme, the low-rank approximated light field is obtained. It is further processed in the Fourier domain to handle intra-view, inter-view and other non-linear redundancies present among adjacent views in both horizontal and vertical directions of sub-aperture images. We have presented this as BLOCK B in the next section.

B. FOURIER DISPARITY LAYERS (FDL) REPRESENTATION

The low-rank representation of the light field with Tucker decomposition using tensor sketching makes it effectively represented for streaming and transmission. Further, our hierarchical coding scheme works in the Fourier domain by constructing a suitable Fourier disparity layers representation [43] which samples the approximated light field in the disparity dimension by decomposing it as a discrete sum of layers. The layers are constructed from the approximated light field sub-aperture views through a regularized least square regression performed independently at each spatial frequency. The FDL representation has been shown to be effective for numerous light field processing applications [43], [67], [80], [81]. We have summarised the use of FDL and encoding of low-rank approximated light field as BLOCK B in Figure. 1.

Our aim is to exploit and eliminate any intra-view, inter-view and other non-linear redundancies present among adjacent views in both horizontal and vertical directions of sub-aperture images, through various scanning patterns. This is followed by corresponding decoding and reconstruction of the light field subsets as illustrated in BLOCK C (Fig. 1). The Fourier Disparity Layer calibration, subset view synthesis, and prediction are described in the following subsections.

1) LIGHT FIELD VIEW SUBSETS

We divide the low-rank approximated light field into different view subsets based on four scanning orders. Hierarchical and circular view prediction orders are adopted as described in the work of Dib *et al.* [67]. The four chosen patterns are Circular-2 (C_2), Circular-4 (C_4), Hierarchical-2 (H_2) and Hierarchical-4 (H_4). For a 9×9 light field, the C_2 and H_2 patterns contain two view subsets and C_4 and H_4 patterns have four subsets. The exact coding orders of each subset of these four chosen scanning orders are shown in Fig. 2. In all subsets of these patterns, the light field views form a circle that spiral out from the center. Generally, the corner views of light fields are of lower quality, and thus we choose to form a circle rather than a square while scanning the views. We partition the approximated light field into subsets based on C_2, C_4, H_2 or H_4 patterns and proceed to work in the Fourier domain.

2) FDL CALIBRATION

The four-dimensional light field $L(u, v, s, t)$ is parameterized by angular coordinates (u, v) and spatial coordinates (s, t) [1], [2]. Without loss of generality, we have considered one spatial coordinate s and one angular coordinate u of approximated light field to present the notations in a simple manner. The approximated light field view L_{u_o} at angular position u_o can be defined as $L_{u_o}(s) = L(s, u_o)$. Construction of the FDL requires the angular coordinates u_j of the input approximated light field views and the disparity values d_k of the layers.

We can obtain Fourier coefficients of the j^{th} input light field view using n disparity values $\{d_k\}_{k\in[\![1,n]\!]}$ [67]. The Fourier transform of L_{u_o} at spatial frequency f_s is

$$\hat{L}_{u_o}(f_s) = \sum_k e^{+2i\pi u_o d_k f_s} \hat{L}^k(f_s). \tag{13}$$

The Fourier transform of the central light field view obtained by only considering a specific spatial region of disparity d_k is given by each $\hat{L}^k$.

By computing the Fourier transforms of all m approximated light field views as $\hat{L}_{u_j}(j \in [1, m])$, the FDL representation can be learnt by solving a linear regression problem for each frequency f_s. The problem is formulated by $\mathbf{Ax} = \mathbf{b}$ with Tikhonov regularization where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{x} \in \mathbb{R}^{n\times 1}$ and $\mathbf{b} \in \mathbb{R}^{m\times 1}$. Elements of matrix $\mathbf{A}$ are $\mathbf{A}_{jk} = e^{+2i\pi u_j d_k f_s}$, $\mathbf{x}$ contains

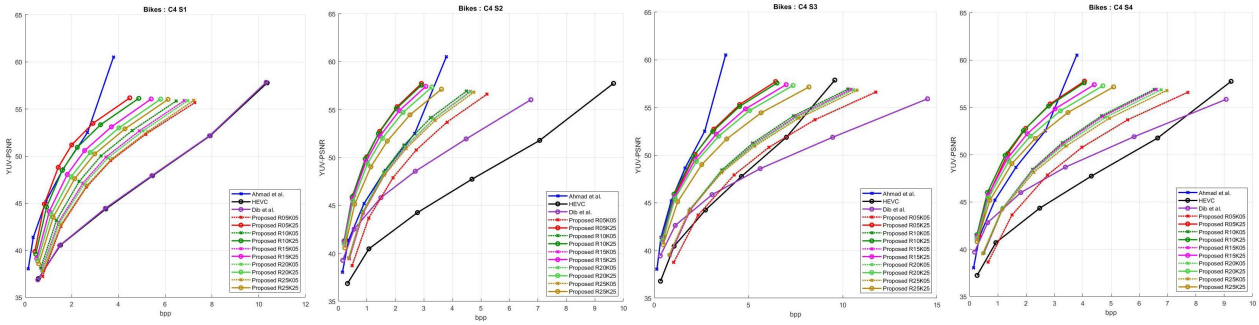


FIGURE 4. Rate-distortion graphs for the proposed compression scheme, Ahmad *et al.* [40], Dib *et al.* [67] and HEVC [48] for *Bikes* in the C_4 pattern.

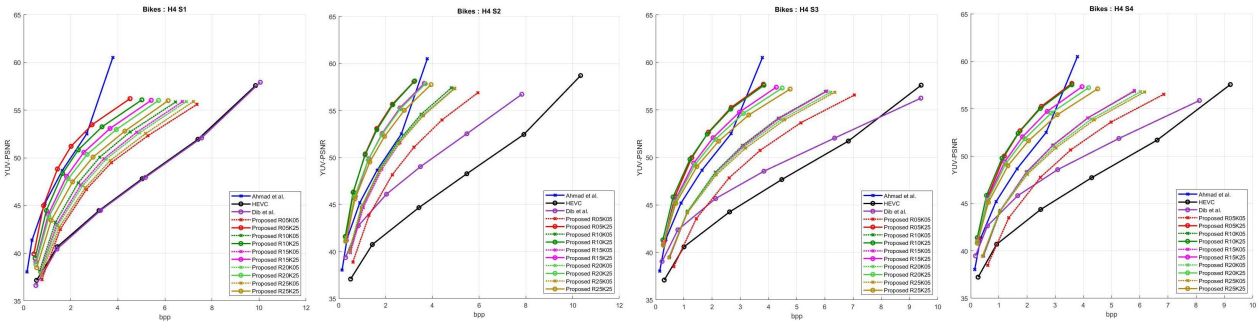


FIGURE 5. Rate-distortion plots for the proposed compression scheme, Ahmad *et al.* [40], Dib *et al.* [67] and HEVC [48] for *Bikes* in the H_4 patterns.

TABLE 2. The total number of kilobytes (kb) written for each subset of the Circular-2 pattern using our proposed coding scheme, Dib *et al.* and HEVC.

	QP 2		QP 14		QP 26	
	S1	S2	S1	S2	S1	S2
HEVC	7578	17806	3643	8405	812	1813
Dib et al.	7389	10595	3493	3745	764	630
RANK5 K5	5486	8909	2812	4557	1161	1686
RANK5 K10	4085	6083	2011	3024	826	1104
RANK5 K15	3617	5232	1728	2573	709	928
RANK5 K20	3432	5115	1624	2479	668	880
RANK5 K25	3290	4779	1545	2316	639	820
RANK10 K5	4917	7829	2490	3920	1023	1385
RANK10 K10	4088	5994	2004	2887	822	978
RANK10 K15	3830	5296	1849	2504	753	828
RANK10 K20	3672	4894	1759	2291	710	744
RANK10 K25	3591	4664	1712	2154	686	685
RANK15 K5	5019	7845	2555	3884	1046	1329
RANK15 K10	4383	6214	2164	2949	876	965
RANK15 K15	4166	5651	2027	2642	807	844
RANK15 K20	4043	5407	1948	2490	766	777
RANK15 K25	3979	5152	1907	2355	748	725
RANK20 K5	5228	8014	2675	3916	1084	1301
RANK20 K10	4627	6592	2294	3092	909	976
RANK20 K15	4426	6025	2152	2771	833	849
RANK20 K20	4335	5748	2080	2613	794	785
RANK20 K25	4260	5532	2013	2498	757	742
RANK25 K5	5400	8363	2770	4067	1111	1325
RANK25 K10	4869	6971	2421	3247	940	1005
RANK25 K15	4691	6435	2274	2938	866	879
RANK25 K20	4590	6034	2208	2713	833	791
RANK25 K25	4543	5888	2163	2635	811	761

the Fourier coefficients of disparity layers $\mathbf{x}_k = \hat{L}^k(f_s)$ and Fourier coefficients of the j^{th} input view, $\mathbf{b}_j = \hat{L}_{u_j}(f_s)$ are contained in $\mathbf{b}$.

3) FDL VIEW SYNTHESIS AND PREDICTION

In section III-B1, four scanning orders in circular and hierarchical patterns are discussed for the synthesis and coding of

TABLE 3. The total number of kilobytes (kb) written for each subset of the Hierarchical-2 pattern using our proposed coding scheme, Dib *et al.* and HEVC.

	QP 2		QP 14		QP 26	
	S1	S2	S1	S2	S1	S2
HEVC	7825	17090	3738	7963	832	1705
Dib et al.	7905	9829	3779	3278	816	522
RANK5 K5	5782	8796	2974	4495	1235	1655
RANK5 K10	4451	6042	2158	3001	885	1095
RANK5 K15	3897	5295	1834	2586	755	926
RANK5 K20	3680	4626	1700	2257	701	806
RANK5 K25	3526	4459	1628	2161	672	764
RANK10 K5	5194	7391	2617	3669	1079	1284
RANK10 K10	4389	5703	2130	2736	876	925
RANK10 K15	4088	5064	1949	2392	797	787
RANK10 K20	3957	4729	1874	2210	760	716
RANK10 K25	3870	4418	1819	2047	727	652
RANK15 K5	5378	7392	2729	3641	1119	1246
RANK15 K10	4663	5957	2286	2825	929	925
RANK15 K15	4435	5430	2141	2525	853	803
RANK15 K20	4337	5171	2075	2376	818	744
RANK15 K25	4265	4845	2016	2207	787	681
RANK20 K5	5580	7688	2838	3739	1156	1238
RANK20 K10	4987	6337	2461	2961	976	933
RANK20 K15	4758	5776	2313	2650	902	813
RANK20 K20	4646	5413	2218	2449	850	735
RANK20 K25	4587	5179	2167	2321	821	688
RANK25 K5	5776	7904	2961	3808	1195	1229
RANK25 K10	5207	6559	2583	3037	1009	934
RANK25 K15	5021	6022	2438	2727	934	814
RANK25 K20	4923	5776	2366	2593	896	270
RANK25 K25	4880	5619	2323	2511	873	731

light field views. Each of these chosen orders has two or four view subsets. This results in four patterns C_2, C_4, H_2 and H_4. In all four cases of the view prediction orders, the images are arranged in a spiral order starting from the center of the light field for each subset. The initial view subset of every pattern is always the first subset as specified by the scanning

TABLE 4. The total number of kilobytes (kb) written for each subset of the Circular-4 pattern using our proposed coding scheme, Dib *et al.* and HEVC.

	QP 2				QP 10				QP 20			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
HEVC	1373	5117	3809	15004	723	2484	1839	6986	201	566	419	1506
Dib et al.	1367	3579	5766	14717	720	1429	2236	5497	196	292	443	1034
RANK5 K5	963	2764	4670	12517	486	1449	2416	6452	203	562	923	2430
RANK5 K10	754	2103	3439	8838	363	1072	1736	4429	150	401	648	1640
RANK5 K15	668	1820	3020	7720	305	924	1515	3823	125	350	566	1406
RANK5 K20	626	1630	2670	6834	280	801	1307	3352	115	290	474	1218
RANK5 K25	594	1544	2561	6591	265	766	1257	3219	110	282	458	1166
RANK10 K5	858	2380	4087	10596	431	1220	2076	5318	178	458	764	1916
RANK10 K10	735	1900	3230	8306	353	946	1593	4041	147	346	570	1417
RANK10 K15	695	1715	2902	7450	326	840	1406	3564	135	296	491	1222
RANK10 K20	673	1597	2703	6948	311	777	1301	3291	129	273	450	1113
RANK10 K25	645	1543	2589	6576	296	743	1233	3084	122	255	416	1025
RANK15 K5	902	2452	4152	10717	459	1257	2100	5339	190	464	761	1886
RANK15 K10	780	1980	3372	8673	379	979	1648	4174	156	349	576	1425
RANK15 K15	732	1778	3052	7901	351	862	1463	3733	144	299	498	1241
RANK15 K20	721	1684	2902	7508	344	811	1380	3506	141	279	465	1147
RANK15 K25	716	1625	2783	7154	338	774	1309	3310	137	259	431	1067
RANK20 K5	923	2454	4204	10987	472	1237	2096	5404	194	444	738	1853
RANK20 K10	833	2045	3477	9157	411	999	1671	4344	169	350	569	1429
RANK20 K15	792	1884	3190	8401	387	907	1511	3920	159	309	503	1258
RANK20 K20	776	1801	3032	7954	376	859	1421	3665	153	289	465	1157
RANK20 K25	769	1737	2927	7645	369	819	1359	3488	148	271	439	1088
RANK25 K5	957	2514	4278	11321	493	1262	2118	5538	204	447	733	1857
RANK25 K10	862	2105	3592	9464	431	1021	1717	4452	175	352	575	1433
RANK25 K15	836	1967	3340	8766	413	942	1573	4054	168	317	514	1274
RANK25 K20	822	1890	3211	8436	405	898	1500	3865	164	300	486	1199
RANK25 K25	811	1915	3264	8264	396	909	1528	3770	160	302	494	1161

TABLE 5. The total number of kilobytes (kb) written for each subset of the Hierarchical-4 pattern using our proposed coding scheme, Dib *et al.* and HEVC.

	QP 2				QP 10				QP 20			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
HEVC	1305	1713	4990	17090	667	907	2370	7963	433	238	532	1705
Dib et al.	1332	1297	4985	15055	686	578	2034	5723	187	138	417	1098
RANK5 K5	975	985	3734	12714	495	532	1963	6586	206	214	762	2506
RANK5 K10	741	702	2667	8896	352	363	1341	4444	145	139	498	1645
RANK5 K15	662	580	2299	7769	305	298	1147	3836	126	117	423	1405
RANK5 K20	627	568	2102	6908	286	284	1030	3385	118	108	377	1229
RANK5 K25	599	533	2024	6657	267	269	991	3242	112	104	364	1173
RANK10 K5	854	797	3189	10768	428	419	1631	5415	178	166	600	1950
RANK10 K10	730	648	2551	8433	346	334	1270	4109	144	132	459	1443
RANK10 K15	686	590	2299	7558	320	301	1125	3619	133	117	398	1242
RANK10 K20	671	535	2109	7013	312	271	1015	3320	130	103	348	1117
RANK10 K25	665	538	2035	6654	307	270	968	3119	128	101	325	1038
RANK15 K5	894	818	3217	10780	452	433	1640	5395	186	172	593	1907
RANK15 K10	767	662	2617	8749	369	339	1282	4210	153	130	445	1430
RANK15 K15	744	611	2416	8020	356	310	1171	3804	147	118	402	1267
RANK15 K20	732	580	2309	7651	345	291	1111	3586	141	108	375	1173
RANK15 K25	718	606	2267	7326	339	307	1078	3409	139	117	361	1111
RANK20 K5	915	818	3289	11163	464	420	1653	5504	191	159	586	1888
RANK20 K10	822	686	2768	9275	404	349	1352	4417	166	131	462	1458
RANK20 K15	795	652	2577	8540	385	331	1244	4003	156	124	417	1294
RANK20 K20	774	619	2448	8049	370	310	1165	3725	149	114	382	1179
RANK20 K25	759	613	2376	7753	357	305	1122	3556	142	111	363	1113
RANK25 K5	955	822	3366	11444	487	425	1692	5611	199	160	591	1885
RANK25 K10	855	712	2874	9603	424	360	1395	4537	172	133	469	1466
RANK25 K15	827	678	2680	8890	404	340	1282	4128	161	124	420	1304
RANK25 K20	812	655	2584	8541	394	328	1227	3933	157	119	397	1225
RANK25 K25	813	654	2527	8353	390	324	1183	3807	155	115	377	1163

order. This subset is directly encoded using HEVC [48] first in BLOCK B. For example, in Fig. 2, the blue coloured subset in C_2 is the first subset.

The angular coordinates u_j and disparity values d_k are determined by the Fourier Disparity Layer calibration [43]. These are required in the further FDL construction and view predictions. This additional information is transmitted to the decoder in BLOCK C as metadata [67]. The initial view subset is used in the basic construction of FDL representation. This aids in the synthesis of succeeding view subsets. The residual signal is also encoded with HEVC to account for the remaining correlations in the prediction residue of synthesized views. The FDL representation is then refined before prediction and encoding of the next subset of views. Thus, the FDL representation is iteratively fine-tuned at every stage, after encoding every view subset, until all the approximated input light field views are encoded.

IV. RESULTS AND ANALYSIS

The performance of the proposed compression scheme is evaluated on real light fields captured by plenoptic cameras. The experiments are performed with *Bikes* light field dataset from the EPFL Lightfield JPEG Pleno database [82]. The raw plenoptic images are extracted into 9×9 sub-aperture views

TABLE 6. Bjontegaard percentage rate savings for the proposed compression scheme with respect to Dib *et al.* on *Bikes* data (negative values represent gains).

	C2		C4				H2		H4			
	S1	S2	S1	S2	S3	S4	S1	S2	S1	S2	S3	S4
RANK5 K5	-14	30	-33	3	9	22	-15	51	-29	-8	-2	19
RANK5 K10	-49	-30	-58	-39	-37	-33	-48	-18	-57	-50	-47	-36
RANK5 K15	-60	-49	-67	-52	-50	-48	-60	-36	-65	-64	-60	-52
RANK5 K20	-63	-53	-71	-63	-62	-59	-63	-52	-68	-66	-66	-60
RANK5 K25	-66	-58	-73	-66	-64	-62	-67	-55	-71	-69	-69	-63
RANK10 K5	-30	2	-46	-23	-16	-10	-32	10	-44	-37	-30	-13
RANK10 K10	-50	-39	-60	-51	-47	-44	-51	-31	-58	-57	-53	-46
RANK10 K15	-56	-53	-64	-60	-57	-55	-57	-47	-63	-63	-62	-56
RANK10 K20	-60	-60	-66	-65	-62	-61	-60	-55	-64	-69	-68	-63
RANK10 K25	-62	-65	-68	-67	-65	-65	-62	-60	-65	-70	-70	-66
RANK15 K5	-28	-1	-41	-20	-14	-10	-28	8	-40	-34	-29	-14
RANK15 K10	-45	-37	-55	-45	-44	-42	-46	-28	-55	-56	-54	-44
RANK15 K15	-50	-49	-60	-58	-54	-52	-51	-42	-57	-62	-60	-53
RANK15 K20	-54	-55	-61	-62	-59	-57	-54	-48	-59	-66	-64	-58
RANK15 K25	-55	-59	-62	-65	-62	-61	-56	-55	-60	-62	-65	-60
RANK20 K5	-24	0	-39	-23	-15	-9	-24	12	-38	-38	-29	-13
RANK20 K10	-40	-32	-50	-46	-43	-38	-40	-23	-49	-54	-50	-40
RANK20 K15	-46	-45	-54	-54	-52	-48	-45	-37	-53	-58	-56	-49
RANK20 K20	-49	-51	-56	-58	-56	-54	-49	-46	-55	-62	-61	-55
RANK20 K25	-51	-55	-58	-62	-60	-57	-51	-51	-57	-63	-63	-58
RANK25 K5	-20	6	-35	-20	-14	-7	-19	15	-34	-36	-27	-11
RANK25 K10	-36	-26	-47	-44	-40	-36	-36	-20	-46	-52	-47	-38
RANK25 K15	-41	-39	-50	-51	-48	-45	-41	-33	-49	-56	-54	-47
RANK25 K20	-44	-47	-52	-55	-52	-49	-44	-40	-51	-59	-57	-51
RANK25 K25	-46	-50	-53	-54	-51	-51	-45	-44	-51	-60	-60	-54
Average:	-45.56	-36.6	-55.16	-48	-44.6	-41.24	-45.76	-29	-53.52	-54.84	-52.12	-43.24

using the Matlab Light field toolbox [83]. Each view of *Bikes* has pixel dimensions of 434×625.

We first perform low-rank Tucker decomposition of the light field using tensor sketching as detailed in section III-A. The light field is approximated for various ranks R and sketch dimension parameters K. Next, patterns C_2, C_4, H_2 and H_4 are constructed from the approximated light field our experiments. Subsets 1 and 2 of C_2 contain 24 and 57 light field views respectively. The first and second subsets H_2 contain 25 and 56 views respectively. In C_4, subsets 1, 2, 3 and 4 have 4, 16, 12 and 49 views respectively. Lastly, subsets 1, 2, 3 and 4 of H_4 contain 4, 5, 16 and 56 light field views respectively. The exact scanning orders of the patterns and their subsets are specified in Fig. 2.

A. EXPERIMENTAL SETTINGS AND IMPLEMENTATION DETAILS

The proposed scheme is implemented on a single high-end HP OMEN X 15-DG0018TX system with 9th Gen i7-9750H, 16 GB RAM, RTX 2080 8 GB Graphics, and Windows 10 operating system. We applied Tucker decomposition on the input *Bikes* light field using tensor sketching [42]. The light field views are considered as large tensors and BLOCK A of the proposed scheme returns an approximate rank R Tucker decomposition of the images in the form of a core tensor and factor matrices. The algorithm also uses sketch parameter K, the dimension reduction parameter in *TensorSketch*. In our implementation, we experimented with ranks R and sketch K values to be 5, 10, 15, 20, 25. The low rank approximation was allowed to converge with a tolerence of 0.001. Next, the approximated light field are processed in the Fourier domain in four scanning patterns, C_2, C_4, H_2 and H_4. The corresponding low-rank approximated subsets are then utilized to form the FDL representation of light fields. The number of layers in the FDL method are fixed to $n = 30$. Views in approximated Subset 1 construct the initial FDL representation. The subsequent view subsets are predicted from this FDL representation and the residues iteratively refine the FDL representation. We used HEVC [48] (codec HM 16.0) to perform the encoding in BLOCK B, choosing quantization parameters 2, 6, 10, 14, 20 and 26.

B. RESULTS AND COMPARATIVE ANALYSIS

We compared performance of our proposed scheme with the light field coding method by Dib *et al.* [67], HEVC encoding of 9×9 *Bikes* light field subsets and compression scheme by Ahmad *et al.* [40]. In Ahmad *et al.*, the multi-view extension of HEVC (MV-HEVC) was utilized to encode the plenoptic image, considering it as a pseudo multi-view sequence. All the comparison methods were subject to same test conditions and quantization parameters, $QP = 2, 6, 10, 14, 20$ and 26. Each of the four scanning patterns, Circular-2, Circular-4, Hierarchical-2 and Hierarchical-4 was run for different combination of rank and sketch dimension values.

The total number of kilobytes (kb) written to file in for each pattern configuration with all combinations of rank, sketch dimension and QPs 2,14 and 26 in comparison to Dib *et al.* [67] and HEVC are shown in Tables 2-5. The bpp and PSNR graphs for the proposed scheme and anchor codecs including Ahmad *et al.* [40] are shown in Fig 3-5. Further, we performed an objective assessment using the Bjontegaard [84] metric. This metric can compare performance of two coding techniques where the average percentage difference in rate change is estimated over the range of six QP values. Percentage bit rate savings for proposed scheme with respect to Dib *et al.* for different layer configurations are summarized in Table 6 and results with respect to HEVC are depicted in Table 7. These results depict the better compression performance of proposed scheme with significant bitrate reduction and PSNR gains.

TABLE 7. Bjontegaard percentage rate savings for the proposed compression scheme with respect to HEVC on *Bikes* data (negative values represent gains).

	C2		C4				H2		H4			
	S1	S2	S1	S2	S3	S4	S1	S2	S1	S2	S3	S4
RANK5 K5	-19	-59	-33	-53	-55	-52	-15	-56	-27	-52	-48	-50
RANK5 K10	-52	-78	-58	-71	-73	-73	-48	-77	-56	-73	-71	-73
RANK5 K15	-62	-83	-67	-77	-78	-78	-60	-81	-64	-79	-78	-78
RANK5 K20	-65	-84	-71	-81	-82	-82	-62	-85	-68	-81	-80	-82
RANK5 K25	-68	-85	-73	-82	-83	-83	-66	-86	-71	-82	-82	-83
RANK10 K5	-34	-68	-46	-65	-65	-65	-32	-69	-43	-66	-63	-64
RANK10 K10	-53	-80	-60	-76	-77	-77	-50	-80	-58	-75	-74	-76
RANK10 K15	-59	-84	-64	-80	-81	-81	-57	-84	-62	-79	-78	-80
RANK10 K20	-62	-86	-66	-82	-83	-83	-59	-86	-64	-82	-81	-83
RANK10 K25	-64	-87	-69	-83	-84	-84	-61	-87	-64	-82	-83	-84
RANK15 K5	-32	-69	-42	-63	-65	-65	-28	-70	-39	-64	-62	-64
RANK15 K10	-48	-79	-56	-74	-76	-76	-45	-79	-54	-75	-74	-76
RANK15 K15	-53	-83	-61	-79	-79	-80	-51	-83	-56	-78	-77	-79
RANK15 K20	-56	-84	-62	-81	-81	-82	-53	-84	-58	-80	-79	-81
RANK15 K25	-57	-85	-63	-82	-82	-83	-55	-86	-59	-78	-80	-82
RANK20 K5	-28	-69	-40	-64	-65	-65	-24	-69	-36	-66	-62	-64
RANK20 K10	-44	-78	-51	-74	-75	-75	-39	-78	-48	-74	-72	-74
RANK20 K15	-49	-81	-55	-78	-78	-78	-45	-81	-52	-76	-75	-77
RANK20 K20	-52	-83	-57	-79	-80	-80	-49	-83	-54	-78	-78	-80
RANK20 K25	-54	-84	-58	-81	-82	-82	-50	-85	-56	-79	-79	-81
RANK25 K5	-25	-67	-36	-63	-65	-64	-19	-68	-32	-66	-61	-63
RANK25 K10	-39	-77	-47	-73	-74	-74	-35	-77	-44	-73	-71	-73
RANK25 K15	-45	-80	-51	-76	-77	-77	-41	-81	-48	-75	-74	-77
RANK25 K20	-47	-82	-52	-78	-79	-79	-44	-82	-50	-77	-76	-78
RANK25 K25	-49	-83	-53	-78	-78	-80	-45	-83	-50	-77	-77	-79
Average:	-48.64	-79.12	-55.64	-74.92	-75.88	-75.92	-45.32	-79.2	-52.52	-74.68	-73.4	-75.24

Among the four patterns chosen in the proposed scheme, it can be observed that the circular scanning patterns have better coding gains than the hierarchical ones. This is because intrinsic redundancies are better exploited in subsets with connected views as in Circular-2 and Circular-4, rather than scattered subset views of Hierarchical-2 and Hierarchical-4. Also, there is a notable improvement in results of orders with just two subsets (C_2 and H_2) over orders with four subsets (C_4 and H_4). Yet, C_4 and H_4 scanning patterns provide additional levels of scalability, which is preferable in some practical scenarios. Overall, our approach achieves scalable light field representation, coding and streaming, with a better performance against state-of-the-art light field coding schemes for all the four discussed scanning patterns.

V. CONCLUSION

In this paper, we have proposed a novel representation, coding and streaming scheme for light fields based on a Tucker *TensorSketch* decomposition and Fourier disparity layers algorithm. An input light field taken in four scanning patterns Circular-2, Circular-4, Hierarchical-2 and Hierarchical-4, is efficiently low-rank represented using Tucker decomposition as core and factor matrices using tensor sketching for many ranks and sketch dimension parameters. This operation happens in a single pass without storing the data in full and also allows streaming. The approximated light field is further processed in the Fourier domain to handle intra-view, inter-view and other non-linear redundancies present among adjacent views in both horizontal and vertical directions of sub-aperture images. The subsets in each scanning pattern hierarchically construct the FDL representation and synthesize subsequent views. Iterative refinement and encoding with HEVC are followed by the final light field reconstruction.

The complete processing pipeline operates as a single integrated system and can flexibly work for a range of multiple bitrates by varying the ranks and sketch dimension parameters. Our implementation has reduced memory requirements because the entire tensor light field data need not be stored in full. Compression performance of the proposed scheme was analyzed on real light fields in the four chosen patterns and substantial bitrate savings compared to state-of-the-art codecs are achieved, while maintaining good reconstruction quality. We accomplish superior results in terms of the size of the encoded bitstream file as well. There are also large PSNR gains obtained in our hybrid model compared to all anchor codecs. Other implications of the proposed hybrid representation and coding model make it quite versatile. It is a scheme that can permit view interpolation or extrapolation and generalized rendering by shifting approximated sub-aperture images in the depth dimension instead of two angular dimensions. The combined Tucker decomposition with sketching and FDL representation allows for streaming, filters the noise and corrects color inconsistencies between reconstructed views. The proposed hybrid representation and coding scheme is also suitable for GPU parallelization since it is built upon simple linear algebra operations carried out independently in the spatial and frequency domains.

There are several possibilities for extension and improvement of the proposed scheme. A Lambertian scene assumption with no occluded objects is made in the present model. In future, we look to develop a more relaxed algorithm that can handle non-Lambertian inputs with occlusions. Also, operations on sparse light fields (with wide baselines) obtained from different acquisition setups must be performed carefully. We aim to build on insights conveyed from the current approach and address this challenge. Another potential direction for future work is the rank-analysis of light fields for

streaming applications using learning-based methods. Further, we would also like to implement the proposed scheme on physical light field display hardware.

REFERENCES

[1] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 43–54.

[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 31–42.

[3] P. Surman and X. W. Sun, "Towards the reality of 3D imaging and display," in *Proc. 3DTV-Conf., True Vis., Capture, Transmiss. Display 3D Video (3DTV-CON)*, Jul. 2014, pp. 1–4.

[4] G. Wetzstein, D. R. Lanman, M. W. Hirsch, and R. Raskar, "Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting," MIT Media Lab, Cambridge, MA, USA, Tech. Rep., 2012. [Online]. Available: https://web.media.mit.edu/ gordonw/TensorDisplays/TensorDisplays.pdf

[5] P. Surman, X. Zhang, W. Song, X. Xia, S. Wang, and Y. Zheng, "Glasses-free 3-D and augmented reality display advances: From theory to implementation," *IEEE Multimedia Mag.*, vol. 27, no. 1, pp. 17–26, Jan. 2020.

[6] H. Isono, M. Yasuda, and H. Sasazawa, "Autostereoscopic 3-D display using LCD-generated parallax barrier," *Electron. Commun. Jpn. II, Electron.*, vol. 76, no. 7, pp. 77–84, 1993.

[7] K. Sakamoto and T. Morii, "Multi-view 3D display using parallax barrier combined with polarizer," in *Proc. SPIE*, vol. 6399, Oct. 2006, Art. no. 63990R.

[8] T. Peterka, R. L. Kooima, D. J. Sandin, A. Johnson, J. Leigh, and T. A. DeFanti, "Advances in the dynallax solid-state dynamic parallax barrier autostereoscopic visualization display system," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 3, pp. 487–499, May 2008.

[9] R. Börner, "Autostereoscopic 3D-imaging by front and rear projection and on flat panel displays," *Displays*, vol. 14, no. 1, pp. 39–46, Jan. 1993.

[10] J. Araii, F. Okano, M. Kawakita, M. Okui, Y. Haino, M. Yoshimura, M. Furuya, and M. Sato, "Integral three-dimensional television using a 33-megapixel imaging system," *J. Display Technol.*, vol. 6, no. 10, pp. 422–430, Oct. 2010.

[11] K. Takahashi, T. Saito, M. P. Tehrani, and T. Fujii, "Rank analysis of a light field for dual-layer 3D displays," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4634–4638.

[12] T. Saito, Y. Kobayashi, K. Takahashi, and T. Fujii, "Displaying real-world light fields with stacked multiplicative layers: Requirement and data conversion for input multiview images," *J. Display Technol.*, vol. 12, no. 11, pp. 1290–1300, Nov. 2016.

[13] Y. Kobayashi, K. Takahashi, and T. Fujii, "From focal stacks to tensor display: A method for light field visualization without multi-view images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2007–2011.

[14] Y. Kobayashi, S. Kondo, K. Takahashi, and T. Fujii, "A 3-D display pipeline: Capture, factorize, and display the light field of a real 3-D scene," *ITE Trans. Media Technol. Appl.*, vol. 5, no. 3, pp. 88–95, 2017.

[15] K. Takahashi, Y. Kobayashi, and T. Fujii, "From focal stack to tensor light-field display," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4571–4584, Sep. 2018.

[16] K. Maruyama, K. Takahashi, and T. Fujii, "Comparison of layer operations and optimization methods for light field display," *IEEE Access*, vol. 8, pp. 38767–38775, 2020.

[17] H. Watanabe, N. Okaichi, T. Omura, M. Kano, H. Sasaki, and M. Kawakita, "Aktina vision: Full-parallax three-dimensional display with 100 million light rays," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.

[18] T. Li, Q. Huang, S. Alfaro, A. Supikov, J. Ratcliff, G. Grover, and R. Azuma, "Light-field displays: A view-dependent approach," in *Proc. ACM SIGGRAPH ET*, 2020, pp. 1–2.

[19] M. Sharma, S. Chaudhury, and B. Lall, "A novel hybrid kinect-variety-based high-quality multiview rendering scheme for glass-free 3D displays," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2098–2117, Oct. 2017.

[20] M. Sharma, S. Chaudhury, B. Lall, and M. S. Venkatesh, "A flexible architecture for multi-view 3DTV based on uncalibrated cameras," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 599–621, 2014.

[21] M. Sharma, "Uncalibrated camera based content generation for 3D multi-view displays," Indian Inst. Technol. Delhi, New Delhi, India, Tech. Rep. TH-5223, 2017. [Online]. Available: http://eprint.iitd.ac.in/bitstream/handle/12345678/7300/TH-5223.pdf?sequence=2

[22] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *Proc. IEEE ICMEW*, Jul. 2016, pp. 1–4.

[23] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 539–543.

[24] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.

[25] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.

[26] R. J. S. Monteiro, P. J. L. Nunes, N. M. M. Rodrigues, and S. M. M. Faria, "Light field image coding using high-order intrablock prediction," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1120–1131, Oct. 2017.

[27] D. Liu, P. An, R. Ma, W. Zhan, X. Huang, and A. A. Yahya, "Content-based light field image compression method with Gaussian process regression," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 846–859, Apr. 2020.

[28] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1132–1145, Oct. 2017.

[29] S. Zhao and Z. Chen, "Light field image coding via linear approximation prior," in *Proc. IEEE ICIP*, Sep. 2017, pp. 4562–4566.

[30] X. Hu, J. Shan, Y. Liu, L. Zhang, and S. Shirmohammadi, "An adaptive two-layer light field compression scheme using GNN-based reconstruction," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 2s, pp. 1–23, Apr. 2020.

[31] E. Dib, M. Le Pendu, X. Jiang, and C. Guillemot, "Local low rank approximation with a parametric disparity model for light field compression," *IEEE Trans. Image Process.*, vol. 29, pp. 9641–9653, 2020.

[32] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 133–147, Jan. 2018.

[33] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson, "Shearlet transform-based light field compression under low bitrates," *IEEE Trans. Image Process.*, vol. 29, pp. 4269–4280, 2020.

[34] Y. Chen, P. An, X. Huang, C. Yang, D. Liu, and Q. Wu, "Light field compression using global multiplane representation and two-step prediction," *IEEE Signal Process. Lett.*, vol. 27, pp. 1135–1139, 2020.

[35] X. Huang, P. An, F. Cao, D. Liu, and Q. Wu, "Light-field compression using a pair of steps and depth estimation," *Opt. Exp.*, vol. 27, no. 3, pp. 3557–3573, 2019.

[36] B. Hériard-Dubreuil, I. Viola, and T. Ebrahimi, "Light field compression using translation-assisted view estimation," in *Proc. PCS*, Nov. 2019, pp. 1–5.

[37] I. Schiopu and A. Munteanu, "Deep-learning-based macro-pixel synthesis and lossless coding of light field images," *APSIPA Trans. Signal Inf. Process.*, vol. 8, Jul. 2019, Art. no. e20.

[38] 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," *Inf. Sci.*, vol. 545, pp. 118–131, Feb. 2021.

[39] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting plenoptic images as multi-view sequences for improved compression," in *Proc. IEEE ICIP*, Sep. 2017, pp. 4557–4561.

[40] W. Ahmad, M. Ghafoor, S. A. Tariq, A. Hassan, M. Sjöström, and R. Olsson, "Computationally efficient light field image compression using a multiview HEVC framework," *IEEE Access*, vol. 7, pp. 143002–143014, 2019.

[41] J. Gu, B. Guo, and J. Wen, "High efficiency light field compression via virtual reference and hierarchical MV-HEVC," in *Proc. IEEE ICME*, Jul. 2019, pp. 344–349.

[42] O. A. Malik and S. Becker, "Low-rank tucker decomposition of large tensors using tensorsketch," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 10096–10106.

[43] M. Le Pendu, C. Guillemot, and A. Smolic, "A Fourier disparity layer representation for light fields," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5740–5753, Nov. 2019.

[44] *JPEG Pleno Light Field Coding VM 1.1*, Standard ISO/IEC JTC 1/SC29/WD1 JPEG, 2018.

[45] C. Perra, P. G. Freitas, I. Seidel, and P. Schelkens, "An overview of the emerging JPEG Pleno standard, conformance testing and reference software," *Proc. SPIE*, vol. 11353, pp. 207–219, Apr. 2020.

[46] P. Schelkens, P. Astola, E. A. Da Silva, C. Pagliari, C. Perra, I. Tabus, and O. Watanabe, "JPEG Pleno light field coding technologies," *Proc. SPIE*, vol. 11137, Sep. 2019, Art. no. 111371G.

[47] C. Perra, P. Astola, E. A. D. Silva, H. Khanmohammad, C. Pagliari, P. Schelkens, and I. Tabus, "Performance analysis of JPEG Pleno light field coding," *Proc. SPIE*, vol. 11137, Sep. 2019, Art. no. 111371H.
[48] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[49] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-D hierarchical coding structure for light-field image compression," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1107–1119, Oct. 2017.
[50] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, "Efficient light field image coding with depth estimation and view synthesis," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1840–1844.
[51] X. Huang, P. An, L. Shan, R. Ma, and L. Shen, "View synthesis for light field coding using depth estimation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[52] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–10, Nov. 2016.
[53] X. Jiang, M. Le Pendu, and C. Guillemot, "Light field compression using depth image based view synthesis," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2017, pp. 19–24.
[54] N. Bakir, W. Hamidouche, O. Déforges, K. Samrouth, and M. Khalil, "Light field image compression based on convolutional neural networks and linear approximation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1128–1132.
[55] Z. Zhao, S. Wang, C. Jia, X. Zhang, S. Ma, and J. Yang, "Light field image compression based on deep learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[56] B. Wang, Q. Peng, E. Wang, K. Han, and W. Xiang, "Region-of-interest compression and view synthesis for light field video streaming," *IEEE Access*, vol. 7, pp. 41183–41192, 2019.
[57] C. Jia, X. Zhang, S. Wang, S. Wang, and S. Ma, "Light field image compression using generative adversarial network-based view synthesis," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 177–189, Mar. 2019.
[58] J. Ravishankar, M. Sharma, and P. Gopalakrishnan, "A flexible coding scheme based on block Krylov subspace approximation for light field displays with stacked multiplicative layers," *Sensors*, vol. 21, no. 13, p. 4574, Jul. 2021.
[59] J. Ravishankar and M. Sharma, "A hierarchical coding scheme for glasses-free 3D displays based on scalable hybrid layered representation of real-world light fields," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 3491–3498.
[60] J. Ravishankar and M. Sharma, "A novel hierarchical light field coding scheme based on hybrid stacked multiplicative layers and Fourier disparity layers for glasses-free 3D displays," 2021, *arXiv:2108.12399*.
[61] M. Hirsch, G. Wetzstein, and R. Raskar, "A compressive light field projection system," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, Jul. 2014.
[62] S. Lee, C. Jang, S. Moon, J. Cho, and B. Lee, "Additive light field displays: Realization of augmented reality with holographic optical elements," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–13, Jul. 2016.
[63] F.-C. Hung *et al.*, "The light field stereoscope: Immersive computer graphics via factored near-eye light field displays with focus cues," Tech. Rep., 2015, vol. 34. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=The+light+field+stereoscope%3A+immersive+computer+graphics+via+factored+near-eye+light+field+displays+with+focus+cues&btnG=
[64] F. Heide, D. Lanman, D. Reddy, J. Kautz, K. Pulli, and D. Luebke, "Cascaded displays: Spatiotemporal superresolution using offset pixel layers," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–11, 2014.
[65] K. Maruyama, H. Kojima, K. Takahashi, and T. Fujii, "Implementation of table-top light-field display," in *Proc. Int. Display Workshops*, 2018, pp. 1–6.
[66] J. Ravishankar, M. Sharma, and S. Khaidem, "A novel compression scheme based on hybrid tucker-vector quantization via tensor sketching for dynamic light fields acquired through coded aperture camera," in *Proc. Int. Conf. 3D Immersion (IC3D)*, Dec. 2021, pp. 1–8.
[67] E. Dib, M. L. Pendu, and C. Guillemot, "Light field compression using Fourier disparity layers," in *Proc. IEEE ICIP*, Sep. 2019, pp. 3751–3755.
[68] C. Battaglino, G. Ballard, and T. G. Kolda, "A practical randomized CP tensor decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 2, pp. 876–901, Jan. 2018.
[69] Y. Wang, H.-Y. Tung, A. Smola, and A. Anandkumar, "Fast and guaranteed tensor decomposition via sketching," 2015, *arXiv:1506.04448*.
[70] C. E. Tsourakakis, "MACH: Fast randomized tensor decompositions," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, Apr. 2010, pp. 689–700.
[71] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 363–372.
[72] J. Li, Y. Ma, C. Yan, and R. Vuduc, "Optimizing sparse tensor times matrix on multi-core and many-core architectures," in *Proc. 6th Workshop Irregular Appl., Architecture Algorithms (IA3)*, Nov. 2016, pp. 26–33.
[73] B. Liu, C. Wen, A. D. Sarwate, and M. M. Dehnavi, "A unified optimization approach for sparse tensor operations on GPUs," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2017, pp. 47–57.
[74] J. Oh, K. Shin, E. E. Papalexakis, C. Faloutsos, and H. Yu, "S-HOT: Scalable high-order tucker decomposition," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 761–770.
[75] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.
[76] H. Avron, H. L. Nguyen, and D. P. Woodruff, "Subspace embeddings for the polynomial kernel," in *Proc. NIPS*, vol. 1, 2014, p. 4.
[77] H. Diao, Z. Song, W. Sun, and D. Woodruff, "Sketching for Kronecker product regression and p-splines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1299–1308.
[78] R. Pagh, "Compressed matrix multiplication," *ACM Trans. Comput. Theory*, vol. 5, no. 3, pp. 1–17, 2013.
[79] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 2002, pp. 693–703.
[80] M. L. Pendu, C. Ozcinar, and A. Smolic, "Hierarchical Fourier disparity layer transmission for light field streaming," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 2606–2610.
[81] M. Le Pendu and A. Smolic, "High resolution light field recovery with Fourier disparity layer completion, demosaicing, and super-resolution," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, Apr. 2020, pp. 1–12.
[82] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. QoMEX*, 2016.
[83] D. G. Dansereau, O. Pizarro, S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1027–1034.
[84] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, 2001.

JOSHITHA RAVISHANKAR received the B.Tech. degree in electronics and communication engineering from the Vellore Institute of Technology, India, in 2018. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Indian Institute of Technology Madras, India. Her research interests include light field technology, 3D displays, machine learning for computer vision, computational photography, and image signal processing.

MANSI SHARMA received the M.Sc. degree in applied mathematics and the M.Tech. degree in computer applications from the Department of Mathematics, Indian Institute of Technology Delhi (IIT Delhi), in 2008 and 2010, respectively, and the Ph.D. degree in electrical engineering from the IIT Delhi, in 2017. Since May 2018, she has been working as an INSPIRE Faculty with the Department of Electrical Engineering, Indian Institute of Technology Madras, India. Her research interests include computational photography; computational imaging; deep learning; VR/AR/light field 3D displays; and tensor methods in computer vision, image, video, and multi-dimensional signal processing. She has been awarded the prestigious INSPIRE Faculty Award by the Indian National Science Academy, in 2017.