A FLEXIBLE CODING SCHEME FOR LIGHT FIELD DISPLAYS WITH STACKED MULTIPLICATIVE LAYERS

A dual degree project report

submitted by

G PRADEEP

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

21 June 2021

THESIS CERTIFICATE

This is to certify that the thesis entitled **A flexible coding scheme for light field displays with stacked multiplicative layers**, submitted by **G Pradeep**, to the Indian Institute of Technology Madras, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

MSharma

Dr. Mansi Sharma
Research Guide
INSPIRE FACULTY
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 21 June 2021

ACKNOWLEDGEMENTS

This project would not have been possible without the support of several people. I would like to express my sincere gratitude to my adviser Dr. Mansi Sharma. She has guided me and provided valuable key insights throughout the project. I would also like to extend my gratitude to Joshitha R. for her immense support.

ABSTRACT

While generating 3D perception on glass-free displays, supporting continuous motion parallax, greater depths of field and wider fields of view are critical to ensure realism. Current advances on Layered or Tensor Light Field displays have made them suitable for such an application. Using only a few light attenuating transparent pixelized layer displays, a high-resolution 3D image can be displayed supporting multiple viewing directions simultaneously. This thesis presents a flexible scheme for efficient layer-based representation and compression of light fields. The proposed coding scheme is a multi-stage pipeline that systematically removes redundancies to provide an efficient compression. In addition, the scheme can realise multiple bitrates using a set of hand-tunable parameters, unlike existing light field compression schemes. Extensive experiments performed demonstrate that the proposed coding scheme achieves substantial bitrate savings compared to pseudo-sequence-based light field compression approaches and state-of-the-artHEVC codecs.

KEYWORDS: Light Field Compression; Layered 3D displays; Convolutional Neural Networks; Randomized Block Krylov Singular Value Decomposition

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **i**

ABSTRACT **ii**

LIST OF TABLES **v**

LIST OF FIGURES **vii**

ABBREVIATIONS **vii**

1 Introduction **1**

2 Related work **3**

3 Proposed Coding Scheme **5**

3.1 Views to Multiplicative Layers 5

3.2 Low Rank Representation of Layers using BK-SVD 7

3.3 HEVC Encoding of Rank Approximated Layers 10

3.4 Reconstruction 10

4 Experiments **12**

4.1 Experimental Details 12

4.2 Baselines 14

4.3 Implementation Setup 15

5 Results **16**

5.1 Performance of the Proposed Coding Scheme 16

5.2 Usage of CNN over Analytical Methods 19

5.3 Usage of multiplicative layers 20

6 Conclusions **22**

7 Future Work 23

Bibliography 27

LIST OF TABLES

5.1 The total number of bytes written to file during compression using our proposed scheme for selected chosen ranks. The values are in *MB*. 18

5.2 Bjontegaard percentage rate savings for the proposed compression scheme with respect to Ahmad *et al.* (2017) and HEVC codec (negative values represent gains) on *Bikes* data. 18

5.3 Comparison of computation speed in *BLOCK II* of proposed scheme for *ML(3)* and *AV(169)*. Experiment evaluated on *Bikes* light field for BK-SVD ranks 20 and 60. The values represent runtime in *s*. 20

5.4 Bjontegaard percentage rate savings for the proposed compression scheme with respect to Ahmad *et al.* (2017), HEVC and Liu *et al.* (2016) on *Bikes* data for *ML(3)* and*AV(169)* for ranks 20 and 60 (negative values represent gain over anchor). . . 20

LIST OF FIGURES

1.1 A tensor display designed for 3D projection 2

3.1 The left image shows the configuration of the display panels and the backlight; the right image describes the parametrization of a light ray 5

3.2 The proposed coding scheme pipeline 7

3.3 The BK-SVD algorithm and the necessary rearrangements of layers 8

3.4 HEVC Pipeline 10

4.1 The left most image shows all the SAIs of *Fountain-Vincent2*; images on the right show the central view of *Bikes*, *Fountain-Vincent2*, and *Stone-Pillars Outside* respectively. 12

4.2 Hyperparameter search results 13

4.3 The three multiplicative layers of *Bikes*, *Fountain-Vincent2*, and *Stone-Pillars Outside* light fields generated by the CNN. The columns indicate layer -1, 0, and 1 respectively. 13

4.4 Comparison of the original and reconstructed views. Column 1: Original central views of the three datasets; Column 2: Reconstructed central views - QP2, rank 20; Column 3: Reconstructed central views -QP 2, rank 60 14

5.1 The bitrate vs PSNR graphs of Ahmad *et al.* (2017) coding scheme for all three datasets. Each plot describes YPSNR, UPSNR and VPSNR respectively. 17

5.2 Rate-distortion curves for the proposed compression scheme and HEVC codec for the three datasets. Each row represents a different dataset. The columns indicate YPNSR, UPSNR and VPSNR respectively. . 17

5.3 Comparative mean SSIM for the proposed coding scheme and the baselines. 18

5.4 Computation time accuracy of reproduced light fields using analytical and CNN-based optimization of multiplicative layers. 19

5.5 View 19 of Bunnies reproduced using analytical method (ANA) and CNN respectively with corresponding difference images. ANA: PSNR:19.94 dB, SSIM:0.895; CNN: PSNR:22.18 dB, SSIM:0.918 19

5.6 Mean SSIM scores over each QP of decoded views in BLOCK III of proposed scheme using *ML(3)* and *AV(169)*. Experiment evaluated on Bikes dataset for BK-SVD ranks 20 and 60 20

ABBREVIATIONS

SAI	Sub-aperture image
HEVC	High Efficiency Video Coding
HLRA	Homography based Low-rank approximation
CNN	Convolutional Neural Network
GNN	Graph Neural Network
SVD	Singular Value Decomposition
BK-SVD	Block Krylov Singular Value Decomposition
GPU	Graphics Processing Unit
SSIM	Structural Similarity Index

CHAPTER 1

Introduction

Realistic 3D presentation on displays has been a long-standing challenge for researchers in the areas of plenoptics, light fields, and full parallax imaging Surman and Sun (2014); Li *et al.* (2020); Watanabe *et al.* (2019). Glasses-free or naked-eye autostereoscopic displays have replaced stereoscopic displays that offer motion parallax for different viewing directions Geng (2013). However, current naked-eye displays fall far short of truly recreating continuous motion parallax, greater depth-of-field, and a wider field-of-view for visual reality **?**Sharma *et al.* (2016, 2014); Sharma (2017).

Designs based on a single display panel attached with a parallax barrier or special lens (lenticular screen or integral photography lens) usually suffer from inherent resolution limitations. The resolution for each view decreases with an increase in multiple viewing directions. Thus, supporting a full parallax visualization of the 3D scene is impractical Geng (2013). On the other hand, both monitor-style and large-scale systems based on several projectors introduce a wide viewing approach but do not maintain a thin form factor and require ample space to set up. Besides, such large-scale systems require costly hardware and compute to reproduce high-quality views Hirsch *et al.* (2014); Balogh *et al.* (2007).

Multi-layered or tensor light field displays offer an optimized solution to support direction-dependent outputs simultaneously, without sacrificing the resolution in reproducing dense light fields Wetzstein *et al.* (2012); Takahashi *et al.* (2015); Saito *et al.* (2016); Kobayashi *et al.* (2017*b*); Maruyama *et al.* (2020); Kobayashi *et al.* (2017*a*); Takahashi *et al.* (2018); Maruyama *et al.* (2019). A typical structure of a layered 3D display is demonstrated in Figure 1.1. It is composed of a few light-attenuating pixelized layers stacked in front of a backlight. The transmittance of pixels on each layer can be controlled independently. A multi-layered display with multiplicative layers can be implemented with liquid crystal display(LCD) panels and a backlight and additive layers are fabricated with holographic optical elements (HOEs) and projectors Maruyama *et al.* (2020).

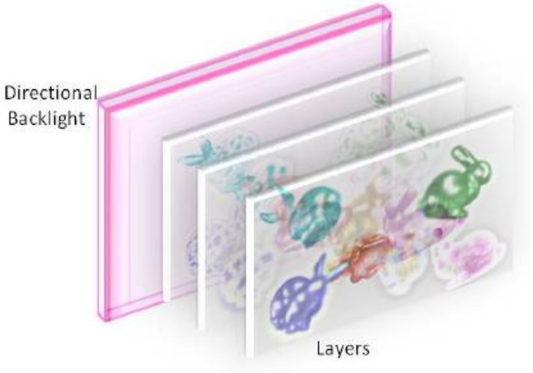

Figure 1.1: A tensor display designed for 3D projection

With this structure, the layer patterns allow light rays to pass through different combinations of pixels depending on the viewing directions. As shown in Figure 3.1, multiplicative layer patterns overlap with different shifts in observed directions, precisely reproducing multi-view images with a high resolution. Further, compactly representing the light field using only a few transmittance patterns offers display adaptation.

Thus, it is critical to analyze the intrinsic redundancy in light fields to generate an efficient 3D production and content delivery pipeline using multi-layer-based approaches. In this work, the problem of light field dimensionality reduction for existing multi-layer or tensor 3D displays is addressed. For the sake of convenience, only multiplicative layer displays are described and used. The corresponding coding scheme for additive layer displays can be built similarly.

CHAPTER 2

Related work

The majority of existing light field coding approaches are not directly applicable for multi-layered displays. Several coding approaches extract the SAIs and encode them as a pseudo video sequence Liu *et al.* (2016); Li *et al.* (2017); Ahmad *et al.* (2017, 2019); Gu *et al.* (2019); Sharma and Ragavan (2019). Existing video encoders like HEVC Sullivan *et al.* (2012) or MV-HEVC are used for inter- and intra-frame hybrid prediction. View estimation-based methods Senoh *et al.* (2018); Huang *et al.* (2018, 2019); Hériard-Dubreuil *et al.* (2019) reconstruct the entire light field from a small subset of encoded views. However, such algorithms fail to remove redundancies among adjacent SAIs and restrict prediction to the local or frame units of the encoder. Also, learning-based view-synthesis methods for light field compression Bakir *et al.* (2018); Zhao *et al.* (2018); Wang *et al.* (2019); Schiopu and Munteanu (2019); Liu *et al.* (2021); Jia *et al.* (2018) require large-scale and diverse training samples. To reconstruct high-quality views, a significant fraction of the SAIs have to be used as references.

Disparity-based methods

Algorithms that exploit low-rank structure in light field data follow disparity-based models Jiang *et al.* (2017); Dib *et al.* (2020). Jiang *et al.* (2017) proposed a HLRA method that aligns light field sub-aperture views by analyzing disparity across views from different depth planes. The HLRA may not optimally reduce the low-rank approximation error for light fields with large baselines. Geometry-based schemes have gained recent popularity for efficient compression at low-bit rates Vagharshakyan *et al.* (2017); Ahmad *et al.* (2020); Chen *et al.* (2020). Such schemes use light field structure/multi-view geometry and are not suitable for coding layer patterns directly.

Content-based methods

Methods that explicitly consider the content of light field data for compression Liu *et al.* (2019); Hu *et al.* (2020) also do not work in present settings. Liu *et al.* (2019) compress plenoptic images by classifying light field content into three categories based on texture homogeneity. Corresponding Gaussian process regression-based prediction methods are used for each category. The performance depends on scene complexity and sophisticated treatment is required to handle the boundaries of the lenslet image. Similarly, the GNN-based scheme presented by Hu *et al.* (2020) separates high-frequency and low-frequency components in sub-aperture images. This scheme needs an accurate parameter estimation model and discards specific frequency components permanently. None of these coding techniques are explicitly designed for layered light field displays. They also usually support only specific bitrates during the compression.

Proposed Coding Scheme

Differing from existing approaches, the proposed Block Krylov SVD based lossy compression scheme works for layered light-field displays with light-ray operations regulated using stacked multiplicative layers and a CNN. The CNN is employed to produce optimal multiplicative layers obtained from light fields. CNN-based methods are proven to be computationally more efficient than the previous analytical optimization approaches based on non-negative tensor factorization Wetzstein *et al.* (2012); Maruyama *et al.* (2020, 2019); Lee *et al.* (2016). The proposed algebraic representation of stacked multiplicative layers on the Krylov subspace approximates the hidden low-rank structure of the light field data. Factorization derived from BK-SVD efficiently exploits the high spatial correlation between multiplicative layers and approximates the light field with varying low ranks. Further encoding using the HEVC encoder eliminates inter-layer and intra-layer redundancies and considerably improves the compression efficiency. By choosing varying ranks and quantization parameters, the scheme allows variable performance considering the device bandwidth constraints. This allows the delivery of 3D content with limited hardware resources and best meets the viewers' preferences for depth immersion and visual comfort.

CHAPTER 3

Proposed Coding Scheme

The pipeline workflow for the proposed coding scheme is shown in Figure 3.2. BLOCK I and II belong at the transmitter end whereas BLOCK III belongs at the receiver or device end. BLOCK I represents a CNN that produces multiplicative layers from input light field SAIs. In BLOCK II, redundancies in the layers are eliminated by performing a low-rank approximation using BK-SVD. The approximated layers are then fed into the HEVC encoder that eliminates inter- and intra-frame redundancies. This compressed bitstream is then transmitted to the device. In BLOCK III, the compressed bitstream is decoded to reconstruct the layers and individual SAIs at the device end.

3.1 Views to Multiplicative Layers

In the first component, the proposed coding scheme generates multiplicative layer patterns from a given set of light field views. A light field can be defined as a 4-D function describing light rays travelling straight through free space Levoy and Hanrahan (1996); Gortler *et al.* (1996). In this work, a plane and angle representation is considered as shown in Figure 3.1. A reference frame $z = 0$ is defined. The intensity of a light ray passing through (u, v) on the plane making an angle (θ, ϕ) with the z-axis is described as $L(s, t, u, v)$, where $s = tan(\theta)$ and $t = tan(\phi)$.

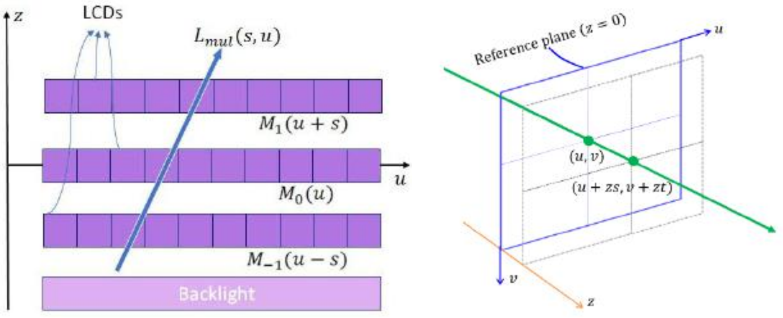

Figure 3.1: The left image shows the configuration of the display panels and the backlight; the right image describes the parametrization of a light ray

Multiplicative layers are light attenuating panels stacked at equal intervals from each other in front of a backlight. The setup is shown in Figure 3.1. A ray passing through the reference frame at (u, v) and at an angle coordinate (s, t) will intersect the frame at depth z at $(u + zs, v + zt)$. Thus the intensity of an emitted light ray from this setup, after normalizing by the intensity of the backlight, is given by

$$L_{mul}(s, t, u, v) = \prod_{z \in Z} M_z(u + zs, v + zt) \tag{1}$$

where $M_z(u, v)$ denotes the transmittance of the pixel (u, v) on a layer at depth z. Here, $Z = \{-1, 0, 1\}$ is assumed, where Z denotes the normalized depth instead of physical depth along the z- axis Maruyama *et al.* (2020).

Thus the optimization goal for layer patterns is given by

$$\underset{M_z | z \in Z}{\arg\min} \sum_{s,t,u,v} \| L(s, t, u, v) - L_{mul}(s, t, u, v) \|^2 \tag{2}$$

where $L(s, t, u, v)$ is the input light field. Instead of performing this optimization analytically, a CNN can be learnt to generate layer patterns from input SAIs. This can be mapped as

$$f : \mathbf{L} \rightarrow \mathbf{M} \tag{3}$$

where $\mathbf{M}$ represents a tensor containing all the layer patterns M_z.

Similarly, 1 can be considered as the inverse mapping from layer patterns to the light field $L_{mul}(s, t, u, v)$. It can be denoted as

$$f_{mul} : \mathbf{M} \rightarrow \mathbf{L_{mul}} \tag{4}$$

where $\mathbf{L_{mul}}$ contains all the light rays in $L_{mul}(s, t, u, v)$. During training, a CNN optimizes the loss function given by

$$\underset{\theta_f}{\arg\min} \|\mathbf{L} - \mathbf{L_{mul}}\|^{\mathbf{2}} \tag{5}$$

where θ_f denotes the weights of the CNN. Note that $\mathbf{L_{mul}}$=$f_{mul}(f(\mathbf{L}))$ is true from definition. $f(\mathbf{L})$ is performed by the CNN and f_{mul} reconstructs SAIs from the obtained layers.

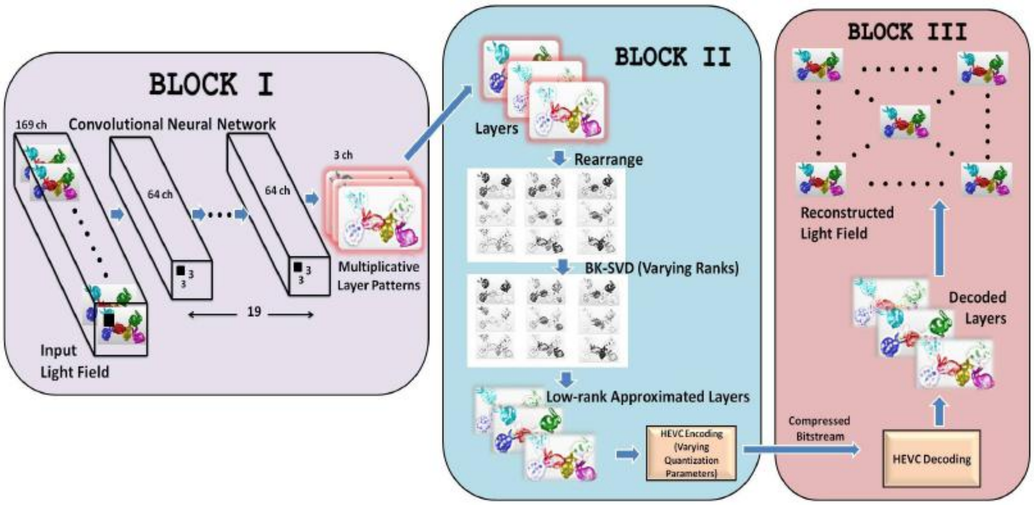

Figure 3.2: The proposed coding scheme pipeline

Earlier works performed this optimization analytically using algorithms based on non-negative tensor factorization. Analytical methods obtain the three multiplicative layers of an input light field by carrying out the optimization of one layer at a time individually Maruyama *et al.* (2020). The solutions are updated in an iterative manner, with there existing a definite trade-off between the number of iterations and accuracy of the obtained solution. On the other hand, learning a CNN proves to be better in terms of the balance between computation time and accuracy than the previous analytical optimization approaches. The merits of using a CNN over analytical methods has been discussed later in this report.

3.2 Low Rank Representation of Layers using BK-SVD

The key goal of the proposed scheme is to remove the intrinsic redundancy in light field data by analyzing the hidden low-rank structure of multiplicative layers. BLOCK II of the workflow involves this low-rank representation. The layers are represented compactly on a Krylov subspace and approximated using BK-SVD.

The three individual color channels from each of the three layers are stacked together. Let this new matrix be B^{ch} where $ch \in \{R, G, B\}$, hold the corresponding color channels from the three layer patterns obtained. This is necessary since the individual color channels have to be decomposed separately. For simplicity, B^{ch} will be denoted as B.

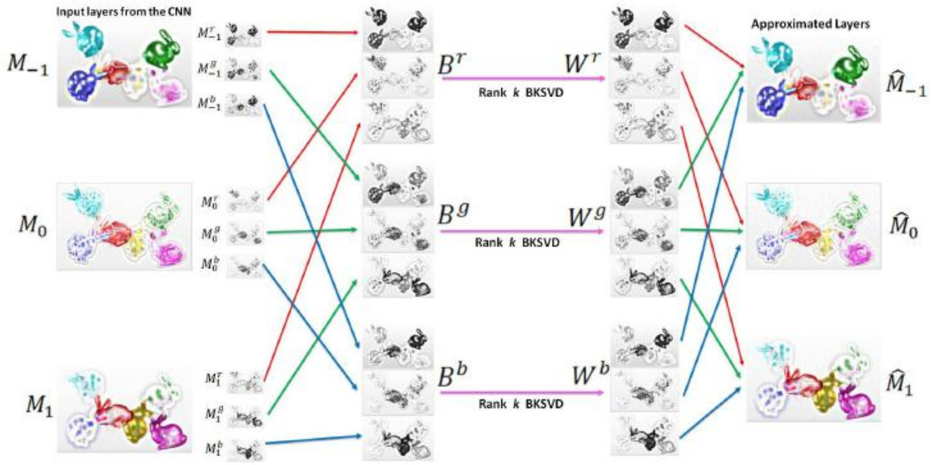

Figure 3.3: The BK-SVD algorithm and the necessary rearrangements of layers

Given a matrix $B \in \mathbb{R}^{c \times d}$ of rank r, SVD can be performed as $B = U\Sigma V^T$, where the left and right singular vectors of B are the orthonormal columns of $U \in \mathbb{R}^{c \times r}$and $V \in \mathbb{R}^{r \times d}$ respectively. $\Sigma \in \mathbb{R}^{r \times r}$ is a positive diagonal matrix containing $\sigma_1 \geq \ldots \geq \sigma_r$, the singular values of B. Conventional SVD algorithms are computationally expensive. Thus, there is substantial research done on randomized techniques to achieve optimal low-rank approximation Halko *et al.* (2011); Pedregosa *et al.* (2011); Musco and Musco (2015). The recent focus has shifted towards methods that inherently do not depend on the properties of the matrix or the gaps in its singular values.

Traditional Simultaneous Power Iteration algorithms for SVD initialized with random start vectors achieve the spectral norm error in nearly $\tilde{O}(\frac{1}{\epsilon})$ iterations. Block Krylov SVD algorithm presented in Musco and Musco (2015), a randomized variant of the Block Lanczos algorithm Cullum and Donath (1974); Golub and Underwood (1977), guarantees to achieve the same in just $\tilde{O}(\frac{1}{\sqrt{\epsilon}})$ iterations. This not only improves runtime for achieving spectral norm error but guarantees substantially better performance practically.

The intuition behind Block Krylov Iteration matches that of many accelerated iterative methods. In order to achieve spectral norm error, the iterative algorithm must reduce this noise down to the scale of $\sigma_{k+1} = \|B - B_k\|^2$. It does this by working with the powered matrix B^q. By the spectral theorem, B^q has exactly the same singular vectors as B, but its singular values are equal to the singular values of B raised to the q^{th} power. Powering spreads the values apart and B^q's lower singular values are relatively much smaller than its top singular values and effectively, the spectral value tail is denoised. For a random matrix $\Pi \sim N(0,1)^{d\times k}$, the Krylov subspace of matrix B is given by

$$K = [\Pi\ B\Pi\ B^2\Pi\ B^3\Pi ... B^q\Pi] \tag{6}$$

Computing B^q directly is costly, so $B^q\Pi$ is computed iteratively. The fact that there are better polynomials than B^q for denoising tail singular values is utilized. In particular, a lower degree polynomial can be used, allowing us to compute fewer powers of B and thus leading to an algorithm with fewer iterations.

Block Krylov Iteration takes advantage of such polynomials by working with the Krylov subspace. From the subspace, $p_q(B)\Pi$ can be constructed for any polynomial $p_q()$ of degree q. The very best k rank approximation to B lying in the span of K at least matches the approximation achieved by projecting onto the span of $p_q(B)\Pi$ Musco and Musco (2015).

Algorithm 1: Block Krylov Singular Value Decomposition

```
Input: $\mathbf{B} \in \mathbb{R}^{c\times d}$, error $\epsilon \in (0,1)$, rank $k \leq c, d$
Output: $\mathbf{W} \in \mathbb{R}^{c\times k}$
$q := \tilde{O}(\frac{\log d}{\sqrt{\epsilon}})$, $\Pi \sim N(0,1)^{d\times k}$
Compute $\mathbf{K} := \left[\mathbf{B}\Pi\ ,\ (\mathbf{B}\mathbf{B}^T)\mathbf{B}\Pi\ ,\ \cdots,\ (\mathbf{B}\mathbf{B}^T)^q\mathbf{B}\Pi\right]$
Orthonormalize the columns of $\mathbf{K}$ to obtain $\mathbf{Q} \in \mathbb{R}^{c\times qk}$
Compute $\mathbf{S} := \mathbf{Q}^T\mathbf{B}\mathbf{B}^T\mathbf{Q} \in \mathbb{R}^{qk\times qk}$
Set $\bar{\mathbf{U}}_k$ to the top $k$ singular vectors of $\mathbf{S}$.
return $\mathbf{W} = \mathbf{Q}\bar{\mathbf{U}}_k$
```

The BK-SVD algorithm is presented as Algorithm 1. The algorithm is run on the rearranged layers B^{ch}. The rank k approximated matrices are denoted as W^{ch}. The channels can be rearranged to obtain compressed RGB layers $\hat{M}_z$. These are sent to the HEVC encoder for further compression.

3.3 HEVC Encoding of Rank Approximated Layers

The High Efficiency Video Coding (HEVC) Sullivan *et al.* (2012) is the latest international standard for video compression. It was standardized by ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. HEVC achieves improved compression performance over its predecessors, with at least a 50 percent bit-rate reduction for the same perceptual quality Sullivan *et al.* (2012).

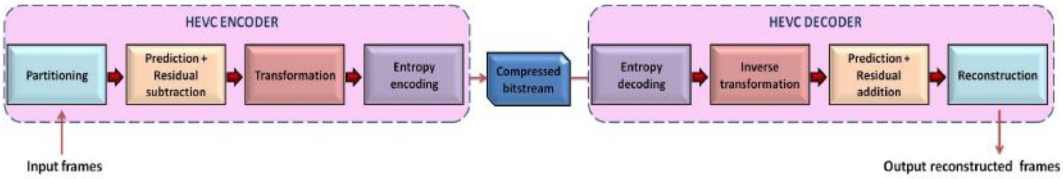

Figure 3.4: HEVC Pipeline

The encoding scheme of HEVC partitions each frame into smaller blocks. This information is relayed to the decoder for reconstruction. Barring the first frame which is coded using intra-frame prediction(spatial prediction from other blocks of the same frame), the rest of the frames are coded using intra- and inter-frame prediction based on one or more reference frames. Motion-compensated prediction is used to remove temporal redundancy across frames during inter-frame prediction. A linear spatial transformation is then applied to the resulting signal to produce the resulting transform coefficients. The quantization of transform coefficients is performed which is controlled by the Quantization Parameters (QP). QP ranges from 0 to 51 and for every 6 increase in QP, the quantizer step size doubles. Then, the quantized transform coefficients are entropy coded and transmitted along with the prediction information. The entire workflow pipeline is shown in Figure 3.4.

3.4 Reconstruction

At the receiving end, the compressed bitstream is decoded by the HEVC decoder. This produces multiplicative layers optimized for the device in specific. These layers can be displayed directly on the device to render a 3D image. The individual SAIs can be obtained by applying 1 i.e the mapping f_{mul} on the obtained layers.

The concept of "one network, multiple bitrates" allows the coding scheme to achieve the goal of covering a range of bitrates, leveraging the generality of low-rank models and data-driven CNNs for different types of display devices. This is enabled by the hand-tunable set of parameters during the encoding process: BK-SVD Ranks and HEVC QPs. This allows the user to flexibly adjust the visual quality in accordance with the bandwidth constraints. In addition, the proposed coding model not just supports multi-view/light field displays, but can also complement existing light-field coding schemes, which employ different networks to encode light field images at different bitrates. The experiments with real light field data demonstrate very competitive results.

CHAPTER 4

Experiments

4.1 Experimental Details

The performance of the proposed compression scheme was evaluated on real light fields captured by plenoptic cameras. From the EPFL Lightfield JPEG Pleno Database Pennebaker and Mitchell (1992), experiments were performed using Bikes, Fountain and Vincent 2, and Stone Pillars Outside light field samples. The raw images yielded 15x15 SAIs, each with a resolution of 434x625 pixels, using the Matlab Light Field Toolbox. Liu *et al.* (2016) and Ahmad *et al.* (2017) consider only the central 13×13 views of the light field as the pseudo sequence for compression. In this case, the border SAIs obtained suffer greatly from geometric distortion and a general loss of detail due to the lenslet structure in the camera. These SAIs are discarded to facilitate a fair comparison with the other baselines. Thus, only the central 13x13 SAIs were considered for all the experiments performed in this work. Figure 4.1 shows the views of the chosen light field images.

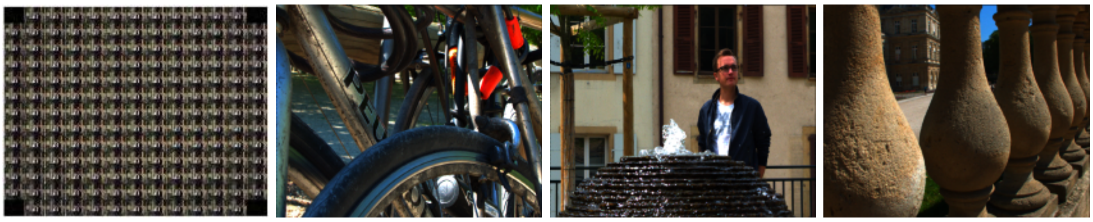

Figure 4.1: The left most image shows all the SAIs of *Fountain-Vincent2*; images on the right show the central view of *Bikes*, *Fountain-Vincent2*, and *Stone-Pillars Outside* respectively.

The network was trained with 30 training images generated from the light fields Friends 1, Poppies, University and Desktop, and Flowers from the EPFL Lightfield JPEG Pleno database [58]. Each training sample was a set of 169 64x64 size image patches extracted from the same pixel location. Various configurations of hyperparameters were analyzed before settling on a learning rate of 1e-4, a batch size of 15 and 20 epochs. The Adam optimizer was used with default parameters. Maruyama *et al.*

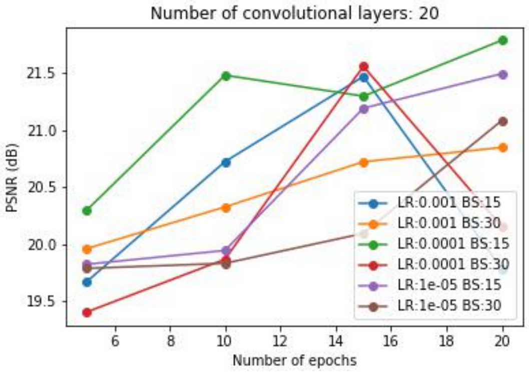

Figure 4.2: Hyperparameter search results

(2020) had demonstrated 3×3 as the optimal filter size for the network and used 64 channels throughout for all the 20 convolutional layers. These were followed in this work as well. The results of the hyperparameter search are presented in Figure 4.2. The resultant output layers for Bikes, Fountain and Vincent 2, and Stone Pillars Outside, obtained from the trained CNN model, are presented in Figure 4.3.

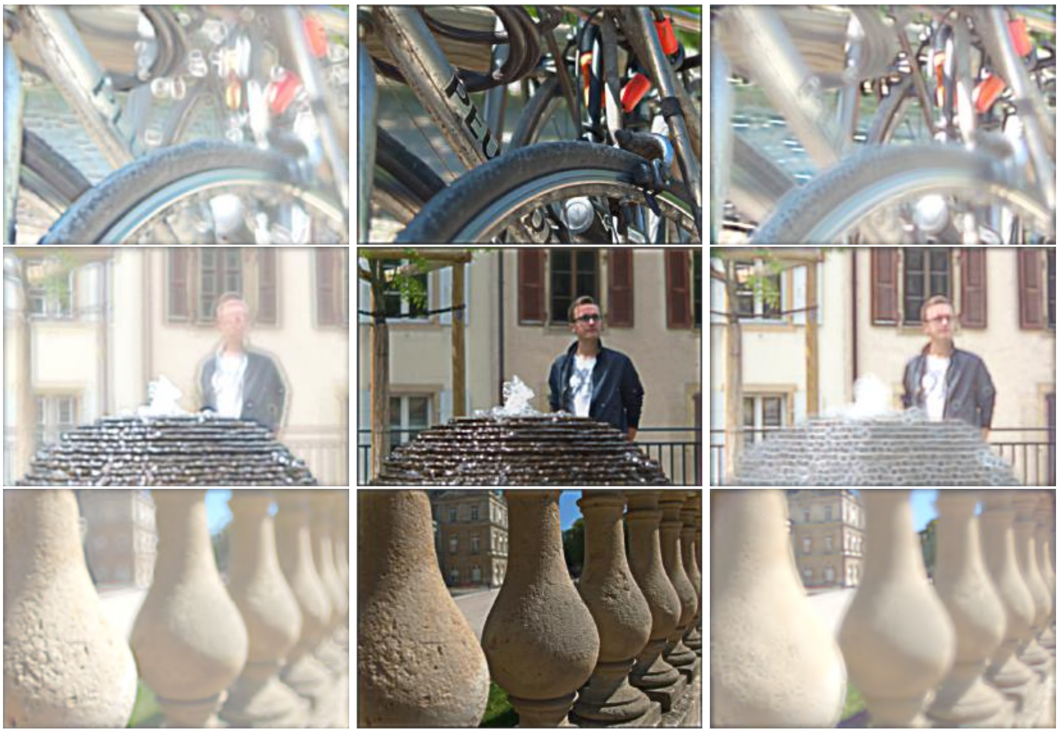

Figure 4.3: The three multiplicative layers of *Bikes*, *Fountain-Vincent2*, and *Stone-Pillars Outside* light fields generated by the CNN. The columns indicate layer -1, 0, and 1 respectively.

The described BK-SVD algorithm was then run on the resultant layers, for a set of ranks between 4 to 60 (15 in all, incremented in steps of four), with 50 iterations each. The approximated matrices were converted into the YUV420 color space for the HEVC encoder. The 32 bit HM encoder (version 11.0) is used throughout the work.

Seven quantization parameters are chosen: QP 2, 6, 10, 14, 20, 26, and 38, to analyze both high and low bitrates. The encoder compression produces a bitstream for a fixed rank and QP, that can be stored or transmitted.

The exact reverse procedure was followed on the compressed bitstream to reconstruct the 13×13 views of the light fields. The HM decoder produces three decoded layers in the YUV color space that are used to reconstruct all the SAIs back in the RGB space. A comparison of the original central view of the Bikes, Fountain and Vincent 2, and Stone Pillars Outside images and the reconstructed central view is shown in Figure 4.4.

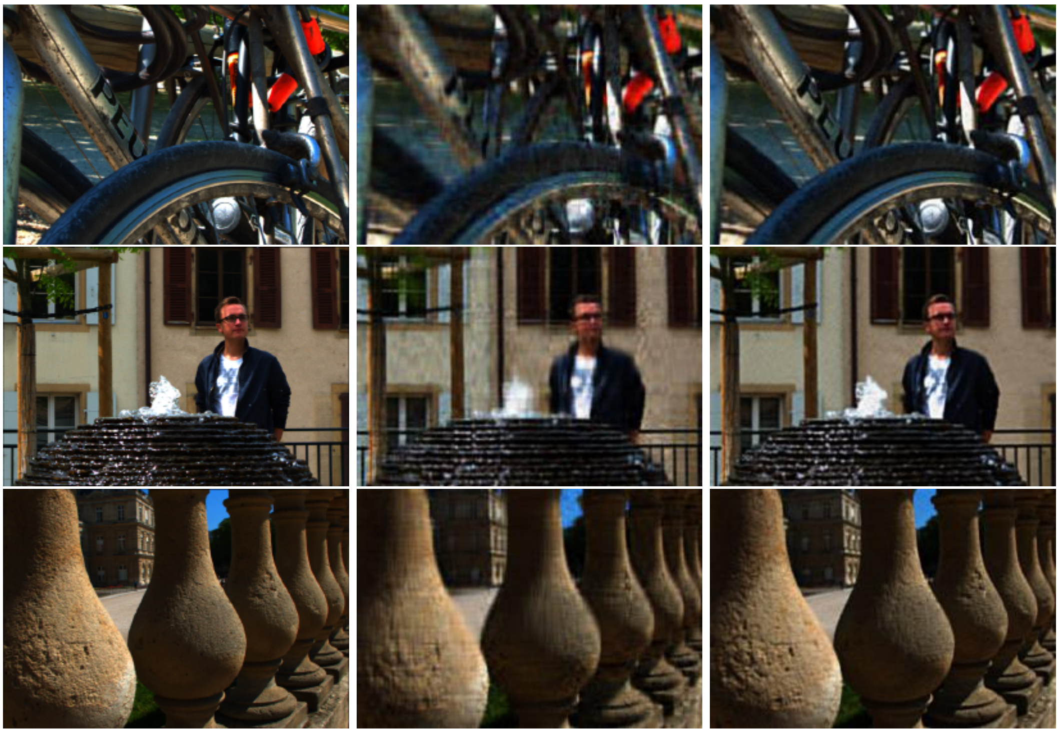

Figure 4.4: Comparison of the original and reconstructed views. Column 1: Original central views of the three datasets; Column 2: Reconstructed central views - QP2, rank 20; Column 3: Reconstructed central views -QP 2, rank 60

4.2 Baselines

Ahmad *et al.* (2017) interpret the sub-aperture views of the light field as frames in a multiview sequence. They use the multiview extension of HEVC to exploit the 2D interview correlation among the views. Bikes, Fountain and Vincent2, and Stone-Pillars Outside light field images have been used to evaluate this multi-view compression algorithm.

Liu *et al.* (2016) formulate a predictive coding approach and treat light field views as a pseudo-sequence like video. They compress the central view first and the remaining views in a symmetric, 2D hierarchical order. Motion estimation and compensation in video coding systems have been used to perform inter-view prediction. Due to the public unavailability of Liu *et al.* (2016)'s datasets, only Bikes have been used for the experiments.

Standard HEVC has been used as a baseline. The SAIs in YUV420 are passed to the encoder as a pseudo-sequence like video and compressed.

4.3 Implementation Setup

All the experiments were performed on a single high-end HP OMEN X Gaming laptop with an Intel 9th Gen i7-9750H processor, 16 GB RAM, and an RTX 2080 8 GB GPU on Windows 10. Chainer (version 7.7.0) was used to implement and train the CNN. The entire pipeline was built using Matlab and Python.

CHAPTER 5

Results

The performance of the proposed coding scheme is compared to the previous pseudo-sequence based coding schemes and the HEVC encoder (32 bit HM version 11.0). All the baselines have been tested under the same conditions and parameters - $QP \in \{2, 6, 10, 14, 20, 38\}$ and Ranks $r \in \{4, 8, 12, ..., 56, 60\}$. Some of the intermediate QPs and Ranks have been omitted while presenting results for the sake of space and readability. Wherever SSIM scores have been mentioned in this chapter, the metric has been computed w.r.t. the central view of the original light field image.

5.1 Performance of the Proposed Coding Scheme

Figures 5.1 show the bitrate vs PSNR curves for Ahmad *et al.* (2017) on the 3 datasets. They have been plotted separately because of the scale mismatch in the required bitrates with other schemes. Figure 5.2 presents the rate distortion curves for the proposed coding scheme and HEVC. The proposed coding scheme well outperforms the other baselines from the above mentioned plots. The proposed coding scheme significantly saves bitrate and retains reconstruction quality. Table 5.1 shows the bytes written to the bitstream by the algorithms. The proposed scheme beats all other baselines by a significant margin. The mean SSIM score, averaged over all the views and QPs, was computed. The plots are shown in Figure 5.3. The visual quality produced by the proposed scheme is comparable to the other baselines at most ranks.

The proposed coding scheme has an edge over the other baselines in maintaining reconstruction quality. Another assessment was performed using the Bjontegaard Bjontegaard (2001) metric. The Bjontegaard metric is a standard metric that benchmarks two compression algorithms. This metric compares the two rate distortion curves and yields bitrate reduction of one curve over the other. The average percent rate savings is estimated over a range of quantization parameters for the fifteen chosen ranks. These results are shown in Table 5.2. For the sake of readability, only a few key ranks are shown.

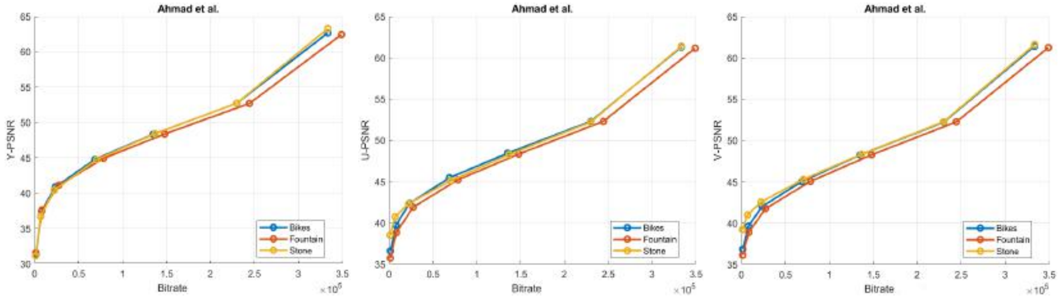

Figure 5.1: The bitrate vs PSNR graphs of Ahmad *et al.* (2017) coding scheme for all three datasets. Each plot describes YPSNR, UPSNR and VPSNR respectively.

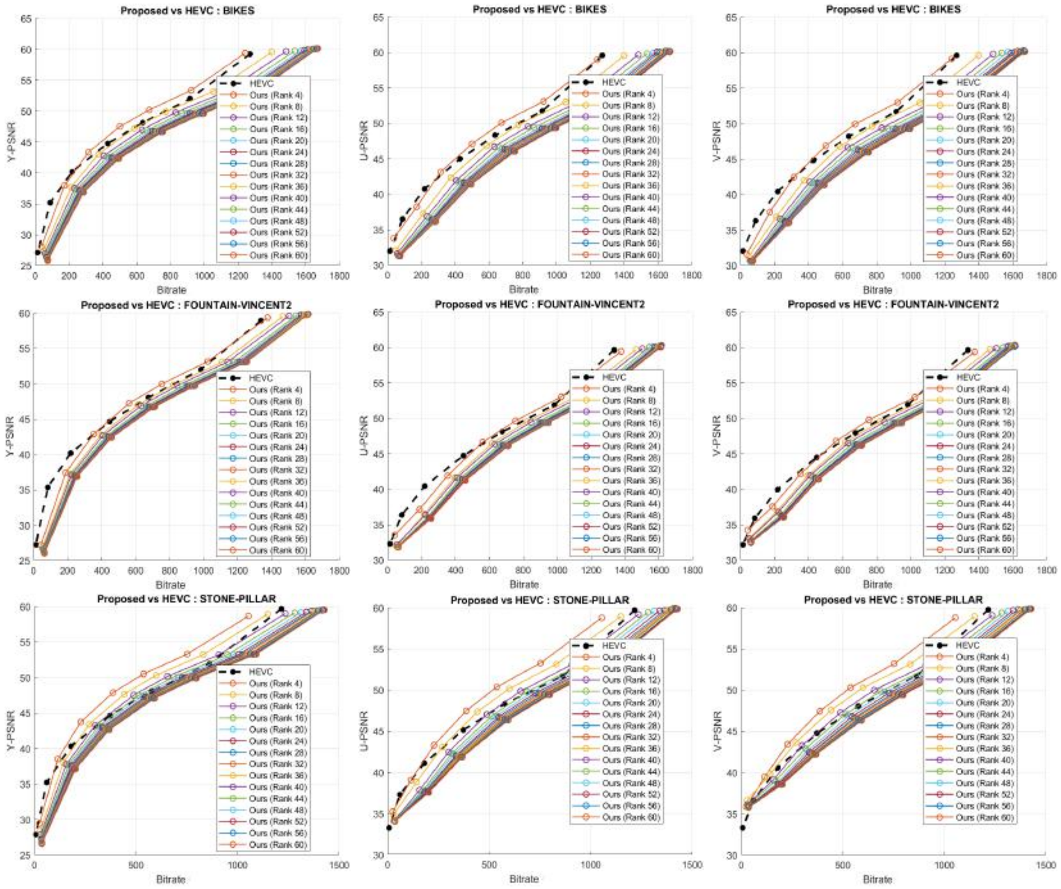

Figure 5.2: Rate-distortion curves for the proposed compression scheme and HEVC codec for the three datasets. Each row represents a different dataset. The columns indicate YPNSR, UPSNR and VPSNR respectively.

On Bikes, the proposed scheme achieves 98.94%, 40.42%, and 81.37% bitrate reduction compared to Ahmad *et al.* (2017), HEVC, and Liu *et al.* (2016) respectively. On Fountain-Vincent 2, the scheme achieves 99.03% and 35.80% bitrate savings compared to Ahmad *et al.* (2017) and HEVC codec respectively. On Stone-Pillars Outside,

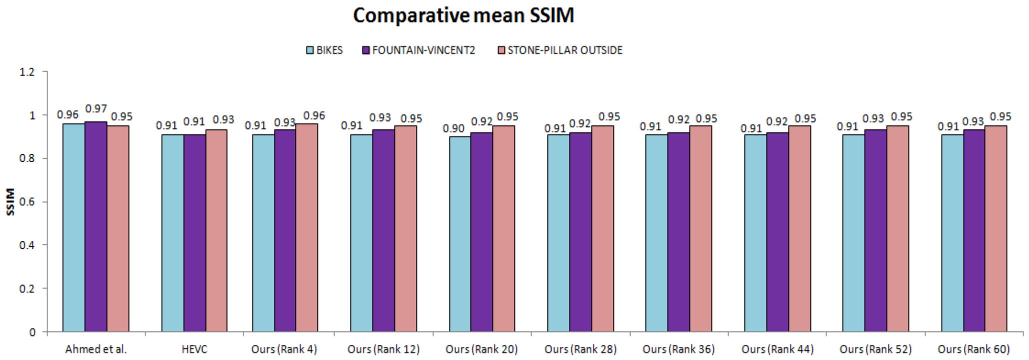

Figure 5.3: Comparative mean SSIM for the proposed coding scheme and the baselines.

Table 5.1: The total number of bytes written to file during compression using our proposed scheme for selected chosen ranks. The values are in *MB*.

Scene		Proposed Scheme			Ahmad	HEVC	Liu
	QP	Rank 4	Rank 32	Rank 60			
BIKES	2	0.47	0.61	0.63	24.31	26.86	13.24
	6	0.35	0.47	0.49	16.77	19.35	8.5
	10	0.25	0.35	0.37	9.85	13.44	4.33
	14	0.19	0.27	0.28	5.02	9.08	1.81
	20	0.12	0.18	0.19	1.7	4.61	0.53
	26	0.06	0.1	0.11	0.58	1.87	0.19
	38	0.01	0.03	0.03	0.11	0.31	0.05
FOUNTAIN-VINCENT2	2	0.52	0.6	0.61	25.46	28.25	-
	6	0.38	0.46	0.47	17.82	20.75	-
	10	0.28	0.35	0.36	10.78	14.3	-
	14	0.21	0.26	0.27	5.73	9.45	-
	20	0.13	0.17	0.17	2.01	4.61	-
	26	0.07	0.09	0.1	0.61	1.78	-
	38	0.02	0.02	0.02	0.11	0.28	-
STONE-PILLARS OUTSIDE	2	0.4	0.52	0.53	24.32	25.69	-
	6	0.28	0.39	0.41	16.76	18.18	-
	10	0.2	0.28	0.3	9.97	12.18	-
	14	0.15	0.21	0.22	5.13	7.88	-
	20	0.09	0.13	0.14	1.61	3.8	-
	26	0.04	0.07	0.08	0.5	1.27	-
	38	0.01	0.01	0.01	0.08	0.17	-

Table 5.2: Bjontegaard percentage rate savings for the proposed compression scheme with respect to Ahmad *et al.* (2017) and HEVC codec (negative values represent gains) on *Bikes* data.

Rank	Ahmad			HEVC			Liu		
	Y	U	V	Y	U	V	Y	U	V
4	-99.25	-99.28	-99.27	-5.88	0.18	-11.74	-67.97	-71.26	-70.64
12	-99.00	-99.03	-99.07	-33.77	-34.64	-35.47	-77.64	-81.00	-79.17
20	-98.90	-98.95	-99.00	-40.61	-41.39	-41.34	-80.15	-83.15	-81.18
28	-98.85	-98.91	-98.95	-43.64	-43.90	-44.04	-81.26	-83.97	-82.24
36	-98.82	-98.88	-98.92	-45.53	-45.35	-45.57	-81.97	-84.51	-82.85
44	-98.80	-98.86	-98.91	-46.79	-46.65	-46.64	-82.43	-84.91	-83.20
52	-98.78	-98.85	-98.90	-47.65	-47.56	-47.21	-82.75	-85.20	-83.46
60	-98.77	-98.84	-98.89	-48.13	-47.80	-47.63	-82.93	-85.33	-83.61
Average	-98.89	-98.94	-98.98	-40.26	-40.01	-40.98	-80.08	-82.86	-81.18

the scheme achieves 99.20% and 22.43% bitrate reduction compared to Ahmad *et al.* (2017) and HEVC respectively. These figures clearly show that the proposed coding scheme outperforms all the other baselines.

5.2 Usage of CNN over Analytical Methods

In BLOCK I of the proposed coding scheme, the layered representation of the light field is generated. This can be done analytically one layer at a time until the solution of 2 converges, or by using a CNN. The 5×5 Bunnies dataset Wetzstein has been used to compare the performance of the CNN and analytical optimization method. The analytical method was evaluated for 10, 25, 50, 75, 100, 125, and 150 iterations. The CNN used was trained with the optimal set of hyperparameters. Figure 5.4 illustrates the PSNR vs computation time plot of this experiment.

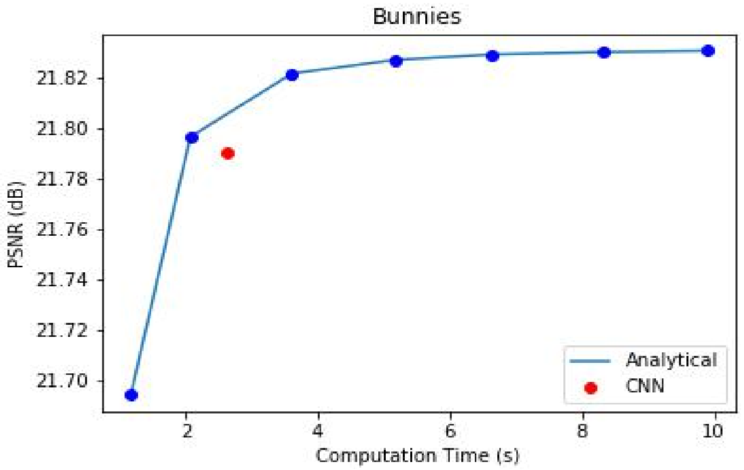

Figure 5.4: Computation time accuracy of reproduced light fields using analytical and CNN-based optimization of multiplicative layers.

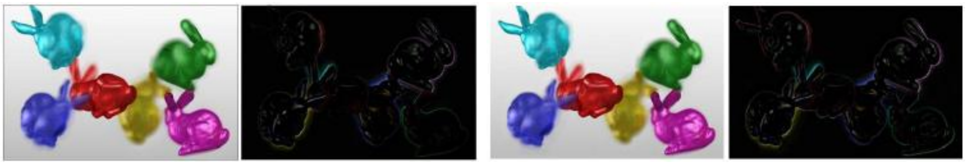

Figure 5.5: View 19 of Bunnies reproduced using analytical method (ANA) and CNN respectively with corresponding difference images. ANA: PSNR:19.94 dB, SSIM:0.895; CNN: PSNR:22.18 dB, SSIM:0.918

In the case of the analytical approach, accuracy gradually saturates with iterations. The inference time for the CNN is marginally longer than that of the analytical method for the same PSNR performance. Nevertheless, Maruyama *et al.* (2020) have demonstrated that with better GPUs and more training data, CNN inference can be performed much faster than the analytical method with the same reconstruction quality. Also, a random view (view 19) reconstructed using analytical and CNN methods along with their error images have been shown in Figure 5.5. The CNN results outperform in terms

Table 5.3: Comparison of computation speed in *BLOCK II* of proposed scheme for *ML(3)* and *AV(169)*. Experiment evaluated on *Bikes* light field for BK-SVD ranks 20 and 60. The values represent runtime in *s*.

Input	Rank	BK-SVD	HEVC		
			QP 2	QP 20	QP 38
ML(3)	20	1.399	38.892	28.261	15.643
AV(169)	20	37.117	2592.1	1360.4	659.835
ML(3)	60	1.711	38.937	28.355	16.101
AV(169)	60	47.22	3068.9	1460.3	680.548

Table 5.4: Bjontegaard percentage rate savings for the proposed compression scheme with respect to Ahmad *et al.* (2017), HEVC and Liu *et al.* (2016) on *Bikes* data for *ML(3)* and*AV(169)* for ranks 20 and 60 (negative values represent gain over anchor).

Input	Rank	Ahmad			HEVC			Liu		
		Y	U	V	Y	U	V	Y	U	V
ML(3)	20	-98.89	-98.94	-98.99	-40.6	-41.39	-41.34	-80.145	-83.14	-81.17
AV(169)	20	-99.5	-99.49	-99.49	64.76	56.71	53.86	-36.78	-50.35	-48.36
ML(3)	60	-98.77	-98.83	-98.88	-48.13	-47.79	-47.62	-82.93	-85.33	-83.6
AV(169)	60	-99.28	-99.29	-99.3	11.08	6.145	5.98	-59.46	-69.2	-67.49

of PSNR and SSIM scores as well. Considering the compute performance tradeoff, the usage of a CNN over analytical methods is justified.

5.3 Usage of multiplicative layers

The proposed scheme generates three multiplicative layers of the input light field. There is a clear advantage to approximate and encode just the three multiplicative layers rather than the entire set SAIs. To confirm this definitively, all 13×13 views of the Bikes light field were directly fed into BLOCK II of the proposed coding scheme. This experiment will be called AV(169). These results were then compared with the usage of 3 layers as described. This will be referred to as ML(3). For both these, BK-SVD was performed for ranks 20 and 60, followed by HEVC encoding with the entire set of QPs.

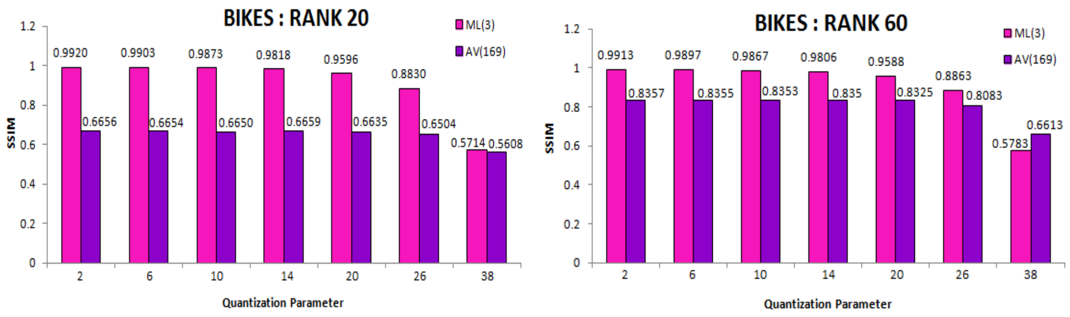

Figure 5.6: Mean SSIM scores over each QP of decoded views in BLOCK III of proposed scheme using *ML(3)* and *AV(169)*. Experiment evaluated on Bikes dataset for BK-SVD ranks 20 and 60

Table 5.3 highlights the computation time for these experiments. The proposed compression scheme works much faster with multiplicative layers. This proves to be advantageous over using all the views of the light field. The bitrate reduction of ML(3) and AV(169) w.r.t. to the baselines is depicted in Table 5.4. As expected, this swings heavily in favor of layers proving that there is a significant amount of redundancy in transmitting views as such. The SSIM was analyzed on the reconstructed views for both ML(3) and AV(169). Mean SSIM over the decoded views was calculated for each QP result and each rank ofBikes. These are illustrated in Figure 5.6. Interestingly, ML(3) significantly outperforms AV(169) in terms of visual quality. Due to all the downhill compression, a deterioration in visual quality is observed when all views are considered. Condensing all the information into a few layer patterns seems to retain enough information to maintain visual quality.

CHAPTER 6

Conclusions

An efficient representation and novel lossy compression scheme for layered light-field displays with light-ray operations regulated using multiplicative layers has been proposed. By allowing the choice of selecting ranks and quantization parameters, it realizes the goal of covering a range of multiple bitrates within a single pipeline. The experiments with benchmarked light field datasets exhibit very competitive results.

Current solutions are not specifically designed to target layered displays. Broadly, compression approaches are classified to work for lenslet-based formats or sub-aperture images based pseudo-sequence representation. The proposed scheme could flexibly work with different light-ray operations and analytical or data-driven CNN-based methods, targeting multi-layered displays. It is adaptable for a variety of multi-view/light field displays. It can also complement existing light-field coding schemes, which employ different networks to encode light field images at different bit rates. This would enable deploying the concept of layered displays on different auto-stereoscopic platforms.

CHAPTER 7

Future Work

The future work spans several directions. The proposed idea can be extended to other light field displays, such as ones with more than three light attenuating layers and projection-based and holographic displays with optical elements constructed using additive layers. Another interesting direction is to extend the proposed mathematical formulation for coded apertures. This could be useful in the compression of dynamic light field content or a focal stack instead of SAIs in tensor displays. The proposed algorithm uses the RGB space in which all channels are equally important. The analysis of compression performance under other color spaces and the implications on image perception would be worth pursuing.

REFERENCES

1. **Ahmad, W.**, **M. Ghafoor**, **S. A. Tariq**, **A. Hassan**, **M. Sjöström**, and **R. Olsson** (2019). Computationally efficient light field image compression using a multiview hevc framework. *IEEE access*, **7**, 143002–143014.

2. **Ahmad, W.**, **R. Olsson**, and **M. Sjöström**, Interpreting plenoptic images as multi-view sequences for improved compression. *In 2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017.

3. **Ahmad, W.**, **S. Vagharshakyan**, **M. Sjöström**, **A. Gotchev**, **R. Bregovic**, and **R. Olsson** (2020). Shearlet transform-based light field compression under low bitrates. *IEEE Transactions on Image Processing*, **29**, 4269–4280.

4. **Bakir, N.**, **W. Hamidouche**, **O. Déforges**, **K. Samrouth**, and **M. Khalil**, Light field image compression based on convolutional neural networks and linear approximation. *In 2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018.

5. **Balogh, T.**, **P. T. Kovács**, and **A. Barsi**, Holovizio 3d display system. *In 2007 3DTV Conference*. IEEE, 2007.

6. **Bjontegaard, G.** (2001). Calculation of average psnr differences between rd-curves. *VCEG-M33*.

7. **Chen, Y.**, **P. An**, **X. Huang**, **C. Yang**, **D. Liu**, and **Q. Wu** (2020). Light field compression using global multiplane representation and two-step prediction. *IEEE Signal Processing Letters*, **27**, 1135–1139.

8. **Cullum, J.** and **W. E. Donath**, A block lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. *In 1974 IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*. IEEE, 1974.

9. **Dib, E.**, **M. Le Pendu**, **X. Jiang**, and **C. Guillemot** (2020). Local low rank approximation with a parametric disparity model for light field compression. *IEEE Transactions on Image Processing*, **29**, 9641–9653.

10. **Geng, J.** (2013). Three-dimensional display technologies. *Advances in optics and photonics*, **5**(4), 456–535.

11. **Golub, G. H.** and **R. Underwood**, The block lanczos method for computing eigenvalues. *In Mathematical software*. Elsevier, 1977, 361–377.

12. **Gortler, S. J.**, **R. Grzeszczuk**, **R. Szeliski**, and **M. F. Cohen**, The lumigraph. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996.

13. **Gu, J.**, **B. Guo**, and **J. Wen**, High efficiency light field compression via virtual reference and hierarchical mv-hevc. *In 2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019.

14. **Halko, N.**, **P.-G. Martinsson**, and **J. A. Tropp** (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, **53**(2), 217–288.

15. **Hériard-Dubreuil, B.**, **I. Viola**, and **T. Ebrahimi**, Light field compression using translation-assisted view estimation. *In 2019 Picture Coding Symposium (PCS)*. IEEE, 2019.

16. **Hirsch, M.**, **G. Wetzstein**, and **R. Raskar** (2014). A compressive light field projection system. *ACM Transactions on Graphics (TOG)*, **33**(4), 1–12.

17. **Hu, X.**, **J. Shan**, **Y. Liu**, **L. Zhang**, and **S. Shirmohammadi** (2020). An adaptive two-layer light field compression scheme using gnn-based reconstruction. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, **16**(2s), 1–23.

18. **Huang, X.**, **P. An**, **F. Cao**, **D. Liu**, and **Q. Wu** (2019). Light-field compression using a pair of steps and depth estimation. *Optics express*, **27**(3), 3557–3573.

19. **Huang, X.**, **P. An**, **L. Shan**, **R. Ma**, and **L. Shen**, View synthesis for light field coding using depth estimation. *In 2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018.

20. **Jia, C.**, **X. Zhang**, **S. Wang**, **S. Wang**, and **S. Ma** (2018). Light field image compression using generative adversarial network-based view synthesis. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, **9**(1), 177–189.

21. **Jiang, X.**, **M. Le Pendu**, **R. A. Farrugia**, and **C. Guillemot** (2017). Light field compression with homography-based low-rank approximation. *IEEE Journal of Selected Topics in Signal Processing*, **11**(7), 1132–1145.

22. **Kobayashi, Y.**, **S. Kondo**, **K. Takahashi**, and **T. Fujii** (2017*a*). A 3-d display pipeline: Capture, factorize, and display the light field of a real 3-d scene. *ITE Transactions on Media Technology and Applications*, **5**(3), 88–95.

23. **Kobayashi, Y.**, **K. Takahashi**, and **T. Fujii**, From focal stacks to tensor display: A method for light field visualization without multi-view images. *In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017*b*.

24. **Lee, S.**, **C. Jang**, **S. Moon**, **J. Cho**, and **B. Lee** (2016). Additive light field displays: realization of augmented reality with holographic optical elements. *ACM Transactions on Graphics (TOG)*, **35**(4), 1–13.

25. **Levoy, M.** and **P. Hanrahan**, Light field rendering. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996.

26. **Li, L.**, **Z. Li**, **B. Li**, **D. Liu**, and **H. Li** (2017). Pseudo-sequence-based 2-d hierarchical coding structure for light-field image compression. *IEEE Journal of Selected Topics in Signal Processing*, **11**(7), 1107–1119.

27. **Li, T.**, **Q. Huang**, **S. Alfaro**, **A. Supikov**, **J. Ratcliff**, **G. Grover**, and **R. Azuma**, Light-field displays: a view-dependent approach. *In ACM SIGGRAPH 2020 Emerging Technologies*. 2020, 1–2.

28. **Liu, D.**, **P. An**, **R. Ma**, **W. Zhan**, **X. Huang**, and **A. A. Yahya** (2019). Content-based light field image compression method with gaussian process regression. *IEEE Transactions on Multimedia*, **22**(4), 846–859.

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

30. **Liu, D.**, **L. Wang**, **L. Li**, **Z. Xiong**, **F. Wu**, and **W. Zeng**, Pseudo-sequence-based light field image compression. *In 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2016.

31. **Maruyama, K.**, **Y. Inagaki**, **K. Takahashi**, **T. Fujii**, and **H. Nagahara**, A 3-d display pipeline from coded-aperture camera to tensor light-field display through cnn. *In 2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019.

32. **Maruyama, K.**, **K. Takahashi**, and **T. Fujii** (2020). Comparison of layer operations and optimization methods for light field display. *IEEE Access*, **8**, 38767–38775.

33. **Musco, C.** and **C. Musco** (2015). Randomized block krylov methods for stronger and faster approximate singular value decomposition. *arXiv preprint arXiv:1504.05477*.

34. **Pedregosa, F.**, **G. Varoquaux**, **A. Gramfort**, **V. Michel**, **B. Thirion**, **O. Grisel**, **M. Blondel**, **P. Prettenhofer**, **R. Weiss**, **V. Dubourg**, *et al.* (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, **12**, 2825–2830.

35. **Pennebaker, W. B.** and **J. L. Mitchell**, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

36. **Saito, T.**, **Y. Kobayashi**, **K. Takahashi**, and **T. Fujii** (2016). Displaying real-world light fields with stacked multiplicative layers: requirement and data conversion for input multiview images. *Journal of Display Technology*, **12**(11), 1290–1300.

37. **Schiopu, I.** and **A. Munteanu** (2019). Deep-learning-based macro-pixel synthesis and lossless coding of light field images. *APSIPA Transactions on Signal and Information Processing*, **8**.

38. **Senoh, T.**, **K. Yamamoto**, **N. Tetsutani**, and **H. Yasuda**, Efficient light field image coding with depth estimation and view synthesis. *In 2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018.

39. **Sharma, M.** (2017). *Uncalibrated camera based content generation for 3D multi-view displays*. Ph.D. thesis.

40. **Sharma, M.**, **S. Chaudhury**, and **B. Lall** (2016). A novel hybrid kinect-variety-based high-quality multiview rendering scheme for glass-free 3d displays. *IEEE Transactions on Circuits and Systems for Video Technology*, **27**(10), 2098–2117.

41. **Sharma, M.**, **S. Chaudhury**, **B. Lall**, and **M. Venkatesh** (2014). A flexible architecture for multi-view 3dtv based on uncalibrated cameras. *Journal of Visual Communication and Image Representation*, **25**(4), 599–621.

42. **Sharma, M.** and **G. Ragavan**, A novel randomize hierarchical extension of mv-hevc for improved light field compression. *In 2019 International Conference on 3D Immersion (IC3D)*. IEEE, 2019.

43. **Sullivan, G. J.**, **J.-R. Ohm**, **W.-J. Han**, and **T. Wiegand** (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, **22**(12), 1649–1668.

44. **Surman, P.** and **X. W. Sun**, Towards the reality of 3d imaging and display. *In 2014 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2014.

45. **Takahashi, K.**, **Y. Kobayashi**, and **T. Fujii** (2018). From focal stack to tensor light-field display. *IEEE Transactions on Image Processing*, **27**(9), 4571–4584.

46. **Takahashi, K.**, **T. Saito**, **M. P. Tehrani**, and **T. Fujii**, Rank analysis of a light field for dual-layer 3d displays. *In 2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015.

47. **Vagharshakyan, S.**, **R. Bregovic**, and **A. Gotchev** (2017). Light field reconstruction using shearlet transform. *IEEE transactions on pattern analysis and machine intelligence*, **40**(1), 133–147.

48. **Wang, B.**, **Q. Peng**, **E. Wang**, **K. Han**, and **W. Xiang** (2019). Region-of-interest compression and view synthesis for light field video streaming. *IEEE Access*, **7**, 41183–41192.

49. **Watanabe, H.**, **N. Okaichi**, **T. Omura**, **M. Kano**, **H. Sasaki**, and **M. Kawakita** (2019). Aktina vision: Full-parallax three-dimensional display with 100 million light rays. *Scientific reports*, **9**(1), 1–9.

50. **Wetzstein, G.** (). Synthetic light field archive - mit media lab. `https://web.media.mit.edu/~gordonw/SyntheticLightFields/`. Accessed: 07-03-2021.

51. **Wetzstein, G.**, **D. R. Lanman**, **M. W. Hirsch**, and **R. Raskar** (2012). Tensor displays: compressive light field synthesis using multilayer displays with directional backlighting.

52. **Zhao, Z.**, **S. Wang**, **C. Jia**, **X. Zhang**, **S. Ma**, and **J. Yang**, Light field image compression based on deep learning. *In 2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018.