Selected topics in learning-based coding for light field imaging

Milan Stepanov

université PARIS-SACLAY

Selected topics in learning-based coding for light field imaging

Sujets sélectionnés dans le codage basé sur l'apprentissage pour l'imagerie en champ lumineux

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Traitement du signal et des images
Graduate School : Informatique et sciences du numérique (ISN),
Référent : CentraleSupélec

Thèse préparée dans la unité de recherche Laboratoire des signaux et systèmes (Université Paris-Saclay, CNRS, CentraleSupélec) sous la direction de Frédéric DUFAUX, Directeur de recherche et la co-encadrement de Giuseppe VALENZISE, Chargé de recherche.

Thèse soutenue à Paris-Saclay, le 04 juillet 2022, par

Milan STEPANOV

Composition du jury

Patrick LE CALLET Prof. des Universités, Université de Nantes	Président
Adrian MUNTEANU Prof., Université de Bruxelles	Rapporteur & Examinateur
Lu ZHANG Maîtresse de conférences, HDR, INSA Rennes	Rapporteur & Examinatrice
Federica BATTISTI Assistant Professeure, Université de Padova	Examinatrice
Frédéric DUFAUX Directeur de recherche, L2S, UPS, CNRS, Centrale-Supélec	Directeur de thèse
Giuseppe VALENZISE Chargé de recherche, L2S, UPS, CNRS, Centrale-Supélec	Co-encadrant de thèse

THESE DE DOCTORAT
NNT : 2022UPASG050

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Sujets sélectionnés dans le codage basé sur l'apprentissage pour l'imagerie en champ lumineux
Mots clés: l'imagerie en champ lumineux, apprentissage profond, codage d'image

Résumé: La tendance actuelle en matière de technologie d'imagerie est d'aller au-delà de la représentation 2D du monde capturée par une caméra conventionnelle. La technologie de champ lumineux, light field, nous permet de capturer des repères directionnels plus riches. Avec la disponibilité récente des caméras portables à champ lumineux, il est possible de capturer facilement une scène sous différentes perspectives en un seul temps d'exposition, permettant de nouvelles applications telles qu'un changement de perspective, la mise au point à différentes profondeurs de la scène et l'édition. profondeur de champ.

Alors que le nouveau modèle d'imagerie repousse les frontières de l'immersion, de la qualité de l'expérience et de la photographie numérique, il génère d'énormes quantités de données exigeant des ressources de stockage et de bande passante importantes. Surpasser ces défis, les champs lumineux nécessitent le développement de schémas de codage efficaces.

Dans cette thèse, nous explorons des approches basées sur l'apprentissage profond pour la compression du champ lumineux. Notre schéma de codage hybride combine une approche de compression basée sur l'apprentissage avec un schéma de codage vidéo traditionnel et offre un outil très efficace pour la compression avec perte d'images en champ clair. Nous utilisons une architecture basée sur un encodeur automatique et un goulot d'étranglement contraint par l'entropie pour obtenir une opérabilité particulière du codec de base. De plus, une couche d'amélioration basée sur un codec vidéo traditionnel offre une évolutivité de qualité fine au-dessus de la couche de base. Le codec proposé atteint de meilleures performances par rapport aux méthodes de pointe ; les expériences quantitatives montrent, en moyenne, une réduction de débit de plus de 30 % par rapport aux codecs JPEG Pleno et HEVC. De plus, nous proposons un codec de champ lumineux sans perte basé sur l'apprentissage qui exploite les méthodes de synthèse de vue pour obtenir des estimations de haute qualité et un modèle auto-régressif qui construit une distribution de probabilité pour le codage arithmétique. La méthode proposée surpasse les méthodes de pointe en termes de débit binaire tout en maintenant une faible complexité de calcul. Enfin, nous étudions le paradigme de codage de source distribué pour les images en champ lumineux. Nous tirons parti des capacités de modélisation élevées des méthodes d'apprentissage en profondeur au niveau de deux blocs fonctionnels critiques du schéma de codage de source distribué : pour l'estimation des vues Wyner-Ziv et la modélisation du bruit de corrélation. Notre étude initiale montre que l'intégration d'une méthode de synthèse de vues basée sur l'apprentissage profond dans un schéma de codage distribué améliore les performances de codage par rapport au HEVC Intra. Nous obtenons des gains supplémentaires en intégrant la modélisation basée sur l'apprentissage en profondeur du signal résiduel.

Title: Selected topics in learning-based coding for light field imaging
Keywords: light field, deep learning, image coding

Abstract: The current trend in imaging technology is to go beyond the 2D representation of the world captured by a conventional camera. Light field technology enables us to capture richer directional cues. With the recent availability of handheld light field cameras, it is possible to capture a scene from various perspectives with ease at a single exposure time, enabling new applications such as a change of perspective, focusing at different depths in the scene, and editing depth-of-field.

Whereas the new imaging model increases frontiers of immersiveness, quality of experience, and digital photography, it generates huge amounts of data demanding significant storage and bandwidth resources. To overcome these challenges, light fields require the development of efficient coding schemes.

In this thesis, we explore deep-learning-based approaches for light field compression. Our hybrid coding scheme combines a learning-based compression approach with a traditional video coding scheme and offers a highly efficient tool for lossy compression of light field images. We employ an auto-encoder-based architecture and an entropy constrained bottleneck to achieve particular operability of the base codec. In addition, an enhancement layer based on a traditional video codec offers fine-grained quality scalability on top of the base layer. The proposed codec achieves better performance compared to state-of-the-art methods; quantitative experiments show, on average, more than 30% bitrate reduction compared to JPEG Pleno and HEVC codecs. Moreover, we propose a learning-based lossless light field codec that leverages view synthesis methods to obtain high-quality estimates and an auto-regressive model that builds probability distribution for arithmetic coding. The proposed method outperforms state-of-the-art methods in terms of bitrate while maintaining low computational complexity. Last but not least, we investigate distributed source coding paradigm for light field images. We leverage the high modeling capabilities of deep learning methods at two critical functional blocks in the distributed source coding scheme: for the estimation of Wyner-Ziv views and correlation noise modeling. Our initial study shows that incorporating a deep learning-based view synthesis method into a distributed coding scheme improves coding performance compared to the HEVC Intra. We achieve further gains by integrating the deep-learning-based modeling of the residual signal.

To My Family and Friends

Acknowledgements

I want to thank my supervisors, Directeur de recherche Frédéric DUFAUX and Chargé de recherche Giuseppe VALENZISE, for the great opportunity they have provided, their guidance, and their help. I was fortunate to receive a warm welcome, openness, and encouragement upon my arrival. Furthermore, you believed in me and supported me along the way, to which I am immensely grateful. I would also like to thank Prof. Søren FORCHHAMMER and Prof. Marta MRAK for their kind supervision during my secondment periods. Furthermore, I want to express my gratitude to my thesis committee: MCF Lu ZHANG, Assistant Professor Federica BATTISTI, Prof. Adrian MUNTEANU, and Prof. Patrick LE CALLET. Thank you very much for accepting to serve as jury members. I appreciate the time you take to read my manuscript and give me suggestions.

I had the privilege to share my office with Yassine, Gordana, Stefano, and Kuba during the pre-Covid period. Thank you for the exciting discussions and pieces of advice and for making a place for me during lunch breaks.

My Ph.D. thesis would not be possible without the support of a Marie Skłodowska-Curie project, RealVision. I am highly thankful as it provided awesome training and mobility opportunities, a chance to meet and collaborate with great people, peace of mind to focus on my research, and a value of belonging to a bigger goal. Also, I am grateful to the RealVision supervisors and administration for bringing this great opportunity to other students and me.

I thank my colleague Muhammad Umair Mukati for engaging in discussions and the great work during our collaboration. I enjoyed our meetings, designing experiments, and writing papers together. Also, from the bottom of my heart, I am thankful to Ali, Abhishek, and all other RV champions with whom I spent great moments during our training schools and social events.

Last but not least, I feel incredibly fortunate to have great support from my family and friends during the period of my Ph.D. thesis. I am grateful to my parents, Jovan and Slobodanka, without whose love and guidance I would not be here. Many thanks to my brother for his support during my long absence from home. I am heavily obliged to my aunt Ljiljana and my first man Milan for their long calls and for bringing new perspectives into my personal life. Finally, I am indebted to my Hélène for the support during the past three years, for dragging me to vacations, and for bringing great joy into my life.

Contents

1 Introduction **17**
- 1.1 Motivation . . . 18
  - 1.1.1 Autoencoder-based lossy compression . . . 18
  - 1.1.2 Deep autoregressive models for lossless compression . . . 18
  - 1.1.3 Deep distributed source coding . . . 19
- 1.2 Objectives and contributions . . . 19
- 1.3 Thesis outline . . . 20

2 Light field imaging **21**
- 2.1 Light fields . . . 21
- 2.2 Light field acquisition . . . 22
  - 2.2.1 Multi-camera array . . . 22
  - 2.2.2 LF gantries . . . 22
  - 2.2.3 Plenoptic cameras . . . 23
  - 2.2.4 Summary . . . 26
- 2.3 Light field representation . . . 26
  - 2.3.1 Lenslet representation . . . 26
  - 2.3.2 Viewpoint-based representation . . . 27
  - 2.3.3 Epipolar plane images . . . 28
  - 2.3.4 Other representations . . . 29
- 2.4 Light field compression . . . 29
- 2.5 Light field rendering and display . . . 32
- 2.6 Light field quality evaluation . . . 33

3 An overview of light field coding technologies **35**
- 3.1 Introduction . . . 35
- 3.2 Transform-based methods . . . 36
  - 3.2.1 Approaches based on DCT . . . 36
  - 3.2.2 Approaches based on KLT . . . 37
  - 3.2.3 Approaches based on DWT . . . 37
  - 3.2.4 Summary . . . 38
- 3.3 Prediction-based methods . . . 39
  - 3.3.1 Inter-view prediction . . . 39
  - 3.3.2 Non-local spatial prediction . . . 40
  - 3.3.3 View synthesis-based prediction . . . 41
  - 3.3.4 Summary . . . 43
- 3.4 JPEG Pleno codec . . . 44
- 3.5 Lossless light field coding . . . 46

3.6 Conclusions and perspectives . . . 47

4 Learning-based lossy light field compression **49**
4.1 Introduction . . . 49
4.2 Related work . . . 50
4.3 Preliminary studies . . . 51
4.3.1 Block-based approach . . . 51
4.3.2 Towards a holistic approach . . . 55
4.4 Hybrid codec . . . 59
4.4.1 Base layer . . . 60
4.4.2 Enhancement layer . . . 61
4.4.3 Quantitative analysis - Enhancement layers . . . 63
4.4.4 Quantitative analysis - State-of-the-art . . . 65
4.5 Conclusion . . . 67

5 Learning-based lossless light field compression **69**
5.1 Introduction . . . 69
5.2 Related work . . . 70
5.2.1 View synthesis methods . . . 70
5.2.2 Autoregressive models and lossless compression . . . 71
5.3 Proposed method . . . 72
5.3.1 View synthesis . . . 72
5.3.2 Entropy model . . . 74
5.3.3 Loss function . . . 75
5.3.4 The architecture . . . 75
5.4 Experiments . . . 77
5.4.1 Datasets . . . 77
5.4.2 Training procedure . . . 78
5.4.3 Scheme ablations . . . 78
5.5 Results . . . 84
5.5.1 Compression performance . . . 84
5.5.2 Runtime . . . 86
5.6 Conclusion . . . 87

6 Distributed light field coding **89**
6.1 Introduction . . . 89
6.2 Background . . . 90
6.3 Related work . . . 92
6.3.1 DLFC approaches . . . 92
6.3.2 DVC and DMVC approaches . . . 93
6.3.3 SI generation methods . . . 93
6.4 Proposed method . . . 94
6.4.1 Distributed LF compression . . . 94

6.4.2 View synthesis . . . 97
6.4.3 Correlation noise modeling . . . 98
6.5 Experiments . . . 102
6.5.1 Datasets . . . 102
6.5.2 Ablation studies . . . 104
6.6 Results . . . 109
6.6.1 RD performance . . . 109
6.6.2 Visual analysis . . . 112
6.7 Conclusion . . . 113

7 Conclusions **117**
7.1 Overview and outcome . . . 117
7.2 Limitations and future prospect . . . 118

A Learning-based Lossy Light Field Compression **121**
A.1 Visual evaluation . . . 121
A.2 Residual signals . . . 125
A.3 Convex hull generation . . . 126
A.3.1 Scalar Quantization . . . 126
A.3.2 Intra Coding . . . 127
A.3.3 Inter Coding . . . 128
A.4 Comparison of enhancement layers . . . 129
A.5 Comparison with state-of-the-art methods . . . 130

B Learning-based Lossless Light Field Compression **131**
B.1 *Base-128* scheme . . . 131
B.2 Comparison between proposed scheme and *Base* . . . 132

C Résumé en français **137**
C.1 Introduction . . . 137
C.2 Motivations . . . 138
C.2.1 Compression avec perte basée sur l'encodeur automatique . . . 138
C.2.2 Modèles autorégressifs profonds pour une compression sans perte . . . 138
C.2.3 Codage source distribué en profondeur . . . 139
C.3 Objectifs et contributions . . . 139
C.4 Aperçu de la thèse . . . 140

Bibliography **141**

List of Figures

2.1 Plenoptic function. 21
2.2 LF acquisition devices based on conventional camera(s). 22
2.3 Plenoptic cameras. 23
2.4 Optical design of the unfocused plenoptic camera [98]. 24
2.5 Optical design of the focused plenoptic camera. 24
2.6 Relation between an object's depth object and the size of its recorded image. 28
2.7 Center views from lenslet LF images. 31
2.8 Center views from HDCA LF images. 31
2.9 Center views from synthetic LF images. 31

3.1 Classification of LF coding solutions (inspired by taxonomy presented in [25]. 35
3.2 Generic block diagram of transform-based coding. 36
3.3 Block diagram of predictive coding. 39
3.4 JPEG Pleno 4D Transform mode encoder. Inspired by [31]. 45
3.5 JPEG Pleno 4D Prediction mode encoder. Inspired by [122]. 45

4.1 The parameters of the autoencoder used in the patch-based codec. Each box defines a layer with a structure [*convolution dimension*]D | f[*kernel size*] | s[*stride step*] | [*padding*: **same** or **valid**] | b[*using bias weight* (+) *or not* (-)] | d[*dilation size*] | [*activation*: **(I)GDN** or **No**]. 52
4.2 RD comparison of learned block-based codec and HEVC in terms of PSNR. 53
4.3 RD comparison of learned block-based codec and HEVC in terms of SSIM. 54
4.4 Visual evaluation of the central view of *Bikes* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d). 56
4.5 Convolution mechanism. 57
4.6 RD comparison in terms of PSNR. 58
4.7 RD comparison in terms of SSIM. 59
4.8 Proposed hybrid coding scheme. 60
4.9 Neural network architecture. The parameters in each block denote the number of filters, the spatial extent of the filter, stride, the usage of the bias, and the activation function. 61
4.10 The histograms of residual signals obtained by compression with different models for content *Bikes*. 62

4.11 (a) RD curves obtained for *Bikes* content using the base layer and scalar quantization as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points. . . 63

4.12 (a) RD curves obtained for *Bikes* content using the base layer and HEVC codec (Intra) as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points. . . 63

4.13 (a) RD curves obtained for *Bikes* content using the base layer and HEVC codec (Inter) as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points. . . 63

4.14 Comparison of performance of the base layer and three variants of the hybrid codec in terms of PSNR and SSIM for the *Bikes* content. 64

4.15 Comparison of the proposed approach to state-of-the-art methods in terms of PSNR. 66

4.16 Comparison of the proposed approach to state-of-the-art methods in terms of SSIM. 66

5.1 The block diagram of the proposed method. The View Synthesis block estimates a view for coding $I_{\tilde{u}}$. The prediction is provided to the Entropy Model block that estimates the probability distribution of the residual signal, which is encoded by the AC module using the predicted distribution. The decoder operates symmetrically with dashed lines illustrating the decoding pipeline. Given the estimated prediction and the probability distribution of the residual signal, the bitstream is decoded using the Arithmetic Decoding (AD) module, and the decoded residual signal is added to the prediction to obtain the final reconstruction. 73

5.2 The grouping introduced in proposed architecture. Each number represents a group, while the arrows denote dependence between groups. E.g., the pixels in the third group rely on pixels from the first and the second group for modeling. 76

5.3 The encoding procedure of the proposed spatial autoregression with four groups. The prediction of the current view $\tilde{x}$ and previously decoded spatial groups are provided to the EM networks that estimate parameters of the probability mass function for each pixel in the current group. The bitstream of each group is obtained by encoding each group using computed probabilities and arithmetic coder. 77

5.4 Hierarchical levels for prediction of views with Corner arrangement (left) and Cross arrangement (middle). At each level i, views can be predicted from decoded views from previous levels, i.e. $i_{prev} < i$. Prediction scenarios applied in Hybrid variant (right). The number denotes the prediction level i while the superscripts denote the arrangement of the reference views used for the prediction, i.e., x and $+$ denote Corner and Cross arrangements. 79
5.5 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Bikes*. 81
5.6 Intermediate results of probability distribution modeling using *Base* scheme for the center view of content *Bikes*. 82
5.7 Intermediate results of probability distribution modeling using Proposed scheme for the center view of content *Bikes*. 83
5.8 Intermediate results of probability distribution modeling using *Base* scheme for view (1, 2) of content *Bikes*. 83
5.9 Intermediate results of probability distribution modeling using Proposed scheme for view (1, 2) of content *Bikes*. 84

6.1 DSC pipeline in the scenario with two separate sources. 91
6.2 Block diagram of transform-domain WZ encoder. 95
6.3 View splitting modes. 95
6.4 Block diagram of transform-domain WZ decoder. 96
6.5 Thumbnails of central views of added test LFs from the EPFL dataset [110]. 104
6.6 RD performance comparison between different variations of the proposed DLFC scheme utilizing three different residual estimation methods, at quantization indices $M = [1, 4, 7, 8]$, using PSNR as distortion metric. 108
6.7 RD performance comparison of distributed and conventional coding schemes using PSNR as distortion metric at quantization indices $M = [1, 4, 7, 8]$, whereas, the quantization parameters specified in Table 6.3 are used for both HEVC plots. 110
6.8 Visual comparison between the outputs of stages in the proposed decoding scheme to decode the central view of the two LF sequences i.e. *Fountain* and *Vespa* at quantization index $M = 8$. The ground truth image and corresponding zoomed patches are shown on the left. The synthesized and the reconstructed WZ view along with the corresponding absolute errors (range normalized to $0.00 - 0.04$) are shown in the next four columns. The zoomed patches are extracted from the highlighted regions in the ground truth images. 113

A.1 Visual evaluation of the central view of *Bikes* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d). . . . 121
A.2 Visual evaluation of the central view of *Danger* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d). . . . 122
A.3 Visual evaluation of the central view of *Pillars* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d). . . . 123
A.4 Visual evaluation of the central view of *Fountain* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d). . . . 124
A.5 Histogram of the residual signals obtained using various *Base* models. 125
A.6 Generation of the convex hull for the hybrid codec with scalar quantization-based enhancement layer in terms of PSNR and SSIM. 126
A.7 Generation of the convex hull for the hybrid codec with HEVC Intra-based enhancement layer in terms of PSNR and SSIM. . . . 127
A.8 Generation of the convex hull for the hybrid codec with HEVC Inter-based enhancement layer in terms of PSNR and SSIM. . . . 128
A.9 RD comparison of hybrid codecs in terms of PSNR and SSIM. . . 129
A.10 RD comparison of the proposed hybrid codec and state-of-the-art methods in terms of PSNR (left) and SSIM (right). . . . 130

B.1 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Bikes*. . . . 132
B.2 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Danger*. . . . 132
B.3 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Flowers*. . . . 132
B.4 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Pillars*. . . . 133
B.5 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Vespa*. . . . 133
B.6 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Ankylosaurus*. . . . 133
B.7 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Desktop*. . . . 133
B.8 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Magnets*. . . . 134
B.9 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Fountain*. . . . 134
B.10 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Friends*. . . . 134

B.11 Distribution of bitrates (top) and PSNRs (bottom) across views for content *Color Chart*. 134
B.12 Distribution of bitrates (top) and PSNRs (bottom) across views for content *ISO Chart*. 135

List of Tables

4.1 BD-rate savings of the proposed hybrid approach facilitating one of three types of enhancement layers against the base layer. 65
4.2 BD-rate savings of the proposed hybrid approach (base layer with HEVC-Inter at the enhancement layer) with respect to the base layer (no enhancement layer), MuLE, and HEVC reference software x265. 65

5.1 The performance evaluation of variants of *Base* method presented in terms of bpp. 79
5.2 The performance evaluation of variants of the proposed method presented in terms of bitrate (bpp). 80
5.3 The performance evaluation in terms of bpp of image compression tools applied on each view separately. 85
5.4 The comparison of the proposed method and available methods from literature in terms of bitrate (bpp). 85
5.5 The comparison of total LF encoding and decoding times presented in minutes. 86

6.1 The network architecture of initial residual estimation. k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation. 101
6.2 The network architecture of refined residual estimation (aided by decoded bands). k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation. In this network each layer is followed by batch normalization. 103
6.3 Quantization parameters of the key views corresponding to four quantization indices $M = [1, 4, 7, 8]$ from the set in [7] to have consistent quality of reconstructed views. 104
6.4 Performance evaluation of four arrangements for view synthesis task across three datasets in terms of PSNR (dB). 105
6.5 Performance evaluation of three loss functions for view synthesis task on Cross arrangement across three datasets in terms of PSNR (dB). 106
6.6 Quantitative evaluation of view synthesis approach given distorted *Cross* arrangement reference views from the *EPFL-LPT* dataset in terms of PSNR (dB). 107
6.7 Average coding performance in terms of BD-PSNR [dB] compared to HEVC-Intra. 111

6.8 Average coding performance in terms of BD-Rate [%] compared to HEVC-Intra. 111

B.1 The performance evaluation in terms of bitrate (bpp) of variant *Base-128*. 131

Acronyms

1D One-Dimensional. 38

2D Two-Dimensional. 17, 22, 26, 31, 32, 35, 37–41, 49–52, 55, 57–60, 67, 68, 137

2PP Two-Plane Parametrization. 21–23, 26

3D Three-Dimensional. 17, 36, 37, 40, 42, 50, 51, 55, 57–59, 67, 72, 100, 137

4D Four-Dimensional. 5, 21, 22, 26, 36–39, 44, 45, 47, 49, 51, 52, 55, 57

4DPM Four-Dimensional Prediction Mode. 44–46

4DTM Four-Dimensional Transform Mode. 44–46

AC Arithmetic Coding. 6, 72, 73

AD Arithmetic Decoding. 6, 73

AR Augmented Reality. 32, 33

AVC Advanced Video Coding. 29

BPG Better Portable Graphics. 71, 72

bpp Bits Per Pixel. 11, 56, 64, 79–81, 85

CALIC Context-based Adaptive Lossless Image Codec. 46, 47

CNM Correlation Noise Modeling. 93, 95

CNN Convolutional Neural Network. 43, 61, 71, 72

CRC Cyclic Redundancy Check. 91, 94, 95

CTC Common Test Conditions. 30, 33, 54

DCT Disrete Cosine Transform. 36–38, 44, 47, 91, 94, 95, 100, 104

DIBR Depth-Image-Based-Rendering. 32, 41, 42

DLFC Distributed Light Field Coding. 90, 94, 97, 105, 106, 109, 111, 114, 115

DMVC Distributed Multi-view Video Coding. 89, 92–94, 109

DSC Distributed Source Coding. 89, 90, 92, 93, 114

DVC Distributed Video Coding. 89, 91–93

DWT Disrete Wavelet Transform. 36–39

EM Entropy Model. 6, 74, 76, 77

EPI Epipolar Plane Image. 26, 28, 42, 46, 51

EPIC Context Adaptive Compression of Epipolar Plane Images. 84–86

FDL Fourier Disparity Layers. 42

FLIF Free Lossless Image Format. 72, 84–86

GOPs Group of Pictures. 91, 93

HDCA High-Density Camera Array. 30, 31, 40, 41, 44

HEVC High Efficiency Video Coding. 29, 49, 54, 58, 62–65, 67, 68, 71, 84, 97, 104, 109, 111, 112, 114, 115

HMD Head-Mounted Display. 32

KLT Karhunen Loève Transform. 36, 37

L3C Learned LossLess image Compression. 81, 84–86

LDPCA Low-Density Parity Check Accumulate. 91, 93–95, 102, 104, 112

LF Light Field. 5, 7, 11, 17–23, 26–33, 35–47, 49–51, 53–55, 60–62, 64, 67–70, 72, 77–82, 84–90, 92–94, 97, 104, 105, 109, 112–114, 117–119, 137–140

LPT Lytro Power Tool. 77, 78, 104–106

MI Micro Image. 23, 26–29, 36, 37, 40, 41, 43, 47, 51, 55, 60

MSE Mean Squared Error. 47, 91

PDB Previously Decoded Band. 94, 95, 101, 114

PSNR Peak Signal-to-Noise Ratio. 6–9, 33, 38, 54, 58, 64–66, 71, 81, 109, 111, 112, 132–135

PVS Pseudo-Video Sequence. 29, 33, 38–40, 42–44, 51, 54, 55, 59, 89, 93

QP Quantization Parameter. 40

RC Residual Compressor. 74, 75, 84–86

RCT Reversible Color Transformation. 84

RD Rate-Distortion. 33, 50, 51, 54, 58, 61–65, 67, 93, 98, 99, 104, 106, 109, 111, 114

RNN Recurrent Neural Network. 71

SAI Sub-Aperture Image. 26–29, 32, 33, 37, 38, 40, 41, 51, 55, 57, 89

SI Side Information. 89, 90, 92–94, 97, 99, 105, 113

SS Self-Similarity. 41

SSIM Structural Similarity Index. 33, 54

ST Shearlet Transform. 42

SW Slepian-Wolf. 90–92

VR Virtual Reality. 32

WZ Wyner-Ziv. 7, 89–95, 98–104, 109, 112–114

1 - Introduction

The current trend in imaging technology is to go beyond the Two-Dimensional (2D) representation of the world captured by a conventional camera. Examples include high definition, high dynamic range, and high frame rate video to provide a more realistic user experience. Stereo and multiview increase the experience of the Three-Dimensional (3D) perception by simulating depth and coming closer to presenting the real-world experience. Finally, Light Field (LF) technology enables us to capture richer directional cues. With the recent availability of hand-held LF cameras, it is possible to capture a scene from various perspectives with ease at a single exposure time, enabling new applications. For example, the fast synthesis of novel views allows a smooth transition from different perspectives to provide a more natural visual experience. Rich LF data sample light rays of a scene and allow various manipulations after the capture. The manipulations include a change of perspective, focusing at different depths in the scene, and editing depth-of-field, from narrow fields to extended depth-of-field.

Whereas the new imaging model increases frontiers of immersiveness, quality of experience, and digital photography, it generates huge amounts of data demanding significant storage and bandwidth resources. To overcome these challenges, LFs require the development of efficient coding schemes. Although efficient standard techniques for compression do exist, these methods are not as efficient in the novel structure as in the case of traditional imagery. The increasing presence of immersive data and the lack of effective coding schemes have motivated the high delivery of research works on LF coding, which recently culminated with the development of a standard called JPEG Pleno.

Deep learning is extremely efficient in learning the finest features in underlying data. The efficiency has been demonstrated across many fields in recent years. From regression to classification tasks, deep learning approaches swiftly outperform conventional methodologies, which were accurately and thoughtfully designed over many years. Nevertheless, there is still a place for improvement and to explore new fields. This especially stands in the case of LF processing, where deep learning brought novelty across various processing applications, including depth estimation, spatial super-resolution, angular super-resolution, and compression.

This thesis aims to explore deep-learning-based approaches for LF compression. We propose a hybrid coding scheme that combines a learning-based compression approach with a traditional video coding scheme for lossy compression of LF images and a learning-based method for lossless LF compression. Moreover, we investigate distributed source coding paradigm for LF compression.

1.1 . Motivation

The following subsections motivate each part of the work, including learning-based methodologies for lossy, lossless, and distributed LF compression.

1.1.1 . Autoencoder-based lossy compression

Autoencoders are neural networks trained to reproduce their input at their output. They consist of an encoder that creates some representation and a decoder that reconstructs the input from the representation. Usually, a constraint is set, which prevents learning to reproduce the input perfectly and forces to learn essential features in data. This design has been recently proposed for lossy image compression. A new component is added to the autoencoder, which models the probability distribution of the representation. The estimated probability distribution allows computing the bit cost of the representation. Bit cost is minimized with the reconstruction distortion, allowing learning parameters of a lossy codec that operates at a single point on the rate-distortion curve. The learning framework is attractive as it offers to learn the parameters of the entire coding system in an end-to-end fashion. In contrast, traditional codecs would require manually designing the framework and independently optimizing different blocks.

Autoencoder-based image codecs reached state-of-the-art performance in just a couple of years which is a fantastic development considering the journey of standard image coding tools. Naturally, the development motivated exploring this architecture for LF image coding. Prior to our work, autoencoders were not explored for LF coding.

1.1.2 . Deep autoregressive models for lossless compression

Autoregressive models consider previous samples of a sequence to model current samples. Recently, these models were integrated into deep learning methods to model the distribution of natural images. Compared to, e.g., generative adversarial networks, which are also generative models, the autoregressive models explicitly model the underlying data distribution. Moreover, as the joint distribution is factorized in a product of conditional distribution, it provides tractable likelihoods. Deep learning methods based on autoregressive models such as PixelRNN and PixelCNN [99] demonstrated state-of-the-art results for natural image modeling. The estimated likelihood's tractability and superior performance make these approaches perfect candidates for lossless compression. These methods have already been proposed for the lossless image coding task, resulting in competitive performance compared to standard coding tools.

Given the potential of these models, an obvious question about their utility arises. We first consider these models for lossless compression of LF images.

1.1.3 . Deep distributed source coding

Distributed source coding is an unconventional coding paradigm that allows flexibility in distributing computational resources (complexity) between an encoder and a decoder. It was proposed to push most computation complexity to the decoder side to operate on acquisition systems with limited resources. This paradigm has already been proposed for LF coding, yet the domain was considerably unexplored compared to the work done in, e.g., distributed video coding.

Motivated by the prospect of advancing the research in distributed LF coding and high modeling capabilities of deep learning methods, we consider improving critical blocks in distributed coding schemes.

1.2 . Objectives and contributions

The transmission capacity and storage resources limit the amount of information to be preserved. The problem gets further exaggerated for LF contents due to the vast amount of information. Lossy compression allows preserving the most important characteristics under limited transmission conditions. Lossless compression facilitates prediction mechanisms to obtain compact input signal representation effectively. Finally, distributed source coding operates efficiently under limited computational resources on the acquisition side. This thesis aims to explore deep learning methods for the efficient coding of plenoptic contents for these tasks.

We observe the following limitations in traditional coding schemes for LF coding:

1. Traditional coding schemes rely on hand-designed processing blocks based on some observed heuristics.

2. Each functional block is usually optimized independently, which limits the performance of the overall processing pipeline.

3. Lack in data-driven approaches can discover delicate cues and understand relations between them.

In this thesis, the following objectives are set:

1. **Lossy LF coding.** To explore the utility of end-to-end trained data-driven approaches for LF coding.

2. **Lossless LF coding.** To investigate a design of the coding methodology and learning-based alternatives for conventional coding blocks.

3. **Distributed LF coding.** To investigate learning-based alternatives to standard processing blocks in distributed schemes.

We define objectives on a high level, while detailed objectives and research questions are presented at the beginning of each chapter describing a particular methodology.

The contributions of the thesis are described below:

- Proposal of an end-to-end coding scheme for lossy LF compression.
- Proposal of a hybrid coding scheme for lossy LF compression.
- Introduction of an autoregressive model for LF coding.
- Design of a coding scheme of lossless LF compression.
- Evaluation of a learning-based view synthesis method in the conventional and distributed coding scenario.

1.3 . Thesis outline

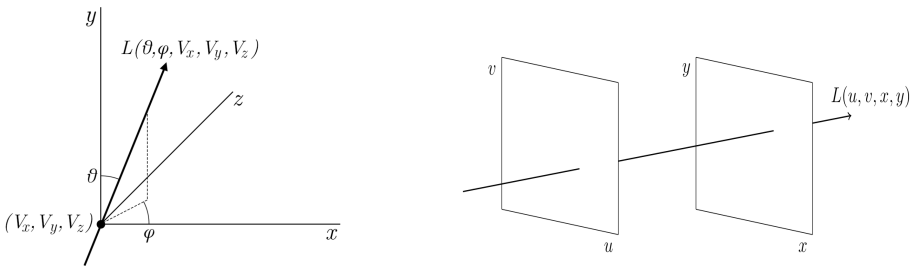
The thesis is organized as follows. Chapter 2 introduces LFs and important tasks in LF imaging. Chapter 3 provides an overview of LF coding technologies, and it includes the solutions adopted in the recent JPEG Pleno standard. Chapters 4, 5 and 6 describe the proposed schemes for lossy, lossless and distributed LF image compression, respectively. The final chapter, Chapter 7, concludes the thesis with the summaries of proposed works and prospects.

2 - Light field imaging

In this chapter, we present an overview of LF imaging technologies. These include acquisition devices and different ways to represent LF data. We mention a standardization initiative that pushed the development of LF coding methods. We conclude the chapter with rendering procedures, display technologies, and methods for quality evaluation.

2.1 . Light fields

To represent the light, Adelson and Bergen used a seven-dimensional function called a plenoptic function [2]. The plenoptic function describes the radiance of every light ray in the scene at a position in space (V_x, V_y, V_z), a direction (ϑ, φ), a wavelength λ while it propagates in time t, $P(\vartheta, \varphi, V_x, V_y, V_z, \lambda, t)$. The plenoptic function is high-dimensional and, thus, cumbersome to capture, process, and visualize, but by making assumptions based on the desired application, it can be simplified. By focusing only on static scenes, the time dependence can be excluded. Furthermore, by limiting the range of wavelengths based on the human visual system, the dimensionality can be reduced to the range of visible light. These constraints lead to the reduction of the plenoptic function to three five-dimensional functions $L(\vartheta, \varphi, V_x, V_y, V_z) = P_c(\vartheta, \varphi, V_x, V_y, V_z)$, where $c \in R, G, B$ denotes a color channel (Figure 2.1a). Levoy and Hanrahan suggested that in a space without occlusions, the radiance of a light ray would stay constant as there are no objects that could interfere with its propagation, making it possible to omit an additional dimension in the plenoptic function [75]. They proposed to define light rays as intersection points between rays and two parallel planes, as illustrated in Figure 2.1b. An LF, then presents a set of light rays parameterized by a Four-Dimensional (4D) function $L(u, v, x, y)$, where each light ray is oriented from a point at the uv plane to a point at the xy plane and, therefore, is called Two-Plane Parametrization (2PP).



(a) Representation in five dimensions.

(b) Representation in four dimensions, i.e., 2PP.

Figure 2.1: Plenoptic function.

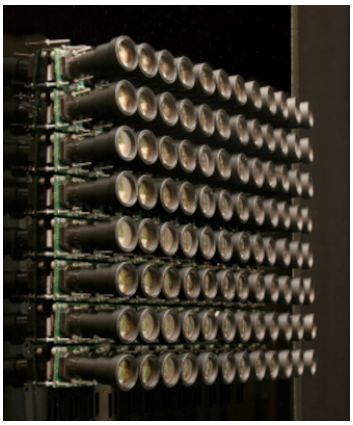
The uv plane can be thought of as a plane where a light ray enters a camera and defines directional or angular information of its propagation, while the xy plane denotes the plane at which the light ray impinges the sensor or the projected spatial position of an object from which the light ray was reflected.

2.2 . Light field acquisition

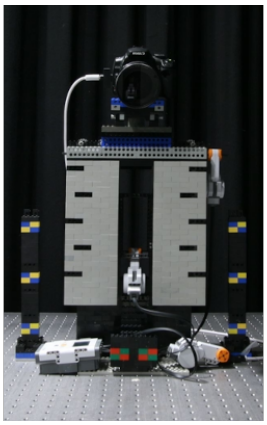
Following the 2PP to interpret the LF, a 2D image can be envisioned as a slice of 4D LF at angular coordinates (u, v). Therefore, by collecting images from different viewpoints, it is possible to capture the LF of the scene. Nowadays, LFs are usually captured with an array of cameras [138, 147, 152], a gantry [75, 146] or plenoptic cameras [98, 107].

2.2.1 . Multi-camera array

An array of cameras utilizes multiple cameras arranged on a grid to capture a scene from multiple positions in space. An example of a multi-camera array is illustrated in Figure 2.2a.The capturing arrangement makes mapping to LF straightforward as every camera denotes a single viewpoint of a scene (u, v) while pixels in each image are defined by spatial positions (x, y). However, an array of cameras is bulky, and the cameras need to be synchronized.



(a) A multi-camera array [147].



(b) Lego Mindstorms gantry [1].

Figure 2.2: LF acquisition devices based on conventional camera(s).

2.2.2 . LF gantries

A simpler approach to capture an LF image is depicted in Figure 2.2b where a camera is mounted on a motor that allows moving camera in different directions. The drawback of gantry approaches lies in their sequential operationality, which allows for capturing only static scenes. Nevertheless, gantries can capture highly dense LFs by using a fine control of the camera movement.

[1] http://lightfield.stanford.edu/acq.html

2.2.3 . Plenoptic cameras

Plenoptic cameras (also called lenslet or LF cameras) are easier to manage: they are operated like any other conventional camera, and the synchronization is inherently supported. Conversely, the captured data is stored in a less intuitive way; spatial and angular coordinates are interleaved, which demands further processing to produce a representation that can be used by existing processing algorithms. There are two types of plenoptic cameras commercially available: an unfocused plenoptic camera manufactured by Lytro [2] and a focused plenoptic camera built by Raytrix [3]. Examples of plenoptic cameras are shown in Figure 2.3.

(a) Lytro Illum camera [4].

(b) Raytrix camera [5].

Figure 2.3: Plenoptic cameras.

The unfocused plenoptic camera

The unfocused plenoptic camera was designed by Ng in his doctoral dissertation [98]. The camera utilizes a lenslet array (also called lenslets or a microlens array) positioned in front of the camera sensor to capture directional information of the incoming light rays. By design, the main lens is focused at the microlens array plane and brings a sharp image of objects at the focal plane to it. Each microlens defocuses the converged light, and the sensor behind a microlens collects the defocused rays and captures a so-called Micro Image (MI). Figure 2.4 illustrates the optical design of the unfocused plenoptic camera. Light rays reflected from a point in space are converged to a microlens which then propagates the incoming rays to the sensor behind it, allowing to capture the light intensity and the direction of the incoming light. Each pixel of a MI captures a single angular component.

Compared to the conventional camera, all the light emitted from a point in the real world is not integrated at a single pixel. Rather it is distributed by a microlens and measured by pixels behind the microlens. Referring back to the 2PP, the microlens plane denotes spatial coordinates as each microlens captures the light reflected from a single spatial position while the sensor plane represents

[2] https://en.wikipedia.org/wiki/Lytro
[3] https://raytrix.de
[4] https://www.xcite.com/lytro-illum-40-megarays-light-field-digital-camera-black.html
[5] https://raytrix.de/

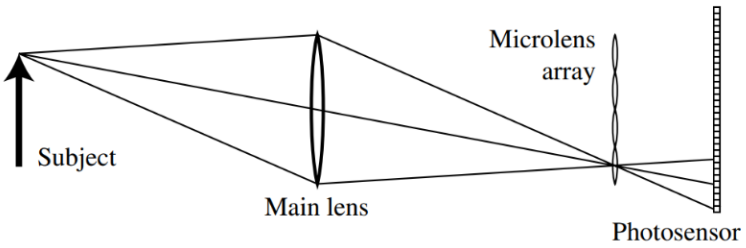
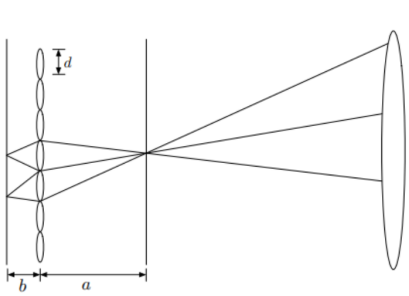


Figure 2.4: Optical design of the unfocused plenoptic camera [98].

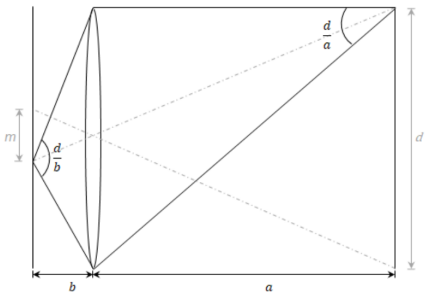
angular coordinates. The observation also suggests the main drawback of the optical design: the limited spatial resolution that corresponds to the total number of microlenses in the array.

The focused plenoptic camera

Lumsdaine and Georgiev [81] proposed a new type of plenoptic camera with an aim to increase the spatial resolution of the traditional plenoptic camera. The new optical design, presented in Figure 2.5a, sets the image plane of the main lens and the lenslet plane at different positions. The lenslets with the focal length f, are positioned at a distance b from the sensor and at the distance a from the image plane in order to satisfy the lens equation $\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$. The generalized design of the plenoptic camera allows setting the trade-off between spatial and angular resolution by changing the distances a and b. Focusing the lenslets to infinity and positioning the sensor at their focal plane (i.e., $a \to \infty$ and $b \to f$), the design of the unfocused plenoptic camera is achieved, and the maximal angular resolution is available at the cost of spatial resolution. On the contrary, moving $a \to 0$ and $b \to 0$, the design of the conventional camera is achieved, and the full spatial of the sensor is utilized while angular resolution diminishes.



(a) Formation of the main lens and the microlenses [81].



(b) The capturing of a single microlens.

Figure 2.5: Optical design of the focused plenoptic camera.

In the new design, the lenslets are focused at the image plane, and light rays

from various points from the image plane are propagated by a single lenslet. Compared to the unfocused plenoptic camera, which captures a single spatial point per lenslet, the lenslet images in the focused plenoptic camera capture more than one spatial point resulting in an increase in the spatial resolution. On the other side, the increase in the spatial resolution in lenslet images comes at the cost of a decrease in the angular resolution. Considering a span d of the image plane bounded by two gray discontinuous lines in Figure 2.5b, it can be noted that the sensor records narrower span m. The reduction origins from the camera's geometry, and it is defined by the ratio $\frac{a}{b}$ following the equation for similar triangles. Furthermore, the same span d is captured by neighboring lenslets in different directions showing that the lenslet array acts as a directional sampler, conversely to the unfocused plenoptic camera where the lenslet array is utilized for spatial sampling. As a single-pixel covers a span of $\frac{d}{a}$ in angular dimension and a single lenslet covers span $\frac{d}{b}$, it means that there are $\frac{a}{b}$ different directions. As the ratio $\frac{a}{b}$ governs both spatial and angular resolution, it can be noticed that for the lenslets defined by the focal length f, the total spatio-angular resolution stays constant as an increase in one resolution results in the decrease of the same level in the other resolution.

Finally, the resolution of the focused plenoptic camera does not depend on the number of lenslets. The geometry of the camera defines the span of the pixels at the sensor, which captures an object in front of the lenslet. Because a single lenslet image captures span $d \cdot \frac{a}{b}$ which goes beyond the boundaries of a single lenslet, there is an overlap in the spans which are captured by multiple lenslets (which is at the origin of the directional information). As a result, the increase in the effective spatial resolution depends on the resolution of the sensor and the overlap.

Trade-offs

The two plenoptic cameras are very similar to each other as both utilize a lenslet array to capture directional information. However, the geometry inside the camera brings a fundamental difference between the cameras. The unfocused plenoptic camera exhibit fixed spatio-angular resolution due to the positioning of the sensor at the focal point of the lenslet array. Conversely, the focused plenoptic camera allows for trade-off between spatial and angular resolutions by positioning the lenslet array at different distances from the sensor and the image plane. The spatial resolution of the traditional plenoptic camera is defined by the number of lenslets, while in the focused plenoptic camera, the geometry of the camera defines the resolution.

2.2.4 . Summary

This section presents three ways of capturing LF data: multi-camera arrays, gantries, and plenoptic cameras. Multi-camera arrays capture wide baseline LF data as the distance between cameras is constrained by the physical size of the cameras. Gantries allow flexible angular sampling via a configurable movement of a mounted camera. Yet, due to the sequential sampling, they are applicable only to static scenes. In converse to previous systems, plenoptic cameras are portable and convenient to use. The insertion of the microlens array provides captures of a narrow baseline. Still, in plenoptic cameras, there is a trade-off between angular and spatial resolution as a single sensor is shared to record both information. Although not explicitly mentioned, LFs can be synthetically created using computer-generated models.

2.3 . Light field representation

The 2PP offers an intuitive explanation of LF structure as it describes the intersection of light rays with two planes. Yet, the parametrization is 4D which is difficult to visualize and not clear how to employ. A remedy to this challenge is the observation of the underlying data along with a subset of dimensions. Typically, 2D slices are extracted, which raise three distinct representations based on the pairs of dimensions that are gathered. These include lenslet or MI representation obtained by fixing spatial coordinates and collecting rays at angular coordinates, Sub-Aperture Image (SAI) or viewpoint representation obtained by collecting all rays from an angular coordinate, and finally, Epipolar Plane Image (EPI) representation where a spatio-angular pair is fixed while the intensities of related rays are collected. These representations will be described in more detail in the following from the perspective of acquisition systems. Moreover, we mention other representations which are employed in LF processing schemes.

2.3.1 . Lenslet representation

Lenslet representation is the raw image representation recorded by a sensor in a plenoptic camera. It consists of a grid of MIs, each of which is recorded by a sensor area behind a corresponding microlens. Depending on the optical design of the plenoptic camera and microlens array, the captured image has a different appearance. E.g., in the unfocused plenoptic camera, the microlenses are focused at the main lens; therefore, MIs capture the image of the back of the main lens. On the other side, in the focused plenoptic camera, microlenses are focused at the image plane of the main lens, and thus, each MI captures a portion of the image plane. Moreover, the microlenses can be positioned on an orthogonal or hexagonal grid which, naturally, impacts the appearance. We also call this representation a MI-based representation or simply a lenslet image.

2.3.2 . Viewpoint-based representation

The representation which is more easily understood is the viewpoint-based representation, as it consists of a grid of images depicting a scene from different perspectives. The viewpoint-based representation is a natural representation of LFs captured with a multi-camera array or a gantry. The captured images are arranged on a 2D grid based on the corresponding camera position at the capture time.

LFs captured with plenoptic cameras can also be represented in this format. However, additional processing is necessary. In particular, it is needed to extract so-called SAIs from the raw representation.

Considering the design of the unfocused plenoptic camera, where the lenslet array defines spatial resolution, and MIs capture directional information about the light incident to the lenslets, the number of SAI corresponds to the resolution of MIs. A SAI can be generated by extracting pixels from the same position at each MI and tiling them together. Since lenslets are extremely small compared to the main lens, their focus on the main lens is equivalent to being focused at the infinity. Consequently, if a light ray is traced back from a single pixel, it can be noticed that it originates from the light passing through a part of the main lens aperture . Furthermore, the light propagating through the same part of the main lens aperture is collected by pixels at the same relative position in all lenslets . As a result, the collection of these pixels generates a picture of the scene seen through the part of the aperture. Selecting a different relative position of the pixels would generate a view with a different perspective. A decoding procedure usually precedes the SAIs extracting, which resamples the lenslet image to compensate for potential rotations of the microlens array and to align it with the sensor grid. Furthermore, when a hexagonal microlens array is used, an additional transformation is applied to convert the hexagonal-grid MI representation to the one with the orthogonal grid [27].

Following the same procedure for the images captured with a focused plenoptic camera gives images with strong artifacts. To overcome this issue, Georgiev et al. [42] presented a basic rendering algorithm for "full-resolution" rendering of a single viewpoint image. Instead of extracting a single pixel from each MI, as it is done with the unfocused plenoptic camera, a patch is extracted and tiled with other patches.

Although the algorithm provides images of higher resolution compared to the ones obtained with the unfocused plenoptic camera, the images exhibit strong artifacts. The artifacts appear due to the employment of the same patch size for all depths in the scene. The approach computes the patch size based on the distance between the image plane and the lenslet array, which corresponds to the depths in the scene which are in focus. For objects at other depths, the patch size should change based on their distance from the camera. For example, an object closer to the camera needs to be extracted using a larger pitch. This can be

understood by envisioning an object positioned between the lenslet array and the image plane, as illustrated in Figure 2.6. The two objects have the same size but are positioned at different depths in the scene. As the object moves from the image plane toward the lenslet array, the ratio ab tends to 1, and the sensor captures an area that tends to the area of the object. Therefore, in order to render the closer object without artifacts, the pitch has to be increased.

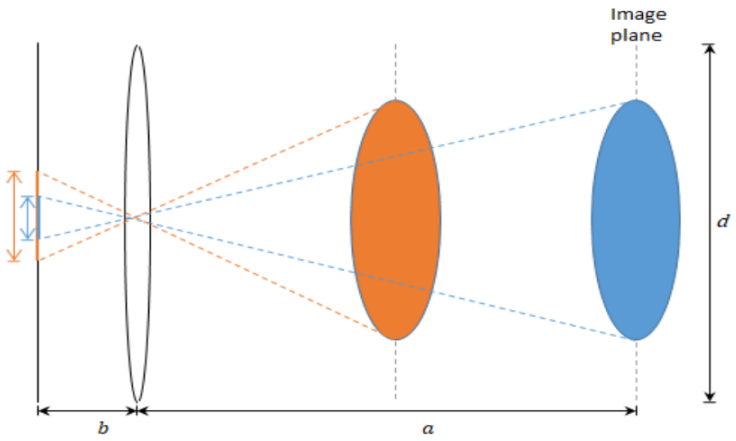


Figure 2.6: Relation between an object's depth object and the size of its recorded image.

Therefore, the basic rendering approach is not suitable for scenes with objects at various depths. In order to cope with the appearing artifacts, Georgiev et al. [42] propose two more approaches: depth-based rendering and blended rendering, which improve the quality of extracted views.

2.3.3 . Epipolar plane images

Previous representations consider spatial and angular dimensions separately from the perspective of elementary blocks, e.g., a MI can be selected by fixing spatial dimensions while its information varies across angular dimensions. Similar behavior stands for SAIs. An EPI is a 2D slice of a 4D LF with a pair of horizontal (vertical) spatio-angular dimensions fixed while the remaining vertical (horizontal) dimension vary. A horizontal EPI $EPI_h(u, x) = L(u, v, x, y)$ is created by vertically stacking rows y from SAIs in angular row v. In an analogous manner, the vertical EPIs can be created. The EPI representation is procured by stacking EPIs on a 2D grid.

EPI are characterized by homogeneous regions bounded with straight lines. The lines appear under different slopes, which is proportional to the object's depth [16]; an in-focus object is characterized by vertical lines, suggesting that the object does not displace along with different views while the objects further away from the focal plane will be characterized by lines with greater slopes.

2.3.4 . Other representations

It is also worth mentioning representation derived from preparing an LF for a particular processing scheme. The most notable schemes are standard video coding tools and their multiview extensions. Three representation rise: Pseudo-Video Sequence (PVS), multiview-PVS, and geometry-assisted representations.

The PVS presents a representation acceptable by a video coding tool such as High Efficiency Video Coding (HEVC). The generation procedure adopts a particular pattern to scan a grid of MIs or SAIs, and the traversed views are arranged along a new (temporal) dimension to create a video sequence. The adjective "pseudo" indicates that it is not a natural video sequence; rather particular processing generates it.

The multiview-PVS is employed to leverage the multiview extension of Advanced Video Coding (AVC) or HEVC, allowing to better exploit the inter-view correlations compared to original variants of the coding tools. This representation considers a full-parallax LF image as a multiview video sequence by accepting one of the angular dimensions as the time dimension.

Last but not least, an LF image could be supported with geometry information. The geometry could be disparity or depth maps, or other models facilitating relations between views. This representation allows the synthesis of novel views given a sparse set of views and attached geometry information. In particular, it shows a great prospect from the coding perspective as a reduced number of views can need to be encoded while the remaining views can be generated at the decoder.

2.4 . Light field compression

As mentioned before, LFs record not only the spatial coordinate of the incident light ray but also its angular orientation. The richer capturing provide new functionalities, yet they come at the cost of huge amounts of captured data which present a major challenge for storing or transmitting this information. E.g., a typical LF image captured by a Lytro Illum camera offers only a 0.25-megapixel resolution although occupying about 218 megabytes of hard disk space (decoded LF images with 15×15 views, 10-bit precision, and three color channels). Given the sheer amount of data this acquisition entails, LF coding is essential to practical applications and is considered as an important research topic.

Considering the acquisition devices and formats presented in the previous sections, LFs could be encoded with already available standard coding tools. We already mentioned this scenario when discussing additional representations of LF images. Nevertheless, these standard coding tools were not designed for full-parallax LFs, so they might not be optimal for the new visual modality, and, in the case of lenslet images, they cannot exploit the correlation between MIs optimally. This situation motivated the exploration of an adaption of available coding technologies and the design of novel LF coding schemes.

More recently, a strong drive for the development of LF image coding technologies can be attributed to a standardization project called JPEG Pleno. The project is motivated by the vision of the committee following the increasing presence of novel immersive technologies. New acquisition technologies allow capturing depth-enhanced, omnidirectional and LF, point cloud and holographic contents characterized with richer cues. Moreover, this data introduced novel applications and visualization, which were difficult or impossible to achieve with traditional imagery. As a result, a new processing pipeline emerged, including new variables such as rendering and interactivity that need to be considered in addition to efficient decoding for the wide adoption of the novel technologies and applications they bring. JPEG Pleno proposes to unify all these technologies by starting from the origin of all these technologies, the light, and the model which describes underlying information, the plenoptic function. Revolving around these ideas, JPEG Pleno aims at deriving a representation framework that provides, in addition to efficient coding tools, support to advanced methodologies for image manipulation, interactivity, random access, and others supporting emerging applications and services [37].

JPEG Pleno initiative organized two LF coding challenges [58, 59] with the aim of collecting the best available solutions at the time. Proposed solutions are evaluated and compared under the same testing conditions, which are also distributed to encourage further benchmarking of state-of-the-art methods. Perceptual quality was considered the most important criterion when choosing the best solutions.

The first grand challenge was organized at the IEEE International Conference on Image on Multimedia and Expo (ICME) in 2016, and it asked for efficient LF compression solutions as alternatives to existing JPEG standards for contents captured by an unfocused plenoptic camera. Quantitative performance analysis of the collected solutions showed that it is possible to do much better compared to (conventional image-based) JPEG anchor by designing schemes that consider LF nature effectively [143].

The second grand challenge was organized at the IEEE International Conference on Image Processing (ICIP) in 2017. In addition to LF solutions for coding of the plenoptic content, participants are invited to provide solutions for contents captured by a High-Density Camera Array (HDCA). The overall results of the challenge demonstrate that there is much to gain compared to the direct application of video coding tools by designing a methodology that considers LFs structure [142].

Following the two grand challenges, JPEG Pleno provided LF test contents described in Common Test Conditions (CTC) [102]. These datasets were selected to provide diverse data based on different acquisition technologies, scene geometry, spatial resolution, the number of views, etc. Furthermore, the test material is selected using an appropriate descriptor, so-called Geometric Space-View Redundancy descriptor [103], which asserts the geometric diversity of the LF images. Four lenslet LF images were selected from Lytro Illum dataset [110], specifically *Bikes*, *Danger de Mort*, *Pillars Outside* and *Fountain & Vincent*. For simplicity, in

the later text, we will use only the first word of LF content names, e.g., *Danger* instead of *Danger de Mort*. Two synthetic LF images, *Greek* and *Sideboard*, are included from synthetic HCI HDCA dataset [146]. Two LF images captured by gantries; *Tarot Cards* captured using a Lego Gantry as a part of Stanford HDCA dataset [131] and *Set2 2K sub* from Fraunhofer dataset [39, 153]. The last LF content, *Poznan Laboratory 1* was acquired using a 2D array of cameras [34, 33]. Figures 2.7, 2.8, and 2.9 present center views of JPEG Pleno test LF images.

(a) *Bikes*. (b) *Danger de Mort*. (c) *Pillars Outside*. (d) *Fountain & Vincent*.

Figure 2.7: Center views from lenslet LF images.

(a) *Set2 2K*. (b) *Poznan Laboratory 1*. (c) *Tarot Cards*.

Figure 2.8: Center views from HDCA LF images.

(a) *Greek*. (b) *Sideboard*.

Figure 2.9: Center views from synthetic LF images.

Besides test data proposed by JPEG Pleno, alternative datasets could be of interest for other processing tasks. E.g., de Faria et al. present a LF image dataset of skin lesions captured with a Raytrix camera [30] that could be of interest for lossless LF compression as it presents contents dedicated to medical purposes. Similarly, high dynamic range LF contents have not been explored much, although they offer a more immersive visual experience [70].

In Chapter 3 we will overview LF coding technologies, including proposed solutions of the JPEG Pleno standard.

2.5 . Light field rendering and display

LF rendering is a process that generates appropriate data that can be visualized on an available display. Traditional 2D contents can be utilized after decoding without the need for additional processing. On the other side, due to their rich structure, LFs need to be processed before visualization in order to extract desired content. There are various ways to manipulate captured contents and obtain new functionalities that were not available in conventional systems, such as:

- *Viewpoint change* allows generating a view of a virtual camera. The new view can be generated using image-based rendering [75, 44] or DIBR methods depending on the format of the input data. In the former approaches, captured light rays are interpolated to estimate novel light rays, while in the later approaches, available views are spatially displaced based on their disparity (the inverse of depth) information.

- *Focus change (Refocusing)* allows rendering an image of a scene at a different focal plane than the one originally captured. The simplest way to achieve the change of focus is by using the shift-and-sum algorithm where LF views are displaced for a fixed disparity value scaled proportionally to the relevant position of a view with respect to, e.g., the center view and averaged together.

- *Depth of field control* allows changing the depth of field after the capture. The depth of field is governed by the camera aperture; the wider the opening, the narrower depth-of-field gets, and vice versa. With conventional cameras, the obtained extended depth-of-field or all-in-focus images, it was necessary to sequentially capture an image with different focal planes and then to combine the in-focus regions [3]. With LF the similar processing is in order, but the scene is captured only once. Moreover, with the unfocused plenoptic camera, the extracted SAIs are already all-in-focus as the aperture of each is very narrow, although they are noisy.

The rendering of LFs is closely related to the content visualization. In fact, it is a display that controls the rendering process to create the best possible user experience given the characteristics of the input data and the display itself. In the early days of the development of LF applications, it is common to visualize these contents on conventional 2D displays, where the rendering has a vital role in preparing a single viewpoint of the scene. On the other side, LF displays and Head-Mounted Displays (HMDs) can provide more immersive experience. LF displays use captured LFs to create a replica of the original field of light and allow depth perception with full-motion parallax without the use of glasses and consequently without discomfort. The HMDs are used in Virtual Reality (VR) and Augmented Reality (AR) applications to provide a strong sense of immersion. VR supports

various real-life applications, such as education, healthcare and travel, using various headsets (PlayStation VR[6], Oculus[7], Samsung Gear VR[8]). AR applications use special glasses or a headset to insert virtual content into the real-world content. On the other hand, as seen, the Pokemon Go AR mobile game [9] does not necessarily require a particular device to run; rather, a smartphone suffices.

2.6 . Light field quality evaluation

LF quality evaluation is a complex task due to a great diversity in the range of available acquisition techniques, distortion types, and rendering methods. To give an idea, we have already mentioned four ways to capture LF images: plenoptic cameras, multi-camera array, gantries, and synthetic generation, while from a rendering aspect, it is possible to evaluate independent images, refocused contents, PVSs generated following various scanning order or based on display types. Moreover, objective and subjective procedures need to be considered. JPEG Pleno has started a new standardization effort on the quality assessment of LF images that aims to identify the test data, investigate rendering procedures, provide guidelines for the evaluation procedures, and define potential use cases [61].

During the development of this thesis, we have followed the guidelines proposed by JPEG Pleno for the purpose of the evaluation of LF coding technologies. JPEG Pleno proposed CTC which would allow fair comparison between different coding methodologies. Besides test contents, the CTC include precise test conditions and evaluation metrics which are presented in e.g. [102]. CTC recommend using two objective quality metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [145]. The metrics are used to measure the distortion between each pair of original SAI and its processed variant in YCbCr color space, and then the measures are averaged across the views. In addition, Bjøntegaard measures are recommended to compute numerical averages between Rate-Distortion (RD)-curves.

[6] https://www.playstation.com/en-us/ps-vr/
[7] https://www.oculus.com/
[8] https://www.samsung.com/global/galaxy/gear-vr/
[9] https://en.wikipedia.org/wiki/Pokémon_Go

3 - An overview of light field coding technologies

3.1 . Introduction

In this chapter, we overview LF coding technologies. LFs enable increasing the degree of realism and immersion of visual experience by capturing a scene with a higher number of dimensions than conventional 2D imaging. Chapter 2 describes various means of capturing LFs, from a plenoptic camera to an array of camera. The captured information also offers novel applications such as refocusing and viewpoint shift. These novelties come at the cost of increased dimensionality and thus storage demand. JPEG committee acknowledged the necessity of efficient compression methods by starting the JPEG Pleno initiative to provide a standard framework for the representation and coding of plenoptic data.

JPEG Pleno organized grand challenges to collect novel LF coding solutions and evaluate them under common conditions following object and subjective quality metrics. The results of these challenges show that it is possible to exploit correlations in LF images more efficiently using solutions specifically designed to reduce redundancies in LF structure compared to standard image and video codecs.

In the following sections, we overview different coding solutions for compression of LF contents. We consider a taxonomy based on a functional part of LF coding tools that is responsible for exploiting LF correlation: transform-based and prediction-based solution as presented in Figure 3.1. Furthermore, we mention

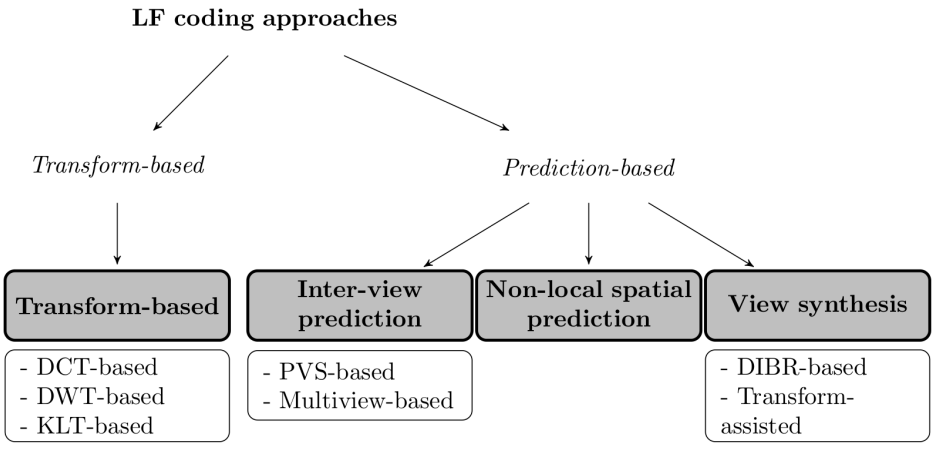


Figure 3.1: Classification of LF coding solutions (inspired by taxonomy presented in [25].

some promising coding solutions leveraging high modeling capabilities of deep learning methods and briefly describe the standard LF framework proposed by JPEG

Pleno. The chapter is concluded with an overview of lossless coding methods and a discussion on promising coding methodologies.

3.2 . Transform-based methods

Transform-based coding tools exploit correlation in LF images using some transformation. In particular, these methods decompose the input vector by representing it as a combination of basis functions that effectively compacts the energy of the input. As a result, a set of coefficients is obtained each of which measures the correlation of the input signal with a particular basis function. These coefficients are then quantized, if an application permits distortions, and entropy coded to reduce statistical redundancies. The inverse set of operations allow recovering of the input signal. A general transform-based codec is illustrated in Figure 3.2.

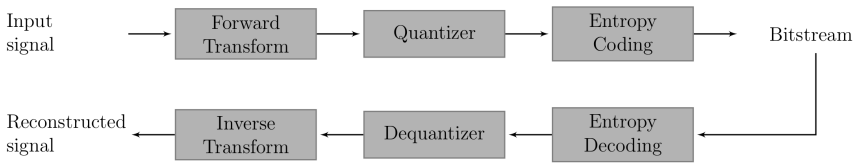


Figure 3.2: Generic block diagram of transform-based coding.

Some of the transforms proposed in the literature includes Disrete Cosine Transform (DCT) [126, 29, 31], Disrete Wavelet Transform (DWT) [84, 19, 40] and Karhunen Loève Transform (KLT) [63, 67]. Initial solutions for LF compression proposed to extend transform-oriented image coding tools such as JPEG [56] and JPEG2000 [85]. These methods employ DCT and DWT, which rely on fixed basis functions to compute the transform coefficients and obtain more compact representations of the input signal. On the other side, KLT computes basis functions for each content allowing better modeling of input data, but in addition to obtained coefficients, it needs to transmit basis functions as well. Differently from image data, LFs present 4D signals, therefore, higher dimensional transformations are typically employed e.g., 3D or 4D, for improved performance.

3.2.1 . Approaches based on DCT

DCT is well known for its application in image coding as it provides a compact representation of the original signal. The frequency representation has most of its energy compacted in a few bands, allowing it to discard high-frequency information efficiently. The utility of the transform has been demonstrated in the deployment in the standard image codec such as JPEG.

Given the success of JPEG for image coding, some methods propose to extend JPEG functionalities for LF coding. As the MI representation facilitates grid-like structure, a natural way of encoding is to apply DCT on each MI, followed by

quantization and entropy coding. This way, only correlations inside each MI are exploited, so a natural way to exploit similarities between neighboring MIs is to create a volume of MIs and apply 3D-DCT. E.g., [126] study the impact of 2D scanning patterns along MIs on coding performance. Among scanning patterns, the 2D Hilbert shows the most promise, suggesting the importance of selecting MIs in such a way that the correlation between MIs inside each 3D block is maximized.

The previous coding strategy is sub-optimal when coding 4D LF structure as it depends on a particular scanning order to exploit the correlation between MIs. Therefore, in [29] is proposed to match LF dimensionality by employing 4D-DCT to decorrelate underlying information. After quantization, the coefficients are processed on a bitplane level by grouping zero-valued coefficients using hexadeca-tree clustering. The clustering presents an alternative 4D variant of encoding position of zero coefficients, e.g., run-length coding used in JPEG.

3.2.2 . Approaches based on KLT

KLT is a data-dependent transformation, which i.e. computes transformation coefficients as well as the basis functions of the transformation for a given input [85]. Thus it is more flexible in adapting to the content compared to a transformation with fixed basis functions such as DCT. KLT has been proposed for the compression of LF images with the basis functions computed on MIs [63] or SAIs [67]. The SAIs are extracted by selecting co-located pixels for every MI. To provide a better modeling of MIs, [63] propose clustering MIs using Vector Quantization and computing and assigning a KLT basis vectors to each cluster.

In the KLT coding scheme operating on MIs, [63] experiments show that increasing the number of clusters improves overall performance. When compared to the coding scheme operating on SAIs, the former shows inferior performance as it is needed more basis vectors to obtain similar reconstruction quality. The superior performance of the latter scheme is likely contributed to a higher correlation among SAIs compared to the scheme based on MIs. Each MI captures only a small part of a scene and has different features in different parts of the scene. On the other side, SAIs contain the whole scene and differ slightly due to a change of perspective. Compared to a standard anchor, such as JPEG, both schemes perform better.

3.2.3 . Approaches based on DWT

DWT is an alternative to the formerly presented block-based transformations. It iteratively computes low-pass and high-pass representations of an input signal. The final output is a multi-resolution representation of the input signal. As such, it offers resolution scalability as well as quality scalability.

The straightforward application of DWT to LF image compression is via wavelet-based, image-coding tools, e.g. JPEG2000. This strategy is adopted for the coding of LF images captured by a plenoptic camera [49, 104]. JPEG2000 shows superior performance when compared to e.g., legacy JPEG and SPIHT codec [49], or JPEGXR [104]. However, standard image coding tools are not suited to efficiently

exploit correlation in the LFs, and a reasonable step would be to use a higher-dimensional transformation, e.g., 4D-DWT. To this end, a separable 1D-DWT is sequentially applied along spatial and angular dimensions resulting in a 4D array of wavelet coefficients and a multi-resolution representation of the LF. The multi-resolution representation allows for progressively reconstructing the LF; low-resolution SAIs can be reconstructed from low-frequency wavelet coefficients, while by including high-frequency coefficients, better quality, and higher resolution can be achieved. Thanks to this approach, it is possible to trade off rendering speed and quality to meet application demands. Nevertheless, quantitative analysis in terms of PSNR shows that this method is inferior to a disparity-compensation-based method [84], which suggests that disparity-compensation allows better correction of inter-view differences.

The lifting scheme [133] is proposed as a way to facilitate the disparity compensation in wavelet transform across views [19]. Given a set of views, two separate sets comprised of even and odd views are created. The predict stage generates the high-pass sub-band by subtracting an odd view from the disparity-compensated even view. The obtained residual is also disparity-compensated to align with the even view and added to it to obtain a low-pass sub-band. This procedure takes advantage of inter-view correlation, and it is followed by an additional step, whereas each sub-band image is processed using a multi-level 2D-DWT to take advantage of the remaining intra-view correlations. In the case of full-parallax LFs, i.e., LFs comprised of horizontally and vertically displaced views, the inter-view transformation is carried by applying the lifting scheme horizontally and vertically across the 2D view grid. In the case of LFs captured by a plenoptic camera, a disparity compensation based on a perspective transformation can be used [40].

As a more sophisticated version of the lifting scheme, Ruefenacht et al. [114] propose a hierarchical inter-view transform and a geometry model. The hierarchical lifting scheme supported by an accurate disparity estimation provided by the geometry model proves to be a highly competitive methodology compared to state-of-the-art methods on densely sampled LF images.

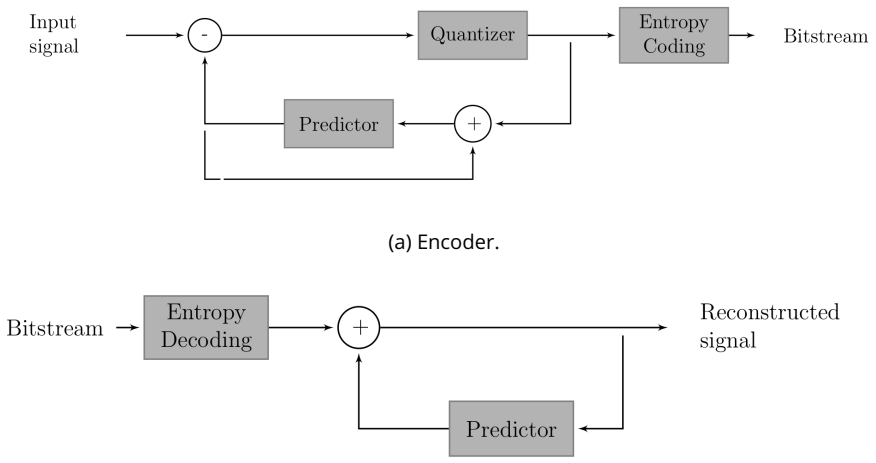
3.2.4 . Summary

Transform-based methods provide a representation of the input signal, which allows exploiting existing correlation by effectively compacting the signal's energy in a small range of frequencies. In general, transform-based methods, designed in particular for compression of LF images, outperform image-based codecs such as JPEG and JPEG2000. Furthermore, 4D transforms proved the most suitable for LF compression as they exploit the intrinsic correlation in a superior manner leading to improved RD performance. Namely, LF coding scheme based on 4D-DCT [29] shows superior performance compared to more sophisticated anchors such as HEVC coding of a LF arranged in a PVS. Still, these results stand only for lenslet data with a high correlation between views. On the other hand, 4D-DWT-based methods that leverage disparity compensation to exploit the correlation between

views provide improved performance on more challenging data, captured by gantry [114]. In addition, methods based on 4D-DWT offer various levels of scalability: quality, viewpoint, and resolution.

3.3 . Prediction-based methods

In contrast to transform-based solutions, prediction-based approaches rely on a prediction mechanism that provides an approximation of the input signal that effectively reduces its redundancy. A generic block diagram of predictive coding is presented in Figure 3.3. Three broad groups of approaches were proposed based on the representation of the input signal and the methodology used to provide an estimate: approaches based on inter-view prediction, non-local spatial prediction, and view synthesis [25].



(a) Encoder.

(b) Decoder.

Figure 3.3: Block diagram of predictive coding.

3.3.1 . Inter-view prediction

Inter-view prediction methods refer to approaches that rely on video coding tools to facilitate prediction between views in LF images. Before coding, an input LF image is converted to a representation acceptable by a video codec. Two representations can be noted depending on the used video codec. In a PVS representation, an LF image views are picked in a particular scanning order across the 2D view grid and stacked along the new ("time") dimensions. Practically, the views become the frames of the created video sequence and can be processed by, e.g., the HEVC codec. In a multiview representation, one of the angular dimensions is considered the time dimension. Then, multiview extensions of standard coding

tools such as H.264/AVC and HEVC can be leveraged to exploit correlations along spatial, inter-view, and pseudo temporal dimensions.

Some coding schemes based on the PVS representation propose to use fixed scanning orders applied across all SAIs [141, 46] or across subsets SAIs, which effectively reduces the distance between scanned views [76]. Other methods consider the content of LF images and drive the scanning order by a similarity metric to adaptively decide on the ordering of view into the PVS [22, 57]. A complementary task to previous approaches is the construction of a reference frame list which can have a considerable role in the overall performance of PVS-based methods. Namely, the scanning order does not explicitly consider the proximity of views in the reference list; instead, it follows a predefined prediction structure [90]. By setting reference frames based on their distance to a current frame, the closer views are used as reference views, and redundancies can be reduced further. A hierarchical order was also proposed as a promising solution. In [80], the central view is selected at the lowest level (coded as I-frame) while the rest of the views are organized in the following hierarchical layers (and coded as P-frames and B-frames). A rate allocation methodology can be adopted to facilitate higher quality in reference views than in the frames not used as references: I-frames should have the lowest Quantization Parameter (QP) as it serves as a reference for many views, while views deeper in the hierarchy can have higher QPs.

Another feature of the hierarchical scheme is possible viewpoint scalability and random access. The scalability allows support for legacy capturing devices, 2D and 3D/stereo. Also, progressive decoding of each layer would provide higher angular resolution as more layers are decoded. Each hierarchical layer is sequentially encoded while relying on previously decoded layers and decoded views from the current layer for prediction. Random access would improve LF navigation efficiency and reduce computational complexity and the decoding time as fewer views need to be processed before, but coding efficiency would degrade as well.

Multiview extensions have been seen as alternatives to standard video codecs allowing to exploit the inter-view correlation more efficiently. MV-HEVC shows superior coding performance compared to PVS-based coding on lenslet data, especially at low rates [4]. A nice feature of these schemes is that they can operate as generic codecs as they do not depend on geometry information and operate on both lenslet and HDCA contents.

3.3.2 . Non-local spatial prediction

Non-local spatial prediction aims to exploit similarities in MI representation. Regardless of the structure of a microlens array and the design of the optical system, captured MIs exhibit positional correlation as the content inside each MI is highly correlated and neighboring MIs also exhibit a high level of similarity due to the proximity of microlenses. Non-local spatial prediction drives inspiration from inter-prediction, which exploits the temporal correlation between frames to obtain a motion-compensated prediction for blocks in the current frame. However, as

MIs are interleaved on a 2D grid and thus present a 2D image, instead of motion vectors searched across decoded frames, spatial displacement is considered.

For LF compression, standard codecs H.264/AVC and H.265/HEVC were modified to facilitate MI prediction using the so-called Self-Similarity (SS) estimation and SS compensation blocks. SS estimation module uses block matching, in an area of already processes MIs, i.e. (the whole) causal (picture) area, to find the best match for prediction of the current MI [23]. Some variations include finding multiple candidates [24] or merging candidates [77, 64]. A standard displacement-driven template matching mechanism can be also replaced by high-order prediction models [91].

SS methods present significant potential for compression of contents captured by a plenoptic camera and, in general, show improved performance compared to Intra coding. Moreover, as they rely on prediction tools from standard video codecs, they can be easily extended to plenoptic video sequences by allowing Inter coding. However, when compared to other prediction-based coding tools, these methods are generally inferior [142, 90].

3.3.3 . View synthesis-based prediction

LF compression schemes based on view synthesis aim at exploiting high inter-view similarity by relying on scene geometry. Typically, a sparse set of reference views together with corresponding geometry information is encoded and transmitted, and at the decoder side, the rest of LF views are reconstructed using transmitted information. The literature presents two distinctive groups of approaches based on view synthesis: Depth-Image-Based-Rendering (DIBR) and transform-assisted synthesis.

Disparity compensation proved to be a potent tool in LF coding scheme. Besides radically reducing the number of references and still achieving high-quality prediction, it also offers some scalability in the coding framework. Moreover, it applies to both MI and SAI representation allowing to adapt to the desired scenario.

Li et al. [78] subsamples the LF image captured by a focused plenoptic camera by selecting every s-th MI and encodes the subsampled version using a non-local spatial prediction scheme. At the next layer, the rest of the LF image can be predicted based on the decoded LF image and estimated disparity maps used to warp the decoded part. This variant can even provide a more generalized solution with respect to the supported range of disparities; in LFs captured with plenoptic cameras SAIs have very small disparity, while in the case of HDCA, disparities are larger between views [10]. Views are divided into hierarchical levels with textures and disparity maps of the lowest hierarchical level being encoded independently while the rest of the views are processed using warping and merging of warped views. Certain views and their disparity maps from previous hierarchical levels are needed to predict a view at some hierarchical level. The adoption in the JPEG Pleno standard acknowledges the potential of this pipeline.

In addition, a standard coding tool shows great prospects in LF coding so-

lutions. Originally, the 3D extension of the HEVC codec was proposed to more efficiently compress the video-plus-depth format. For the LF compression, Huang et al. [55] propose a multiview plus depth architecture based on the 3D-HEVC. Columns of views are organized in video sequences, and computed depth maps are assigned to these sequences. Columns are sampled in a uniform step to select the reference set of views that is encoded using 3D-HEVC. The rest of the columns are synthesized using the DIBR technique.

As an alternative to DIBR-based approaches, transform-assisted approaches exploit sparseness in the Fourier domain. The spectrum of the entire LF is recovered from the spectrum of a subset of initial samples. Differently compared to the transform methods presented in Section 3.2 where the entire LF is transformed in a sparser representation and encoded, a sparse representation is recovered from a limited set of samples. It is assumed that LFs have a sparse representation in the angular domain, so the entire LF can be recovered from a limited set of views.

[32] presents an iterative, hierarchical scheme for compression of LF images based on Fourier Disparity Layers (FDL) [74]. An initial set of reference views is encoded using HM implementation of the HEVC codec by arranging views in PVS. Then, following the decoding, reference views are used to construct the FDL model. The FDL model consists of a set of Fourier spectrums wherein each spectrum corresponds to a view consisting of regions of only the same disparity. The spectrum of an LF image can be recovered as a sum of shifted FDL. The construction of the FDL model starts with the calibration phase, which estimates the angular positions of the input views and the set of disparities for which FDL will be constructed. Then, coefficients of the FDL model can be computed and used to predict the remaining views.

[5] propose a coding approach based on Shearlet Transform (ST). In ST each EPI is transformed in the Fourier domain and filtered using a set of pass-band filters designed to remove aliased components in the frequency domain. The proposed scheme starts by decimating an input $N \times M$ LF image along both angular dimensions and organizing the remaining views in a multiview pseudo-video sequence which results in M sequences N frames long. The sequences are encoded using MV-HEVC. A reconstruction algorithm based on ST predicts the decimated views from the decoded sequences.

In [144], a solution based on the graph learning approach is proposed to estimate the disparity between the views in LF. Based on the observation of strong smoothness between neighboring views in a LF image, graph learning is used to model the relation between views. Each view is considered a vertex in the graph, and the edges that model the relations between views/vertices are learned from underlying data. The graph is encoded in a lossless manner and transmitted with a set of reference views. The remaining views are reconstructed at the decoder by solving an optimization problem.

Like traditional DIBR approaches, deep learning methods were employed to

leverage their high modeling capacity in generating novel views using a sparse set of views and geometry information. In addition, some works have shown that geometry can also be estimated at the decoder side, i.e., not transmitted nor estimated from original images, and still provide high-quality reconstruction. These methods typically adopt a similar coding framework, whereas a set of reference views is selected and encoded, while at the decoder side, the remaining views are predicted from the decoded reference views.

In [149] a checkerboard pattern is adopted to select the views to be coded and the views to be generated. Authors propose a CNN which takes four input views (luminance) arranged along channel dimensions and predicts a view in-between. As the quality of synthesis operation depends on the input quality (affected by coding artifacts), the authors also propose an enhancement network that reduces the artifacts before synthesis. As different methods use different reference view arrangements, and some might perform better than others under different conditions, [12] evaluate the implications of deep learning and optimization-based view synthesis. The experiments show that the combined effort can improve overall performance on some contents compared to scenarios, whereas either approach is independently used. Similarly, some methods include the residual coding of views predicted by a view synthesis block. [50] propose using the view synthesis method of Kalantari et al. [66] to estimate the non-reference views. The predicted views are subtracted by the original views, and obtained residual frames are encoded by a PVS-based approach. Like other methods in this group, the scheme shows improved coding performance at low rates. However, it saturates towards higher bitrates.

3.3.4 . Summary

Predictive coding methods use an approximation of the input signal to reduce correlation in the signal effectively. Among prediction-based methods, schemes leveraging video coding tools' rich and powerful apparatus appeared to be extremely popular. Their versatility and efficiency, coupled with various ways of capturing and representing LF data, offered a vast spectrum of possibilities. Clearly, schemes based on the recent video coding standard such as H.265/HEVC show high competitiveness. Moreover, multiview-based coding solutions show state-of-the-art results in recent studies and are highly attractive as they could be easily extended for a video scenario. Methodologies based on non-local spatial prediction extend standard video coding tools to exploit the intrinsic structure of MI representation. Motion estimation and compensation are leveraged for prediction in the interleaved angular and spatial information domain and showed improved performance compared to standard image-based coding tools. Finally, view synthesis-based LF compression methodologies have received much interest recently. They rely on the transmission of a sparse set of input views, efficient off-the-shelf coding tools, and the ability to generate a high-quality approximation of the input signal. This framework offers superior RD performance, especially at low bitrates.

Among different prediction methods, it has been reported that the self-similarity based LF image compression methods cannot achieve the comparable performance with the pseudo-sequence-based methods due to their inflexibility to exploit the correlations among various views, especially in low bitrate case [143, 90]. Comparing PVS-based methods to view synthesis-based approaches in a generalized manner is challenging due to different conditions and limited experiments. Typically, view synthesis methods are compared to native PVS-based methodology and report significant gains. Nevertheless, the following works based on PVS show huge improvements as well. View synthesis-based approaches show great potential, especially at low rates, as only a small set of views is transmitted and used to recover the rest of the LF.

PVS-based approaches work well on both lenslet and HDCA contents. Although for HDCA content exploiting correlation becomes more challenging, PVS schemes still perform robustly. This behavior is different compared to, e.g., some transform-based methods (Section 3.2) where performance deteriorates significantly.

3.4 . JPEG Pleno codec

This section briefly introduces the coding technologies proposed by JPEG Pleno. LF codec can take as an input LF data, and potentially the corresponding camera parameters and depth maps. JPEG Pleno proposes two modes for the encoding of LF images: Four-Dimensional Transform Mode (4DTM) and Four-Dimensional Prediction Mode (4DPM). The former is designed as a transformed-based coding tool, inspired by the native JPEG codec and extended to 4D data structure, while the latter uses a prediction mechanism to exploit similarities in LF data.

4D Transform Mode . LF image is firstly partitioned into fixed-size 4D blocks following a predefined, fixed scanning order. 4D blocks can be partitioned further across spatial dimensions, angular dimensions, or not partitioned at all, depending on the cost that partitions generate. 4D-DCT is applied to each block, and the quantized coefficients are encoded using bitplane-wise hexadeca-tree clustering. The hexadeca-tree clustering groups zero coefficients together at each bitplane by partitioning blocks with more than one non-zero coefficient. This partitioning is implemented across all dimensions and generates 16 4D sub-blocks, hence the name hexadeca-tree. The partitioning, 4D-DCT and hexadeca-tree clustering are together driven by Lagrangian optimization. The result is a bitstream that consists of the partition flags, clustering tree, and kept coefficients, which are encoded with binary arithmetic coding.

Fig. 3.4 shows the encoder of LF image coding solution based on block partitioning, 4-dimensional Discrete Cosine Transform (4D-DCT), and hecadeca-tree-oriented bit plane clustering which allows to exploits redundancy in LF data as a

whole.

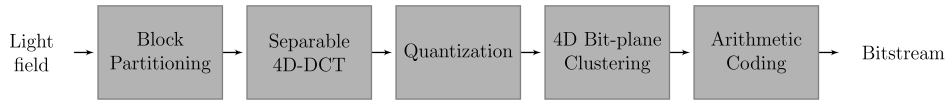


Figure 3.4: JPEG Pleno 4D Transform mode encoder. Inspired by [31].

4D Prediction Mode. 4DPM exploits high correlation between different views in a LF image using geometry-driven warping [10]. By relying on geometry information, 4DPM presents a more universal LF coding tool as it has a capacity for effective compression of narrow and wide baseline LFs.

Fig. 3.5 illustrates the overall block diagram of the prediction mode. The LF is divided into disjoint sets of views corresponding to different hierarchical levels. At the lowest hierarchical level, a set of views (the reference views) and a set of corresponding disparity maps are encoded using an external coding tool such as JPEG2000. At higher hierarchical levels, disparity-based warping and an optimal linear prediction merging of the warped views provide a prediction of current views, denoted as intermediate views. Predicted intermediate views are further refined by applying sparse filtering. Finally, the prediction residual can be encoded using an external codec.

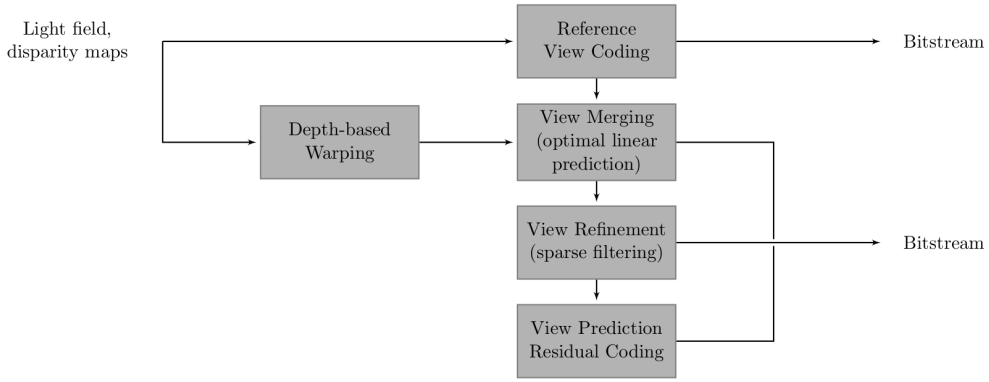


Figure 3.5: JPEG Pleno 4D Prediction mode encoder. Inspired by [122].

4DTM or 4DPM. 4DTM offers good performance on the LFs with high inter-view redundancy, e.g., LFs captured with a plenoptic camera, while not relying on any additional information about a scene, e.g., geometry. This scenario offers great simplicity and flexibility in the coding procedure. Furthermore, the scheme is designed to operate independently on fixed-size blocks, which provides random access capabilities. On the other side, the drawback of the methodology is the lack of expressiveness for sparser LF data.

4DPM uses disparity-based warping and merging, which is different compared to block-based processing applied in 4DTM. Moreover, the disparity information compresses contents captured by multi-camera arrays or gantries efficiently. Of course, the richer set of functional tools increases the complexity of the approach. 4DPM adopts a hierarchical approach that provides a high level of flexibility and adaptability to a particular application. E.g., if random access is required, a single view could be assigned to the lowest hierarchical level, and its decoded version can be used to predict the rest of the LF, which are assigned to the second hierarchical level.

The two modes are typically compared only on lenslet contents as 4DTM is not suitable for LFs with wider-baselines. On widely used test contents, 4DTM achieves from 10% to 30% Bjøntegaard Delta Rate (BD-rate) savings compared to 4DPM [11].

3.5 . Lossless light field coding

Lossless compression of LFs can be broadly divided into two groups. The first group of methods tackles the problem of coding the raw lenslet image [105, 89, 96, 134, 135]. The other group of methods deals with decoded LF images, i.e., LFs obtained from a raw sensor image via a decoding procedure [27, 83].

Helin et al. [47] first encode the center view and its quantized depth map. Then, the center view is partitioned based on the depth values, and for each partition, disparities with respect to side views are computed and transmitted to partition the side views. Residuals, predictor coefficients, and binary sparsity masks are encoded. In a follow-up work [48], the compression performance is improved by segmenting the central view using a complex color- and depth-based segmentation approach, while variable length coding is used to encode the prediction mask and the coefficients of the sparse predictor. Santos et al. [118] conduct a study on the impact of reversible color transformations and data arrangements for pseudo-sequence generation for lossless LF coding. The study shows the superiority of the forward reversible multiple component transform [62]. Schiopu et al. [125] propose a local, context-based method for lossless compression. Based on the edge information, each pixel is adaptively predicted from close co-located pixels in the reference frame. The prediction is subtracted from input values, and three matrices are generated: small residual, high residual, and error sign. The small residual matrix is encoded using context modeling defined by the regions obtained from image segmentation. In [119, 120, 121], minimum-rate predictors are employed that obtain great coding performance but with a high computational cost. Another line of works achieve interesting results with a complexity similar to standard coding tools by adapting the CALIC codec [148] to operate on the EPI representation of LF [92] [93].

Recently, deep learning has been applied to lossless compression of LFs [123,

68, 124]. Schiopu et al. [123] design a deep convolutional neural network that uses a neighborhood of six MIs to predict the current MI. Then, the residual is encoded using a modified version of the CALIC codec. Kaya et al. [68] propose CEPINET, a variant of EPINET [127], which estimates disparity maps of corner views. Textures and disparity maps of corner views and the center view are encoded. Then, a disparity map is generated for each target view by warping the closest reference disparity map to the target view location. The disparity map is divided into connected regions, and for each region, the index of the best reference view (that minimizes MSE over the region) is assigned. Finally, the residual image is also computed and encoded. In [124], view synthesis and prediction methods based on MIs are proposed. The authors also study the influence of the size of the reference image set and modify the CALIC codec's binary mode to utilize different causal neighborhoods.

3.6 . Conclusions and perspectives

This chapter presents an overview of LF lossy and lossless coding solutions based on conventional and deep learning methodologies and describes two coding modes of JPEG Pleno LF codec.

Considering recent advances, LF coding methods based on view synthesis promise outstanding compression performance. Its potential comes from the possibility of radically reducing the number of views that needs to be encoded and transmitted. This pipeline has also been adopted in JPEG Pleno coding solutions, where geometry information is used to reconstruct missing views at the decoder side. Moreover, the high modeling power of deep learning methods allows omitting to transmit even geometry information and still recover missing views in high quality. Besides views synthesis, conventional video coding tools are also popular for LF coding either as the main functional block or as a supporting block, e.g., for residual coding. Moreover, the multiview extension of HEVC is presented as a highly competitive solution for LF compression, with the advantage of being applicable to dynamic contents. Last but not least, transform-based methods show promising results for the compression of contents captured with plenoptic cameras. In particular, the LF coding solution based on 4D-DCT has been adopted in JPEG Pleno standard.

The majority of the chapter is associated with a book chapter:

- M. Stepanov, G. Valenzise and F. Dufaux, Chapter on Compression and transmission of Light Fields: Image/video compression standards & Learning-based coding of light fields, *Immersive Video Technologies*.

4 - Learning-based lossy light field compression

4.1 . Introduction

Most of the approaches proposed for the compression of LFs rely on the available coding tools such as HEVC and its extensions as presented in Chapter 3.2. However, recently it has been shown that, at least for conventional 2D images, the compression pipeline can be replaced by a deep autoencoder, which is optimized in an end-to-end fashion using a rate-distortion loss function. Autoencoders are neural networks that aim to reproduce the input at the output while learning useful representations of the data. Recent quality evaluation of end-to-end learning methods showed a more natural appearance of images compressed in this way, compared to conventional tools such as JPEG2000 [139] and HEVC [21] and present great potential considering their new appearance and the achieved performance. Nevertheless, the end-to-end compression of LFs has not been explored, and we note two main reasons for that. The first one is related to the structure of a convolution operator employed in neural networks. For LFs, 4D operators would be the most natural structure to employ to learn statistical dependencies, but the high dimensionality makes it costly in terms of computation resources. We will consider alternative structures and discuss their strengths and weaknesses. The second reason comes from the size of the data used in the training phase. In the learning-based image compression, the size of the training patches are selected according to available resources, but in the case of LFs, the presence of spatial and angular domains makes the selection of the patch size more challenging.

Even though the potential of these methods has been shown by their competitive performance in terms of objective and subjective metrics compared to traditional coding schemes in previous studies and experiments conducted by JPEG AI [9], auto-encoder-based approaches provide significant gains at low bitrates, if the network capacity stays unchanged, they generally fail to provide high quality and near-lossless reconstructions at high bitrates. This phenomenon is mainly due to the nature of autoencoders, which are intrinsically lossy. Conversely, traditional codecs are designed to span the full quality range, and in particular to provide near-lossless performance at higher bitrates.

To incorporate the benefits of both approaches, this chapter proposes overcoming the observed lack of scalability of deep learning approaches by adding an enhancement layer. We propose a hybrid scheme consisting of a base layer that provides high gains at low/mid bitrates and serves as an efficient predictor for high bitrates. This is complemented by an enhancement layer which allows the coding of the residual signal via a traditional coding scheme and provides improved

performance at high bitrates. Furthermore, we explore various traditional coding schemes for the residual signal and show that even with a simple approach such as scalar quantization it is possible to achieve significant gains with respect to the base layer and to be competitive with state-of-the-art LF codecs.

In this chapter, we aim at exploring the following questions:

- How can learned image-based codecs be extend to LF data?
- Are there any particular convolutional layers that result in superior coding performance?
- How the learned, baseline LF codec compares to state-of-the-art coding schemes?
- How can we extend the baseline model to improve coding gains?

The coming sections, first, compare different ways of organizing and processing input LFs, including patch-based and scalable, resolution-invariant approaches, and network architectures based on 2D and 3D convolutional layers. The scalable approach based on 2D filters is deemed the most efficient in terms of RD performance and it is, finally, integrated into a hybrid pipeline and the overall system is compared to state-of-the-art methods.

4.2 . Related work

Related work includes recent end-to-end learning-based image compression methods and LF schemes. Among LF coding schemes, the most relevant methodologies are based on in transform-based coding methods group presented in Section 3.2. In addition, deep learning-supported schemes can be included. Besides being designed to exploit features of LFs, both of these groups, also, share in common that they consist of different functional blocks which were optimized independently.

Conversely, an end-to-end scheme learns a single function that jointly optimizes and integrates all needed operations. As a result, the overall scheme leads to potentially more efficient solutions.

End-to-end learning-based compression has been recently proposed in [14] [137] [136] [113] and has gained huge popularity due to its ability to replace the whole traditional compression pipeline with a single function. Ballé et al. [14] propose an end-to-end compression approach that consists of analysis and synthesis functions corresponding to an encoder and a decoder in conventional pipelines, respectively, plus a uniform quantizer. In addition, they propose a differentiable quantization mechanism allowing to optimize the RD function directly. Theis et al. [136] propose a similar approach but deal with quantization and bitrate estimation in a different manner. Rippel et al. [113] propose a real-time codec that applies a pyramidal analysis for the feature extraction and an adaptive coding module and regularization. Conversely, to the previous approaches, Toderici et al. [137] overcomes the

necessity to train a separate model for each lambda value in the RD function by adopting the encoding in a progressive manner.

Our approach extends the work of Ballé et al. [14], and operates on data of a higher dimension, requiring a careful design of the network architecture to handle particular filtering across different views. Furthermore, motivated by the limitations of auto-encoders in providing high-quality reconstructions, we also introduce an enhancement layer to encode the residual signal.

4.3 . Preliminary studies

The high dimensionality of LF data allows partitioning different dimensions and adapting a new form to available processing tools. Considering image-based coding tools, besides individual processing of each SAI, MI, or EPI, interleaving angular and spatial dimensions on a 2D grid presents a good prospect as it suggests higher gains due to the possibility of jointly exploiting redundancies along different dimensions. Similarly, if video coding tools are considered, arranging LF views in a PVS allows reducing redundancy along spatial and inter-view dimensions by leveraging rich improved prediction mechanisms.

Analogously, we experiment with a similar configuration in a learning scenario inspired by deep learning-based image compression approaches. First, a block-based scenario is considered wherein each block is independently processed. Second, a scalable design is adopted allowing size-agnostic processing. The high dimensionality of LF images allows considering various ways to reshape it prior to the encoding. Finally, two architectures are considered based on the dimensionality of the convolutional kernel.

4.3.1 . Block-based approach

In an initial study, 4D blocks are independently encoded similarly to traditional JPEG coding scenario. Following a recent study on the redundancy in LF blocks [103], LF image is divided into operational 4D blocks of size $5 \times 5 \times 13 \times 13$. The selected size is sufficiently wide in spatial dimensions so that a 3D point remains in the co-located spatial blocks across different views 80% of the time while maintaining affordable computational demand. Zhong et al. [150] conducted a similar study and estimate the scattering of a 3D point across different views. Their findings concluded that in LF images captured by plenoptic camera focused points scatter in the neighborhood of about 2-3 macro-pixels while defocused points scatter in the neighborhood of 7-8 macro-pixels. Finally, they selected a neighborhood of 2-3 macro-pixels due to high resource demand in storing references from a greater causal window.

In our case, each block is organized in MI representation allowing a less complex processing pipeline compared to potentially operating on a native 4D block. The reshaped block is propagated through a series of 2D convolutional layers as illustrated in Figure 4.1. We adopt autoencoder architecture which gradually re-

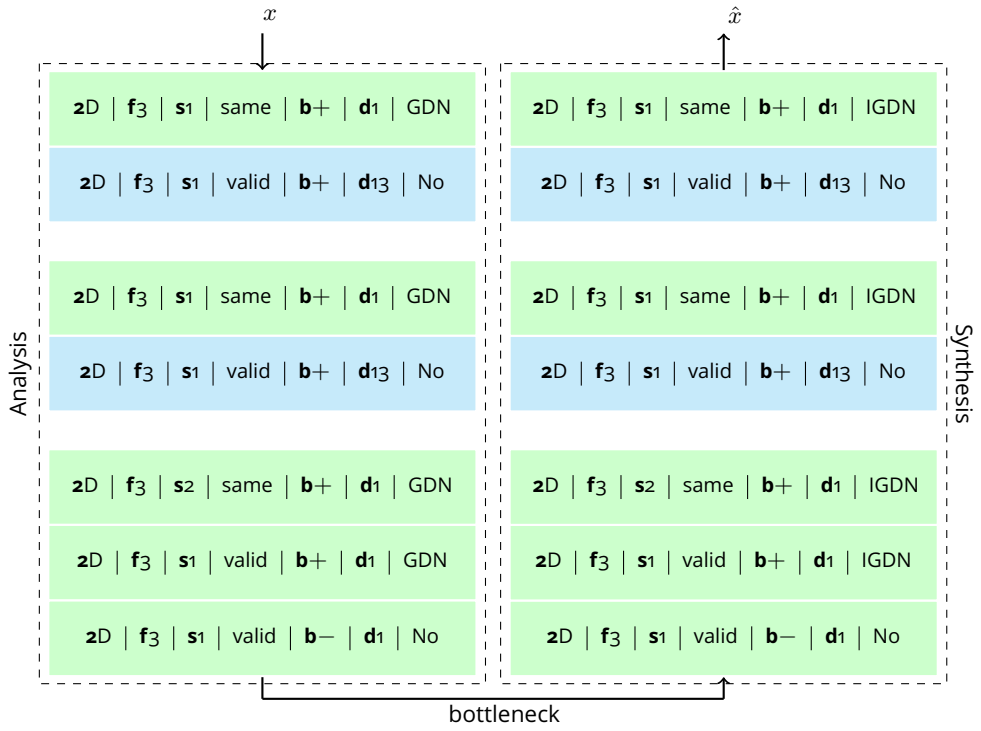
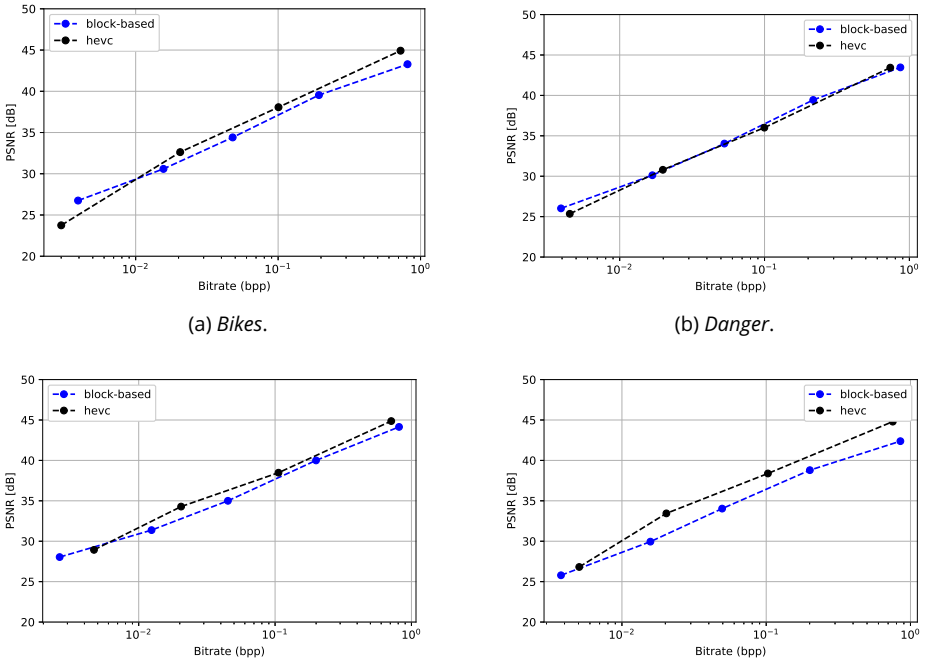


Figure 4.1: The parameters of the autoencoder used in the patch-based codec. Each box defines a layer with a structure [*convolution dimension*]D | f[*kernel size*] | s[*stride step*] | [*padding*: **same** or **valid**] | b[*using bias weight* (**+**) *or not* (**-**)] | d[*dilation size*] | [*activation*: **(I)GDN** or **No**].

duces the spatial resolution of the input signal along with the encoding module, or analysis function, and generates a compact representation, analogous to transform coefficients. The encoding modules consist of two blocks of interleaved filters and three 2D convolutional layers. Interleaved layers were proposed as a computationally and storage-friendly alternative to 4D convolutional layers and consist of alternating the processing along spatial and angular dimensions. This is achieved by using a dilated 2D convolution that operates on co-located pixels across different macro-pixels, and traditional 2D convolution which operates across the local, dense neighborhood. Each interleaved block reduces spatial resolution (the number of macro-pixels of the block) by convolving only on the valid region inside the block i.e. no padding applied, resulting in a 1×1 spatial support after the two blocks. The following three layers contain solely dense layers and compress in the angular domain. The decoding part of the autoencoder, the synthesis function, reconstructs the input block by following the design of the analysis function in the reverse order, and by replacing the forward convolutions with transposed convolutions. The autoencoder is trained following the framework of Ballé et al. [14]. Besides the autoencoder network, an entropy model is used to obtain the probability distribution of each quantized feature map at the bottleneck. The weights of all blocks (analysis, synthesis, and the entropy model) are learned by minimiz-



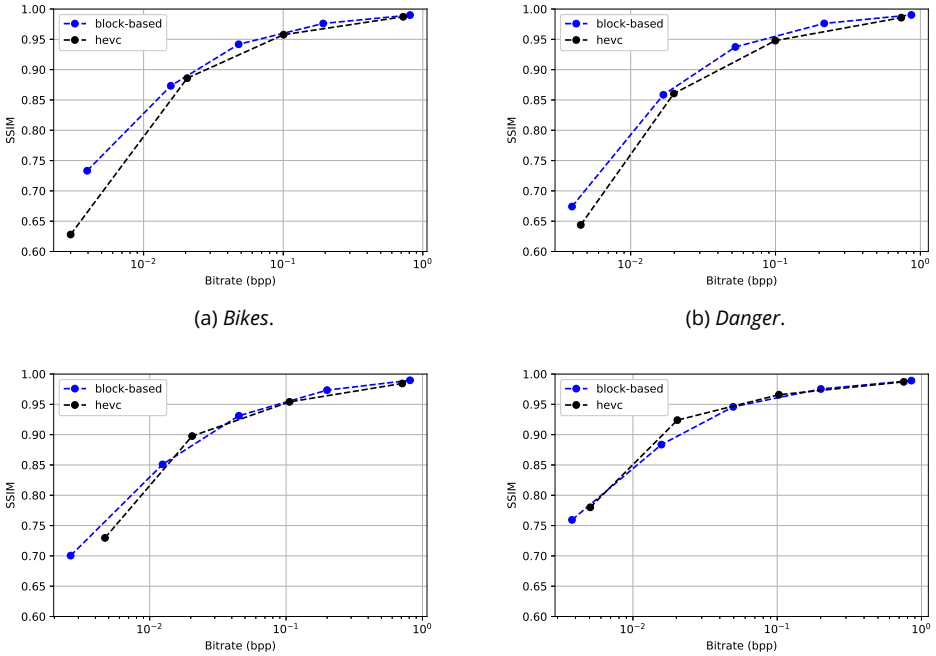
(a) *Bikes*. (b) *Danger*.

(c) *Pillars*. (d) *Fountain*

Figure 4.2: RD comparison of learned block-based codec and HEVC in terms of PSNR.

ing the rate-distortion function $R + \lambda D$. The rate R is the entropy of learned probability density function of the representation at the bottleneck obtained as $\frac{1}{N}\sum_n^N log_2 p(x)$ where N is the number of coefficients at the bottleneck, x is a value of a coefficient, and $p(x)$ is its probability obtained by the entropy model. The distortion D is the mean square error between the input and the decoded LF, and λ governs the trade-off between the two. As the quantization adds discontinuities to the loss function, it prevents learning. The non-differentiable quantization is overcome by introducing two operational modes of the quantizer. During the test phase, the quantization block works as a uniform quantizer rounding estimated coefficients to the nearest value. In contrast, in the training phase, the rounding is simulated by adding uniform noise $U(0, 1)$. After encoding all patches, we use the gzip codec to pack all of them together (and potentially reduce the redundancies between patches).

We use EPFL LF image dataset [110] which consists of 118 images. The images were decoded using LFToolbox version 0.4 [27, 28] and we utilize only the subset of 13×13 sub-aperture view. The images were converted to YCbCr color space and we keep only the Y component for our experiment. Four widely accepted LF images, illustrated in Figure 2.7, are kept for testing purposes, while the remaining part of the dataset is divided into training and validation sets in an $80/20$ ratio. More than 800k luma patches of size $5 \times 5 \times 13 \times 13$ are extracted for training by selecting uniformly across spatial dimensions. Five neural networks were trained



(a) *Bikes*. (b) *Danger*.

(c) *Pillars*. (d) *Fountain*.

Figure 4.3: RD comparison of learned block-based codec and HEVC in terms of SSIM.

with $\lambda \in \{10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$ using Adam optimizer with the learning rate of 10^{-4} assigned to the main loss, i.e. update the weight of the autoencoder, and the learning rate of 10^{-3} assigned to the entropy loss, i.e. update the weight of the entropy model. We set the batch size to 32. The learned codec is compared to x265 implementation of HEVC codec. Prior to the encoding with x265, an input LF image is rearranged in PVS as it is proposed in CTC of JPEG Pleno [60]. As the proposed scheme operates only on the luma component the input to the anchor is modified by setting chroma components to neutral color in order to provide a fairer comparison. RD curves of the two methods are presented in Figure 4.2 and Figure 4.3. In Figure 4.2 PSNR is used as the quality metric, while SSIM is used in Figure 4.3. To compute the bitrate, we construct bitstream by encoding all range coded patches and adding to each of the two bytes to denote the size of the coded patch, and dividing the total number of bits with the size of LF image i.e. $13 \times 13 \times 434 \times 625$.

Results in Figure 4.2 show that the proposed codec offers competitive performance compared to the state-of-the-art anchor. The *Fountain* content, for example, proved more challenging for the proposed scheme, likely due to the presence of large smooth surfaces which HEVC can encode more efficiently. RD curves in Figure 4.3 illustrate that, in terms of SSIM, the proposed scheme shows the potential especially at the lower rates.

These results show the capacity of autoencoder-based codec that leverages

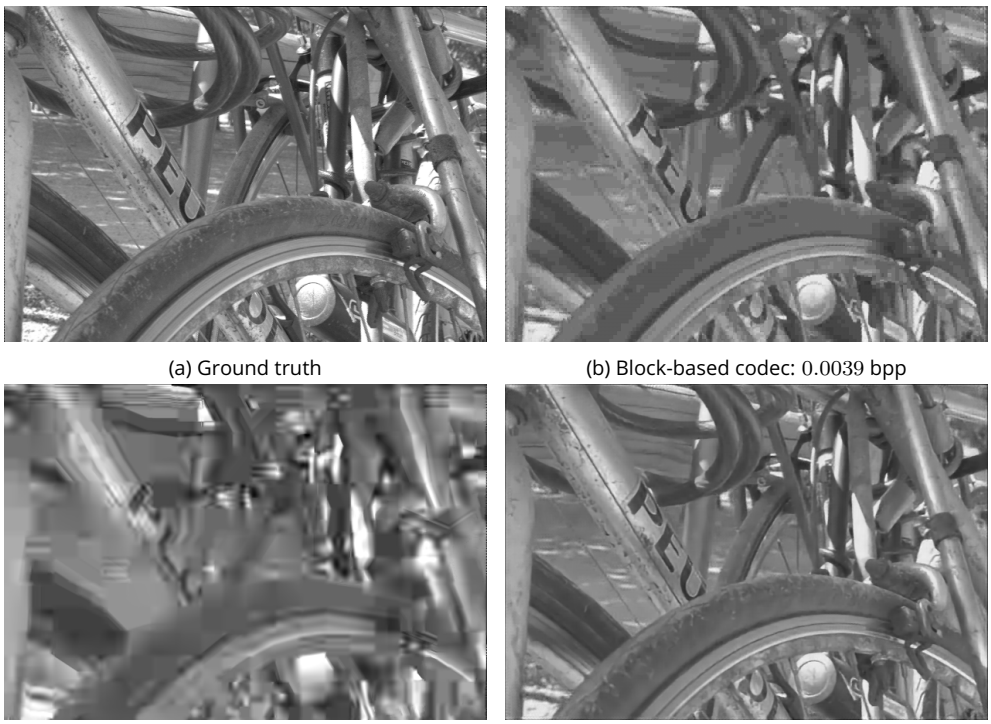
high modeling capabilities deep learning to achieve superior performance. On the other side, the patch-based mechanism limits the potential of further redundancy exploitation in spatial dimensions as it operates on a fixed spatial extent. Moreover, the dilated convolution layer in the interleaved block relies on prior knowledge of the size of the MIs. These two features of the learned codec limit the flexibility of the scheme and the applicability to LFs with different parameters without additional reparametrization and retraining.

4.3.2 . Towards a holistic approach

The block-based approach accepts a particular input size, i.e., a 4D block of size $5 \times 5 \times 13 \times 13$, and reconstructs it. Therefore, it cannot be applied to other inputs and recover the appropriate input size. The limitation of this design lies in processing small, fixed patches, which prevents exploiting (primarily) spatial and angular correlations in larger regions in LF image. The consequences are clear blocking artifacts at lower rates, as seen in Figure 4.4. To overcome these limitations, we explore various architectures that would offer a codec that is agnostic to spatial and angular sizes of input LFs. In particular, we considered three kernel structures for the network's building blocks and their relationship to the input data structure. A 4D kernel presents the most natural way to explore the LF structure, as it traverses all dimensions and would present the optimal kernel to explore correlations in 4D data. On the contrary, it is highly complex regarding computation and memory. Kernels of lower dimensions, on the other side, are less computationally demanding and sub-optimal as the required reshaping to a lower-dimensional structure decreases the correlation in data. Another by-product is operability on inputs of particular spatial and angular dimensions. In the following experiments, due to the challenging complexity of the 4D kernel, only architectures based on 2D and 3D convolutional layers are considered. Note that scalability typically addresses particular functionalities of a codec, e.g., quality and resolution scalability, which is not the case here.

2D convolution

In this case, we consider a different architecture compared to the one used in Section 4.3.1. Instead of using a MI representation, SAIs are scanned following horizontal raster scan order, i.e. row by row selection, and stacked along a third dimension to obtain PVS representation. The motivation behind this arrangement is two-fold. First, 2D filters are used for the processing which implies low complexity and computational footprint. Second, by treating the stacked view as channels of a 2D input, each filter can learn correlation among all views in the LF. The drawback is that the scheme depends on the angular resolution and the scanning order that generates PVS. Therefore, if the angular resolution or the scanning order change the network needs to be re-trained to adjust to the parameters. The top illustration on the Figure 4.5 tries to make the intuition clear. A patch of size $H \times W \times C \cdot V$



(a) Ground truth

(b) Block-based codec: 0.0039 bpp

(c) HEVC (x265): 0.003 bpp

(d) Scalable codec: 0.0032 bpp

Figure 4.4: Visual evaluation of the central view of *Bikes* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d).

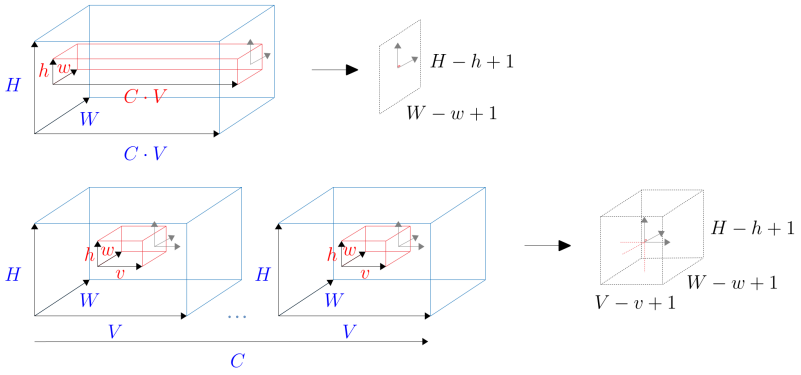
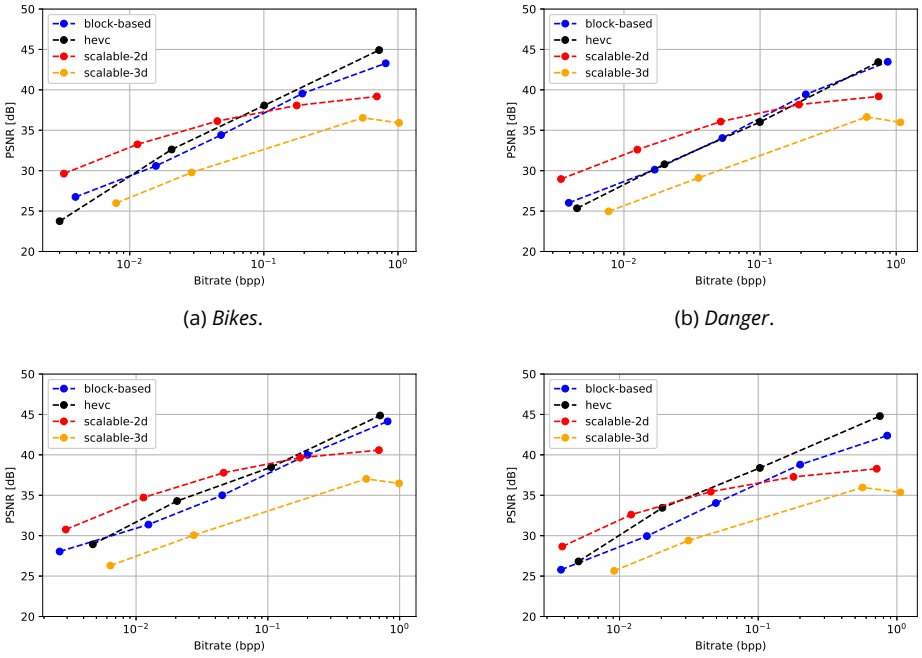


Figure 4.5: Convolution mechanism.

(shown in blue) is created by stacking perspective views (V views each having C channels), a 2D filter of size $h \times w \times C \cdot V$ (depicted in red) traverses along two spatial dimensions as depicted by gray arrows. The convolution of the two generates the resulting 2D feature map. As illustrated, co-located neighborhoods from all views and channels contribute to the computation of a particular pixel in the feature map. In the later text, a codec employing described dynamics will be referred to as a scalable-2D codec.

3D convolution

A drawback of the setup based on the 2D convolution is that it does not scale with the angular dimension, in the sense that if the number of views changes a new model must be trained. The reason for this limitation comes from the stacking of the views which inherently produce a fixed number of channels at the first layer of the network. On contrary, a setup facilitating 3D convolution could circumvent the lack of angular scalability as 3D kernels traverse along the third dimension. This case does not come as a free lunch either as a trained model is likely to be less efficient in the scenarios where a different scanning of SAI is used compared to the one(s) employed during training, i.e. it has to be re-trained. Still, this lacking applies to the previous scheme as well. Another point that makes this scheme less effective compared to the 2D alternative is limited receptive field of the 3D kernel. The bottom part of Figure 4.5 illustrates the mechanism of 3D convolution where the input patch, which is a 4D structure spanning spatial dimensions with sizes H and W, V views, and C channels, is represented by a blue hyper-cuboids, the convolutional filter is represented by the red hyper-cuboid and the resulting features maps are pointed by the black arrow. A 3D kernel, differently compared to the 2D kernel, operates along three dimensions as denoted with the gray arrows. Similar to the naming of scalable-2D codec, the approach based on 3D convolution will be referred to as scalable-3D codec.



(a) *Bikes*. (b) *Danger*.

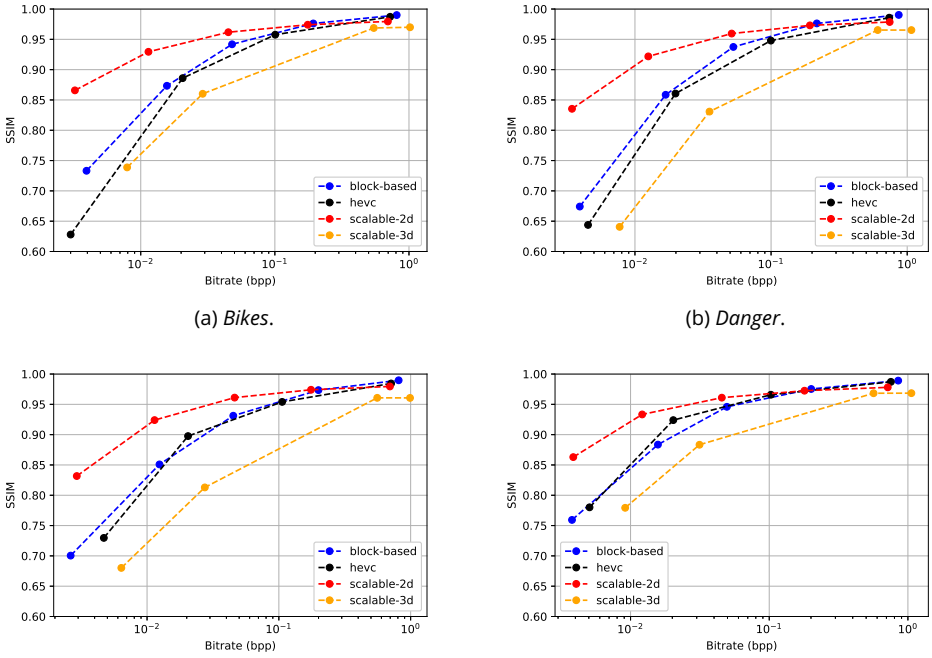
(c) *Pillars*. (d) *Fountain*.

Figure 4.6: RD comparison in terms of PSNR.

Quantitative analysis

Similar to the analysis in Section 4.3.1 we compare the new scalable architectures with the block-based codec and HEVC anchor. Figure 4.6 shows RD comparison in terms of PSNR. Considering scalable-2D codec, it is clear that it provides improved performance compared to other methods at low rates. On contrary, as the available bandwidth increase, the performance of the method tends to saturate, and both, the block-based approach and HEVC anchor, report gains. The maximum quality and the cross-point from superior performance to the saturating depend on the content. The saturation is likely related to the capacity of the network, i.e. the number of parameters, as similar observation can be noted in learned image compression codec [14]. The quantitative analysis in terms of SSIM magnifies the potential of the scalable codec compared to the block-based codec. As depicted in Figure 4.7, 2D method significantly outperform other anchors especially at low rates. Observing the quality further along increasing bitrates, the gains gradually diminish with all methods, except 3D scalable codec, showing similar performance for bitrates in the range $[0.1, 1]$.

In the case of the scalable-3D codec, the situation is greatly different as its performance is significantly lower compared to other methods in terms of both PSNR and SSIM. There are a couple of reasons that could drive this outcome. First, the capacity of the network or the number of parameters. Compared to



(a) *Bikes*.

(b) *Danger*.

(c) *Pillars*.

(d) *Fountain*.

Figure 4.7: RD comparison in terms of SSIM.

the scalable 2D codec, the scalable 3D codec has a lower number of parameters which could limit its capabilities. Still, an experiment with an increased number of filters, which results in a drastic increase in the number of parameters, showed similar performance suggesting that the capacity is not the main contributor to the observed performance. Second, the architecture itself might be too simple to exploit the intrinsic structures in PVS provided at the input. Namely, besides spatial relations, the 3D kernel attempts to learn the relationships along the third dimension, i.e. among neighboring views. These relationships depend on the scanning order during the construction of the input PVS and in the case of the spiral order, there might be a dozen of them significant, which might be challenging to learn. On the contrary, the scalable 2D codec takes into account all views simultaneously, allowing to exploit similarities between views more efficiently.

Considering its superiority and potential, the scalable 2D codec is used as a baseline for a codec operational across all bitrates. In the coming chapters three enhancement layers, built on top of the base codec are explored and the entire system is compared to the state-of-the-art method proposed by JPEG Pleno.

4.4 . Hybrid codec

As observed in Section 4.3.2 the scalable 2D approach has great potential and significantly outperforms other schemes with significant gains at low rates.

Nevertheless, the gains diminish at higher rates rendering the entire method with operable challenges for these scenarios. In this section, the scalable 2D codec is considered as a base coding layer of a hybrid codec wherein an enhancement layer is included to remedy the lack of efficiency of the base layer at high rates. Three hierarchical layers are considered and evaluated subsequently, and the best approach is further compared to state-of-the-art methods. Figure 4.8 shows our proposed scheme. The base layer is denoted with red blocks while the enhancement layer is colored in cyan. The layers are presented in more detail in the following subsections.

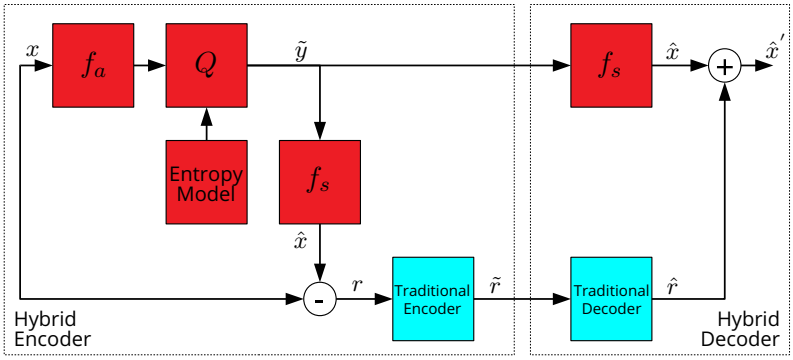


Figure 4.8: Proposed hybrid coding scheme.

4.4.1 . Base layer

An end-to-end trained compression scheme based on the recent works of Ballé et al. [13, 14] is used as a base layer. Similarly, as explained in Section 4.3.2, the scheme takes a LF image as input, reshapes it by extracting sub-aperture views, and stacking the views along the third dimension following horizontal raster scan order. We employ slightly modified architecture compared to the one presented in [13] and a factorized prior used in [14]. In particular, the compression architecture is comprised of four functional blocks: an analysis function $f_a(\theta)$ which creates a more compact representation of the input $y = f_a(x)$, a quantization block $Q(\eta)$, which provides quantized version of y, $\tilde{y}$, an entropy model that learns marginal probability distribution (i.e. per channel) of the quantized coefficients, and a synthesis function $f_s(\phi)$ whose goal is to reconstruct the input from the quantized compact representation $\hat{x} = f_s(\tilde{y})$. See diagram in Figure 4.8.

The analysis function comprises a set of sequential non-linear, downsampling, and convolutional layers while the synthesis function is the symmetric counterpart of the encoding function with downsampling layers replaced by upsampling layers. Differently compared to the original architectures [13, 14], we introduce layers with 1×1 kernels to limit the processing inside MIs. Furthermore, we use a simpler architecture without the hyper-prior [14]. The output of the analysis function is a vector of the same dimensionality as the input but with the reduced spatial size and increased number of channels. The quantizer works in two modes: the training

or testing phase. During the training phase, the quantizer adds a uniform noise to the coefficients obtained from the analysis function to approximate quantization in a differentiable manner. At the test phase the coefficients are approximated with the nearest integer value. The entropy model is a CNN that learns the probability density function of each feature map of the coefficients. It is based on the cumulative function [14]; it takes an input coefficient and provides its cumulative value. Thus, the cumulative values around the coefficient value are simply subtracted to obtain the probability of each coefficient. A detailed description of the network parameters is summarized in Figure 4.9.

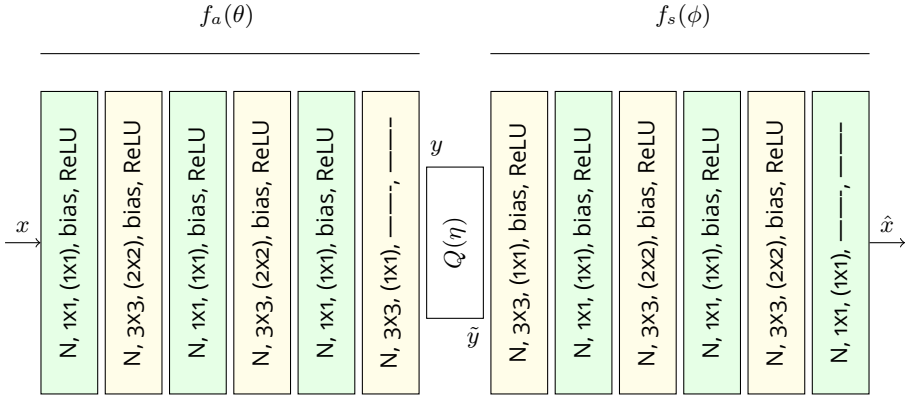


Figure 4.9: Neural network architecture. The parameters in each block denote the number of filters, the spatial extent of the filter, stride, the usage of the bias, and the activation function.

Weights of the analysis function, the entropy bottleneck and the synthesis function, θ, η and ϕ respectively, are learned by minimizing the RD function $J(\theta, \eta, \phi; x) = R(\tilde{y}) + \lambda D(x, \hat{x})$ where the rate $R(\tilde{y})$ is modeled with the entropy of the compressed bottleneck, the distortion D is the mean square error between the input x and the decoded LF $\hat{x}$, and the parameter λ governs the trade-off between the rate and the distortion. Five models are trained by selecting five different lambda values. For each model, the weights are learned using Adam optimizer with a learning rate of 1×10^{-4} and 1×10^{-3}, for θ and ϕ, and η respectively. The final bitstream is obtained by packing encoded coefficients, the LF size, and the lambda parameter.

The base layer operates at the number of RD points equivalent to the number of models trained. The discrete number of available models and a different level of complexity of input content make providing a particular performance challenging. Note, there are already some methods that allow operating in a more flexible manner, e.g., [36].

4.4.2 . Enhancement layer

As observed before, the autoencoder-based approach used in the base layer reaches saturation in performance at higher bitrates. A possible solution to this problem is increasing the network's capacity, as Ballé et al. investigate in the

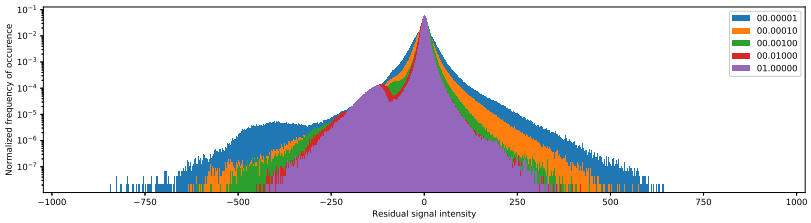


Figure 4.10: The histograms of residual signals obtained by compression with different models for content *Bikes*.

case of learned image compression (see Figure 10 in [14]). However, this requires increasing model complexity when approaching higher quality points, and we do not explore it now. Instead, the introduction of an enhancement layer is proposed to encode the residual signal between the original LF and the reconstruction from the base layer. The advantage of this hybrid approach is that the residual coding allows incorporating any available coding system. We compute the residual signal by subtracting an input LF image and its prediction obtained using the base layer

$$r = x - \hat{x},$$

followed by thresholding of the residual signal in the range $[-2^{n-2}, 2^{n-2}-1]$, where n corresponds to the precision of intensities in the original signal. The thresholding has been selected heuristically by observing the residual signals. E.g. Figure 4.10 illustrates histograms of residual signals obtained by subtracting decoded versions of *Bikes* content from the original LF. Each color denotes a lambda value used to train a model and subsequently obtain the decoded version of the original content.

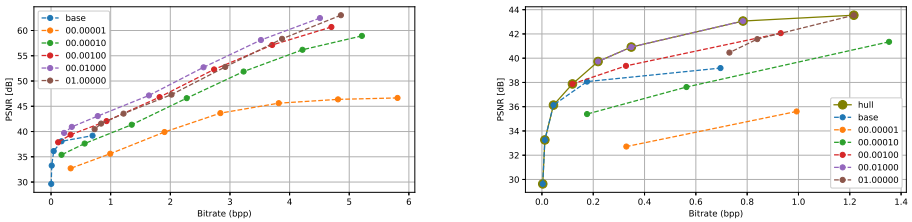
After thresholding, the intensities are translated to the range of values $[0, 2^{n-2}-1]$. Note the precision of the original LFs is 10 bits. Obtained representation is then encoded using one of two types of enhancement layers: HEVC Intra and HEVC Inter. In addition, the residual signal is processed with a scalar quantizer of a form

$$\tilde{r} = \lfloor \frac{r}{q} \rfloor \cdot q,$$

with $q = 2^i, i \in \{1, ..., 7\}$ and the introduced distortion and the entropy of the quantized signal are used as a baseline anchor of the hybrid coder. For each base layer codec that operates at a particular bitrate a local RD curve around each RD point of the base layer is obtained by quantizing the corresponding residual signal. The resulting set of RD curves for *Bikes* content is depicted in Figure 4.11a.

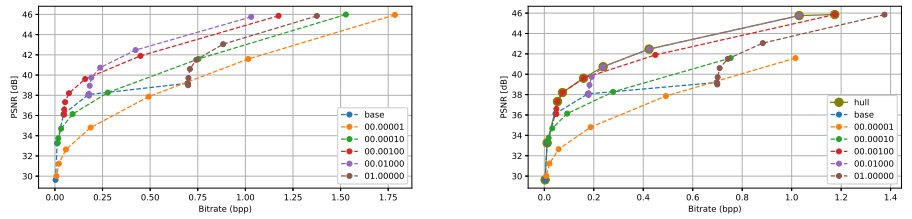
In the case of HEVC Intra and Inter coding, each view of the residual LF signal is padded to 632×440 size. Padded views are then scanned using a spiral order pattern, arranged in a pseudo-video sequence, and, finally, provided to the HEVC codec. In the case of HEVC Intra coding, the Intra Main10 profile is employed with quantization parameters selected as $Q_p \in \{17, 22, ..., 42, 47\}$. In the latter

case, the Low Delay-P, Main10 profile, and quantization parameters selected as $Q_p \in \{12, 17, ..., 32, 37\}$ are used. For experiments reference software HM 16.0 is utilized. For the *Bikes* content the sets of RD curves obtained using the hybrid codec with the two HEVC variants are presented in Figure 4.12a and 4.13a.



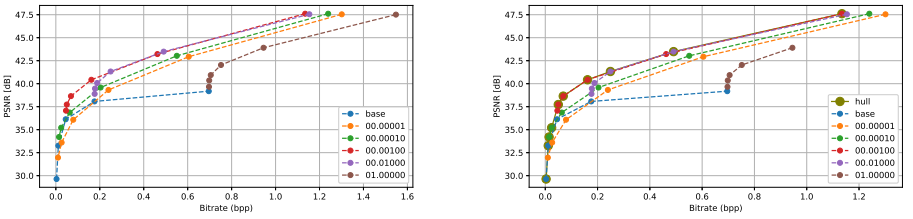
(a) Initial RD curves. (b) Convex hull.

Figure 4.11: (a) RD curves obtained for *Bikes* content using the base layer and scalar quantization as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points.



(a) Initial RD curves. (b) Convex hull.

Figure 4.12: (a) RD curves obtained for *Bikes* content using the base layer and HEVC codec (Intra) as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points.

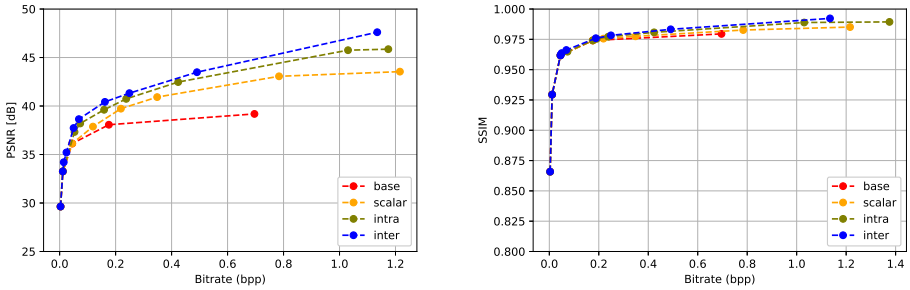


(a) Initial RD curves. (b) Convex hull.

Figure 4.13: (a) RD curves obtained for *Bikes* content using the base layer and HEVC codec (Inter) as the enhancement layer. (b) The final RD curve obtained by computing the convex hull of all RD points.

4.4.3 . Quantitative analysis - Enhancement layers

In order to compare proposed variants, a single RD curve corresponding to each variant needs to be computed. These final curves are obtained by computing the



(a) Comparison in terms of PSNR.

(b) Comparison in terms of SSIM.

Figure 4.14: Comparison of performance of the base layer and three variants of the hybrid codec in terms of PSNR and SSIM for the *Bikes* content.

convex hull of RD points. The procedure starts with the collecting of all RD points and the sorting of the points in ascending order based on bitrate value. Then, the sorted array is evaluated across a quality metric dimension and RD points not satisfying increasing quality are removed. Finally, the remaining information is processed by MATLAB's *convexhull* function that selects the points that lie on the boundary of the convex hull. The resulting RD curves of the hybrid codec facilitating scalar quantization, HEVC Intra codec or HEVC Inter codec are shown in Figures 4.11b, 4.12b or 4.13b, respectively. In each diagram, the RD points of the convex hull are depicted as underlying wider, olive-color circles. It can be noted that the convex hulls are computed in a narrower range compared to the initial operability range of bitrates, e.g. see Figure 4.11, as common test conditions suggest bitrates up to 0.75 Bits Per Pixel (bpp).

The performance of the three variants is evaluated in terms of PSNR and SSIM and compared in terms of Bjøntegaard Delta Rate (BD-rate) measure to the base layer. Figure 4.14 compares the performance of the base layer to the proposed extensions for the LF content *Bikes*. It can be noticed that at low bitrates enhancement layer does not improve performance, in terms of PSNR, suggesting the superiority of the base layer. On the contrary, at high bitrates, we can notice the benefit of adding the enhancement layer and note that even a simple method such as scalar quantization provides $\sim 7-13\%$ bitrate savings compared to the base layer. Further improvements are obtained with HEVC's Intra and Inter prediction modes which gain by exploiting the spatial and inter-view correlations in the residual signal. The comparison in terms of SSIM follows similar trends but the gains are observable smaller compared to the PSNR evaluation. The quantification of the performance is presented in Table 4.1 for all test contents. The diagrams showing RD curves in terms of PSNR and SSIM for the rest of the LF contents are presented in Appendix A.5.

The proposed hybrid approach computes the convex hull to determine the optimal performance and the combinations of lambda values and quantization parameters to operate at a particular RD point. Although this approach limits the

Table 4.1: BD-rate savings of the proposed hybrid approach facilitating one of three types of enhancement layers against the base layer.

	Base		
Sequence	Hybrid-Scalar	Hybrid-Intra	Hybrid-Inter
Bikes	-11.864%	-27.426%	-31.964%
Danger	-7.867%	-13.866%	-16.085%
Pillars	-9.637%	-22.131%	-22.679%
Fountain	-12.408%	-22.249%	-31.266%

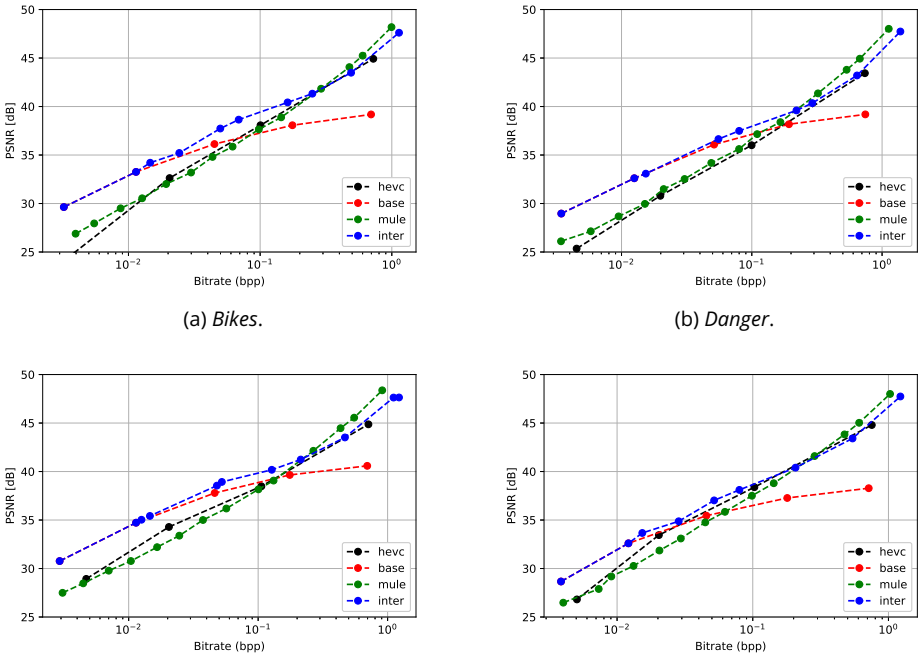
practical utility of the proposed method, it still offers an insight into the interaction of the base layer and the enhancement layer. Figures 4.11, 4.12 and 4.13 show that the base layer is crucial for the efficient performance at lower bitrates while the impact of the enhancement layer is extremely low. In contrast, the situation becomes the opposite at mid and high rates. Moreover, higher lambdas might not even contribute to constructing the optimal convex hull.

4.4.4 . Quantitative analysis - State-of-the-art

The best-performing hybrid variant is compared to the base layer and two anchor methods proposed by JPEG Pleno. RD comparison in terms of PSNR and SSIM is presented in Figure 4.15 and Figure 4.16, respectively, while BD rate evaluation in terms of PSNR is shown in Table 4.2. The hybrid approach gains at lower bitrates thanks to the learning-based base layer, while the introduction of the enhancement layer increases the performance and makes the approach competitive with the anchors at high bitrates. Nevertheless, the overall performance suggests significant gains of the approach against the two anchors ranging from $\sim 16\%$ to $\sim 50\%$ saving with respect to HEVC and from $\sim 25\%$ to $\sim 40\%$ saving with respect to MuLE [29].

Table 4.2: BD-rate savings of the proposed hybrid approach (base layer with HEVC-Inter at the enhancement layer) with respect to the base layer (no enhancement layer), MuLE, and HEVC reference software x265.

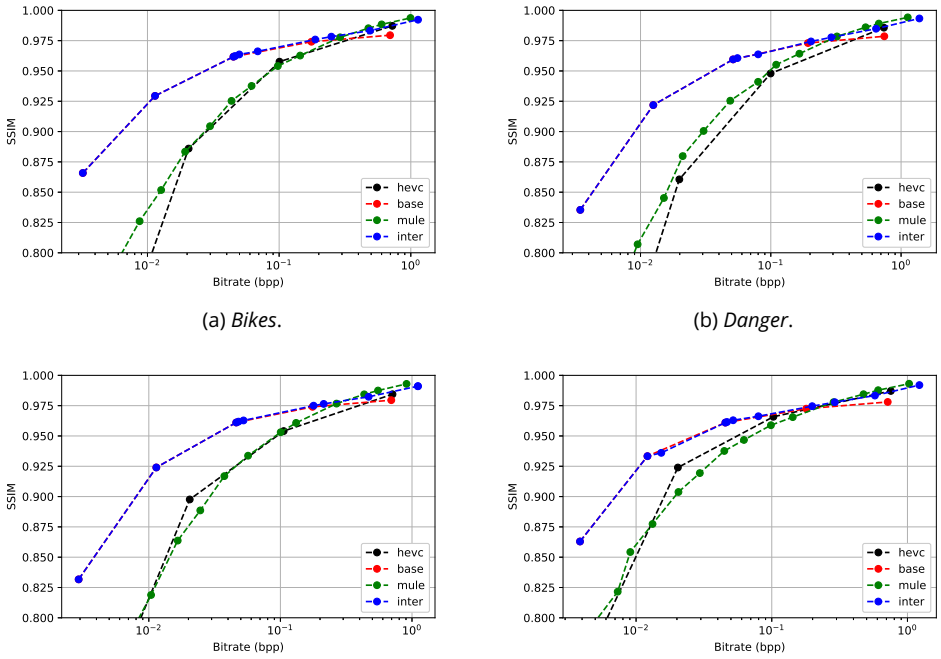
	Base	MuLE	HEVC
Sequence		Hybrid-Inter	
Bikes	-31.964%	-36.035%	-40.321%
Danger	-16.085%	-25.135%	-49.480%
Pillars	-22.679%	-39.454%	-45.739%
Fountain	-31.266%	-25.012%	-16.393%
Average	-25.497%	-31.409%	-37.982%



(a) *Bikes*. (b) *Danger*.

(c) *Pillars*. (d) *Fountain*.

Figure 4.15: Comparison of the proposed approach to state-of-the-art methods in terms of PSNR.



(a) *Bikes*. (b) *Danger*.

(c) *Pillars*. (d) *Fountain*.

Figure 4.16: Comparison of the proposed approach to state-of-the-art methods in terms of SSIM.

4.5 . Conclusion

In this chapter, a hybrid coding scheme for LF image compression is presented. The scheme facilitates two operational layers: a base layer and an enhancement layer. The base layer leverages the high modeling power of the deep learning methodology. More precisely, an auto-encoder-based architecture is employed together with entropy constrained bottleneck to achieving particular operability (in RD sense) of the resulting codec. The enhancement layer, on the other side, uses traditional coding to achieve fine-grained quality scalability. The two layers complement each other as the former provides superior performance at lower rates while the latter allows efficient operability at higher rates. The proposed approach achieves better performances against state-of-the-art anchors. Namely, the results show that the hybrid scheme greatly improves the performance at high bitrates and moderately at mid bitrates compared to the independent working of the base layer. Compared to other state-of-the-art methodologies such as the transform mode of JPEG Pleno codec, MuLE, and HEVC anchor, the proposed scheme offers superior performance at low bitrates induced by the learned representation of the base layer.

Some results are associated with the following publication:

- M. Stepanov, G. Valenzise and F. Dufaux, "Hybrid Learning-Based And Hevc-Based Coding Of Light Fields," *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3344-3348, doi:10.1109/ICIP40778.2020.9190971.

We conclude the chapter by explicitly answering initially posed questions in the Introduction:

1. How can learned image-based codecs be extend to LF data? *Considering the functional blocks of a learned codecs, the encoder consists of transform, quantization and entropy model blocks, while the decoder is symmetric. The whole framework is optimized so that the representation obtained after encoding requires minimum cost given the quality of reconstructed input. Naturally, the transform should consider all dimensions in the input to find the efficient representation. Yet, a high dimensional kernel might be too costly so we consider re-arranging the input signal to allow exploiting correlation with a lower dimensional kernel.*

2. Are there any particular convolutional layers that result in superior coding performance? *In this chapter, we have experimented with 2D and 3D kernels and observed that the 2D kernels show superior performance to the 3D kernel. Although this result was initially surprising as the 3D kernel should provide a greater extent in exploiting correlation across more dimensions, we note that the design of the coding solution based on 2D convolutional layers offered greater receptive field across all dimensions.*

3. How the learned, baseline LF codec compares to state-of-the-art coding schemes? *The baseline LF codec facilitating 2D convolutional layers is highly competitive to state-of-the-art methods especially at lower bitrates. On the other side, it lacks flexibility to operate at higher rates. Practically, this could be circumvented by increasing the capacity of the model or by introducing an enhancement layer which encodes the residual signal.*

4. How can we extend the baseline model to improve coding gains? *We consider a hybrid design to overcome the limitations of the baseline model. Initially, we observed that the most trivial approach to the coding of the residual signal adds more flexibility in the overall scheme and improves coding gains. Finally, we adopt a standard coding tool, the HEVC codec, to encode the residual signal, but note that any codec can be used.*

5 - Learning-based lossless light field compression

5.1 . Introduction

As presented in Chapter 3 LF compression has a rich spectrum of possible solutions. Among solutions, methods based on disparity compensation and view synthesis, especially those based on deep learning, showed the most promise. These methods radically reduce the number of reference views that need to be encoded, which effectively decreases the required bitrate, while at the decoder side, they leverage geometry information or high modeling power of deep learning to obtain high-quality reconstructs of the rest of the LF. In addition, a residual signal can be transmitted for further gains.

Motivated by the high reconstruction quality from a sparse set of references that practically offers accurate prediction at low bitrate demand, we explore this paradigm for lossless compression of LF images. In order to compensate for errors in the prediction and facilitate artifact-free reconstruction, the residual signal is afterward encoded. For the encoding of the residual signal, we drive motivation from the deep image modeling. Namely, deep generative models, more precisely likelihood-based models generative models, explicitly estimate the probability mass function of the underlying pixels, which can be coupled with adaptive arithmetic coding for lossless compression. This design has been recently employed for lossless compression of image data and point clouds, while the approach presented in this chapter is, to our knowledge, the first work applying deep conditional probability estimation in conjunction with view synthesis for lossless coding of LFs. The proposed method shows gains in terms of bitrate compared to state-of-the-art methods with competitive encoding and decoding time.

In this chapter, we aim at exploring the following questions:

1. How do recent traditional and learning-based image compression approaches for lossless compression behave on LF data?
2. How can we effectively utilize view synthesis in lossless coding of LF images?
3. How can we factorize views in the LF for a more accurate estimation of probabilities for coding?
4. How does lossless coding based on view synthesis and autoregressive modeling of probabilities compare to state-of-the-art methods?

In this chapter, we first mention relevant work to the proposed scheme in Section 5.2. Then, in Section 5.3 we explain the building blocks of the proposed scheme, the view synthesis approach and the entropy model, as well as the structure

of the proposed scheme. We describe the ablation studies that compare different variants of both base scheme and proposed scheme in Section 5.4, and conclude the section with the comparison of the base method and the proposed scheme. Section 5.5 contains a quantitative analysis of the proposed scheme with respect to state-of-the-art methods in terms of bitrate and execution time. The chapter conclusion is presented in Section 5.6.

5.2 . Related work

We note multiple sets of solutions related to the methodology proposed in this chapter. First, we note the methods for lossless compression of LF images. As covered in Section 3.5, we mention two groups of methods based on the representation of the input data: methods designed for the compression of raw lenslet images and methods for compression of rectified LF images, i.e., LFs obtained from raw sensor image through decoding procedures [27, 83]. Next, we consider view synthesis methods as the view synthesis is one of the main functional blocks designated for the prediction in our proposed scheme. Finally, we include autoregressive models that explicitly model the log-likelihood of image data and their extensions for lossless image compression.

5.2.1 . View synthesis methods

View synthesis is a method that allows estimating novel views from a set of reference views. Typically a scene geometry is estimated based on the reference set and used together to render views at a novel position in a scene. In LF compression scenarios, view synthesis promises the excellent potential to exploit the inter-view similarity between the views. Typically, a sparse set of reference views and corresponding geometry information is encoded and transmitted, and at the decoder side, the rest of the LF views are reconstructed using transmitted information. Recently with the widespread use of deep learning tools, many works on view synthesis appeared, which propose estimating scene geometry on the fly. Kalantari et al. [66] present the first work on view synthesis based on deep learning. They propose a network that consists of two sequential networks: the disparity network and the color network. The disparity network takes corner views of a LF image and the novel position of the view to be synthesized, and it estimates the disparity of the novel view with respect to the input views. The reference views are then back-warped to obtain the estimates of the novel view and merged by the color network to obtain the final estimate. Srinivasan et al. [130] tackle the problem of estimating the entire LF image from a single image. In particular, the authors estimate disparity maps of all views in a LF image with respect to the input image (positioned in the center of the LF) and warp it to generate a Lambertian LF image. The additional network follows to refine the predicted LF around occlusion and non-Lambertian effects. The drawback of the approach is the non-uniform distribution of quality of generated views across the LF; the quality reduces when

moving away from the center view. More recently, Navarro et al. [97] propose a novel view synthesis approach inspired by these two approaches. The authors estimate a novel view from the corner views as done in Kalantari et al. [66] but propose to estimate a disparity map of each corner view and merge warped corners using the weights estimated by a selection network. We select the method [97] which achieves superior performance compared to other methods and incorporate it into our scheme.

5.2.2 . Autoregressive models and lossless compression

An autoregressive model is a type of deep generative model with tractable likelihoods. Given a sequence of samples, the likelihood of the sequence can be decomposed as a product of distributions of each sample conditioned on previously processed samples. Recently, image-based autoregressive models based on deep learning successfully modeled image distribution. [99] propose PixelRNN method where an image is arranged in a sequence, and the probability distribution off each pixel in the sequence is estimated sequentially using Recurrent Neural Network (RNN). Furthermore, a variant based on CNN and kernel masking is introduced to preserve 2D relations between pixels. Due to the sequential processing nature of these methods, i.e., the number of network calls equals the number of sub-pixels, they are characterized by high complexity. PixelCNN++ [116] proposes various improvements to the PixelCNN architecture, including parametric modeling of the probability distribution and joint estimation of its parameters for each pixel which allowed to reduce the number of networks calls to the number of pixels in the input image. MS-PixelCNN [109] tackles the problem of the slow inference of PixelCNN and proposes a parallelized version by grouping particular sets of pixels and modeling them as conditionally independent. This strategy reduced the complexity to $\mathcal{O}(\log N)$ compared to the initial $\mathcal{O}(N)$ complexity of PixelCNN. Kolesnikov and Lampert [72] aim at improving the naturalness and global structure of generated images by including auxiliary information in the form of a quantized grayscale image.

Joining an entropy coding method, e.g., arithmetic coding, to mentioned likelihood-based models provides a framework for lossless compression. This observation has been exploited recently to develop deep learning-based methods for lossless compression. Mentzer et al. [87] propose a hierarchical approach with learned feature extractors which generate latent representations transmitted to the decoder. At the decoder, latent representations are processed by decoding blocks and are used to estimate the probability distributions necessary for decoding the latents at the higher level. In [88], the authors design a two-layer lossless compression method by leveraging the standard coding tool Better Portable Graphics (BPG). BPG is a lossy and lossless image compression scheme based on the HEVC [132]. It was selected due to its ability to faithfully reproduce the original image with high PSNR and effectively restrict the residual signal in a narrow intensity range around zero intensity. After encoding the input image with BPG, the residual between

the input and its processed variant is entropy coded using the probability distribution estimated by a neural network. The network provides probability distribution parameters for each pixel, allowing efficient, parallelized processing. These tools show similar or better performance compared to state-of-the-art image codecs such as BPG and Free Lossless Image Format (FLIF) [129].

5.3 . Proposed method

Our approach takes inspiration from learning-based lossless image compression [88] and image generation [109][72], which are combined with view synthesis [97] to enable the prediction of views from a set of reference views. Figure 5.1 presents the overall design of the proposed method. The View Synthesis module takes reference views and the input view position at the encoder side, and predicts the input view. Then, the prediction is subtracted from the input view. The obtained residual signal is entropy coded using an Arithmetic Coding (AC) provided with probability distributions computed on a per-pixel basis by the Entropy Modeling block given the prediction. At the decoder, we use the prediction from the View Synthesis module to estimate probability distributions for decoding and add the decoded residual signal to the prediction to reconstruct the input view. View Synthesis and Entropy Model are briefly described in the following subsections.

5.3.1 . View synthesis

To estimate a novel view, Navarro et al. [97] propose a CNN that consists of three distinctive blocks, a feature extractor, a disparity estimator and a view fusion. In more details, the view synthesis approach can be formalized as follows. Given a set of corner views $I_{\mathbf{c}} = \{I_{\mathbf{c}_i}\} = \{I_{0,0}, I_{0,N}, I_{N,0}, I_{N,N}\}$ of an LF image $L(\mathbf{x}, \mathbf{u})$ of angular size $(N+1) \times (N+1)$ where $\mathbf{x}$ and $\mathbf{u}$ denote spatial and angular positions of light rays, and the position of a view to be synthesized $\tilde{\mathbf{u}}$ the feature extraction network $\mathcal{F}_e$ computes the feature map of each corner view $I_{\mathbf{c}_i}$ independently:

$$F_{\mathbf{c}_i} = \mathcal{F}_e(I_{\mathbf{c}_i}, \tilde{\mathbf{u}}). \tag{5.1}$$

The network facilitates six convolutional blocks: a 2D convolutional layer, the ELU activation unit and the batch normalization layer, and two pooling layers.

The feature maps are then concatenated to generate a 3D vector $\mathbf{F} = (F_{0,0}, F_{0,N}, F_{N,0}, F_{N,N})$ and together with the position of the novel view $\tilde{\mathbf{u}}$ provided to the disparity network $\mathcal{F}_d$ that estimates the vector of disparity maps $\mathbf{D} = (D_{0,0}, D_{0,N}, D_{N,0}, D_{N,N})$ of the novel view:

$$\mathbf{D} = \mathcal{F}_d(\mathbf{F}, \tilde{\mathbf{u}}). \tag{5.2}$$

The network comprises six convolutional blocks, among which the first four employ dilated filters to increase the receptive field of the network, and a single convolutional block comprises a 2D convolutional layer and the tangent hyperbolic

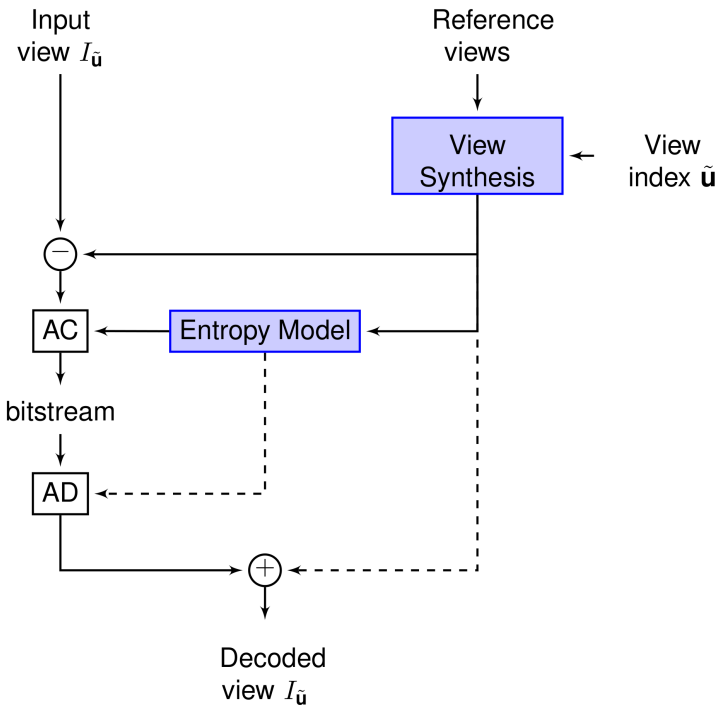


Figure 5.1: The block diagram of the proposed method. The View Synthesis block estimates a view for coding $I_{\tilde{u}}$. The prediction is provided to the Entropy Model block that estimates the probability distribution of the residual signal, which is encoded by the AC module using the predicted distribution. The decoder operates symmetrically with dashed lines illustrating the decoding pipeline. Given the estimated prediction and the probability distribution of the residual signal, the bitstream is decoded using the Arithmetic Decoding (AD) module, and the decoded residual signal is added to the prediction to obtain the final reconstruction.

activation function. The final activation function limits the output in the range $[-1, 1]$ which is multiplied by a constant d_{max} to procure final disparities in the range $[-d_{max}, d_{max}]$. The constant d_{max} is empirically set depending on the disparity values between neighboring views in the dataset at use and for the contents captured with a plenoptic camera it is sufficient to set d_{max} to 4.

Initial estimates of the novel view $W^{\mathbf{c}}$ are obtained by backward warping the corner views using the corresponding disparity maps,

$$W^{\mathbf{c}_i}(\mathbf{x}) = I_{\mathbf{c}_i}(\mathbf{x} + D_{\mathbf{c}_i} \cdot (\mathbf{c}_i - \tilde{\mathbf{u}})). \tag{5.3}$$

In the final stage of the scheme, the selection network $\mathcal{F}_s$ computes the contribution $M_{\mathbf{c}_i}$ for each initial estimate $W^{\mathbf{c}_i}$

$$\mathbf{M} = \mathcal{F}_s(\mathbf{W}, \mathbf{D}, \tilde{\mathbf{u}}). \tag{5.4}$$

where $\mathbf{M} = (M_{0,0}, M_{0,N}, M_{N,0}, M_{N,N})$ and $\mathbf{W} = (W_{0,0}, W_{0,N}, W_{N,0}, W_{N,N})$ are concatenated vectors of merging maps $M_{\mathbf{c}}$ and initial estimates $W^{\mathbf{c}}$. The merging parameters for each pixel are constrained to sum to 1 by computing the softmax

function along the pixels at the same position at four merging maps

$$M_{c_i}(\mathbf{x}) = \frac{e^{\beta V_{c_i}(\mathbf{x})}}{\sum_{\mathbf{c}_j \in \mathbf{c}} e^{\beta V_{c_j}(\mathbf{x})}} \tag{5.5}$$

where V_{c_i} is the feature map keeping the contribution of pixels at estimate $W^{\mathbf{c}_i}$ before the softmax function and β is learned variable which allows favoring a single pixel among four estimates which can be helpful in occluded regions. The final estimate of the novel view is obtained by merging initial estimates with each pixel in the final estimate $Y_{\tilde{\mathbf{u}}}$ being obtained as a weighted sum

$$Y_{\tilde{\mathbf{u}}} = \hat{X}_{\tilde{\mathbf{u}}}(\mathbf{x}) = \sum_{\mathbf{c}_i \in \mathbf{c}} M_{\mathbf{c}_i}(\mathbf{x}) W^{\mathbf{c}_i}(\mathbf{x}). \tag{5.6}$$

All blocks are trained jointly by minimizing the sum of L1 losses between the ground truth view and the predicted view and between the ground truth and predicted view gradients. See Section 5.3.3.

5.3.2 . Entropy model

Mentzer et al. [88] propose a Residual Compressor (RC) network, which we will denote as Entropy Model (EM), that takes a decoded input image $\tilde{X}$, compressed with BPG codec [10], and estimates a set of parameters that model the probability mass function of the residual signal R, the difference between the input image X and its decoded version $\tilde{X}$. The joint probability of the residual signal of an RGB image is defined as

$$p(R|\tilde{X}) = \prod_{\tilde{x} \in \tilde{X}} p(r_{\tilde{x}\mathbf{r}}, r_{\tilde{x}\mathbf{g}}, r_{\tilde{x}\mathbf{b}}|\tilde{x}), \tag{5.7}$$

where $r_{\tilde{x}\mathbf{r}}$, $r_{\tilde{x}\mathbf{g}}$, and $r_{\tilde{x}\mathbf{b}}$ denote intensities of color components of a residual pixel corresponding to a pixel $\tilde{x}$ in decoded image, and each pixel is modeled with an autoregression over color channel defining the joint probability of each pixel

$$\begin{aligned} p(r_{\tilde{x}\mathbf{r}}, r_{\tilde{x}\mathbf{g}}, r_{\tilde{x}\mathbf{b}}|\tilde{x}) &= p(r_{\tilde{x}\mathbf{r}}|\tilde{x}) \cdot p(r_{\tilde{x}\mathbf{g}}|r_{\tilde{x}\mathbf{r}}, \tilde{x}) \cdot p(r_{\tilde{x}\mathbf{b}}|r_{\tilde{x}\mathbf{r}}, r_{\tilde{x}\mathbf{g}}, \tilde{x}) \\ &= p_m(r_{\tilde{x}\mathbf{r}}|\tilde{\mu}_{\mathbf{r}}(\tilde{x}), \sigma_{\mathbf{r}}(\tilde{x})) \\ &\quad \cdot p_m(r_{\tilde{x}\mathbf{g}}|\tilde{\mu}_{\mathbf{g}}(\tilde{x}, r_{\tilde{x}\mathbf{r}}), \sigma_{\mathbf{g}}(\tilde{x})) \\ &\quad \cdot p_m(r_{\tilde{x}\mathbf{r}}|\tilde{\mu}_{\mathbf{b}}(\tilde{x}, r_{\tilde{x}\mathbf{r}}, r_{\tilde{x}\mathbf{g}}), \sigma_{\mathbf{b}}(\tilde{x})). \end{aligned} \tag{5.8}$$

As it can be seen in Equation 5.8, the autoregression is facilitated in the means of the probability distribution p_m which are updated based on the intensities of previously processed channels. More formally,

$$\begin{aligned} \tilde{\mu}_{\mathbf{r}}(\tilde{x}) &= \mu_{\mathbf{r}}(\tilde{x}), \\ \tilde{\mu}_{\mathbf{g}}(\tilde{x}, r_{\tilde{x}\mathbf{r}}) &= \mu_{\mathbf{g}}(\tilde{x}) + \alpha(\tilde{x}) \cdot r_{\tilde{x}\mathbf{r}}, \\ \tilde{\mu}_{\mathbf{b}}(\tilde{x}, r_{\tilde{x}\mathbf{r}}, r_{\tilde{x}\mathbf{g}}) &= \mu_{\mathbf{b}}(\tilde{x}) + \beta(\tilde{x}) \cdot r_{\tilde{x}\mathbf{r}} + \gamma(\tilde{x}) \cdot r_{\tilde{x}\mathbf{g}}. \end{aligned} \tag{5.9}$$

[10] Fabrice Bellard. BPG Image format. https://bellard.org/bpg/

Beside means, $\tilde{\mu_{\mathbf{r}}}$, $\tilde{\mu_{\mathbf{g}}}$ and $\tilde{\mu_{\mathbf{b}}}$, variances, $\sigma_{\mathbf{r}}$, $\sigma_{\mathbf{g}}$ and $\sigma_{\mathbf{b}}$, and coefficients of the autoregressive model, α, β and γ, RC estimates also a weight π of each component in the logistic mixture model $p_m = \sum_{k=1}^{K} \pi^k \cdot p_l(r_x|\tilde{\mu}^k, \sigma^k)$, with p_l being the logistic distribution:

$$p_l(r|\mu, \sigma) = (\text{sigmoid}((r + 0.5 - \mu)/\sigma) - \text{sigmoid}((r - 0.5 - \mu)/\sigma)).$$

Note, the number of mixtures K is fixed and for a mixtures with K components and 3-channel input, as presented before, RC estimates $3 \cdot K + 4 \cdot K + 5 \cdot K$ parameters per pixel (wherein each term corresponds to a factor in Equation 5.8).

5.3.3 . Loss function

To learn the parameters of the view synthesis block, we use the sum of L1 loss between ground truth view and the predicted view and weight L1 loss between gradients of ground truth view and the predicted view:

$$J(\eta) = \frac{1}{N} \sum_{X \in B} \|X - \tilde{X}\|_1 + \frac{1}{2} \|\nabla X - \nabla \tilde{X}\|_1, \tag{5.10}$$

where N is the number of ground truth images X in mini-batch B and corresponding predictions $\tilde{X}$, while ∇ denotes gradient computation using Sobel filter.

Similar to previous works on learned image compression, we aim to reduce the cross-entropy between the real distribution of the encoding representation and estimated model, e.g., $p(R|\tilde{X})$. Reducing the cross-entropy will require fewer bits to encode the representation at hand. In our scenario, given an estimate of a to-be-encoded view generated by the view synthesis module and the corresponding residual signal, we minimize the cross-entropy loss

$$J(\theta) = -\sum_{\tilde{X} \in \tilde{B}} \log_2 \left(p(R|\tilde{X})\right), \tag{5.11}$$

where θ are learnable parameters of the neural network, and $\tilde{X}$ and R are synthesized training samples and their corresponding residual samples in a mini-batch $\tilde{B}$.

5.3.4 . The architecture

The overview of the coding pipeline was presented in Figure 5.1. For view synthesis we employ the method described in Chapter 5.3.1, while the entropy model is described in Chapter 5.3.2. For entropy model, we utilize the same architecture as proposed in [88] while changing the number of filters in intermediate layers, i.e., $C_f = 64$, and setting the number of residual blocks to 8. Note, the main difference compared to the work of [88] is using the *synthesized view* $\hat{I}_{\tilde{\mathbf{u}}}$ as input and compute the residuals with respect to it. We denote this pipeline as *Base*.

We propose to extend the *Base* architecture by including spatial autoregression. *Base* proved very efficient in learning the relation between different channels, which

could also be leveraged in spatial domain considering the high correlation between neighboring pixels. Furthermore, the view synthesis method is based on disparity compensation making it more likely to generate errors concentrated around areas with occlusions and errors in high-frequency areas. Including spatial autoregression, the entropy model could compensate for the errors in prediction by leveraging decoded, neighboring, spatial pixels. Therefore, the proposed mechanism should improve the local estimation and reduce the required bitrate. In order to facilitate spatial autoregression, we propose to divide pixels into coding groups and to use previously decoded groups to estimate the parameters for the following groups.

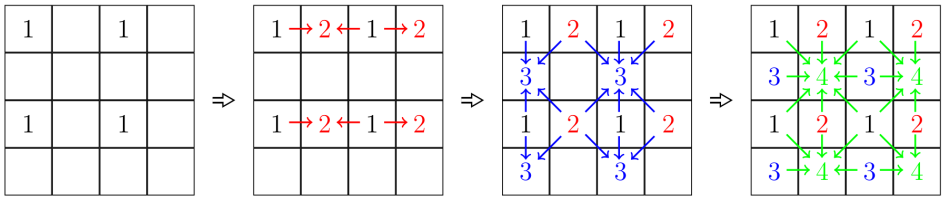


Figure 5.2: The grouping introduced in proposed architecture. Each number represents a group, while the arrows denote dependence between groups. E.g., the pixels in the third group rely on pixels from the first and the second group for modeling.

The idea is illustrated in Figure 5.2, wherein the image is divided into four groups, and each pixel group depends on pixels from previous groups. For clearer representation, the relations exist only among the closest, neighboring pixels, but in practice, all pixels from previous groups can contribute to pixels in a current group. Changing the number of groups makes it possible to trade-off between complexity/execution time and bitrate cost. In the *Base* method, all pixels are considered conditionally independent, i.e., belonging to a single group, and estimated in parallel. Increasing the number of groups makes it possible to improve performance by introducing dependence to previously decoded groups. In the extreme scenario each pixel presents a single group as proposed in PixelRNN [99] and PixelCNN++ [116]. The latter scheme should offer the best coding performance at the cost of high computational complexity as the network calls would scale linearly with the number of pixels. In our experiments, we use four groups containing every other pixel in the horizontal and vertical directions, similar to a checkerboard pattern.

Figure 5.3 depicts the block diagram of the encoder of the proposed framework. The input view x is divided into four groups as denoted by pixels classes and encoded sequentially in the ascending order of the class set. In the case of the first group, The EM takes the input view prediction and estimates the parameters of the mixture of logistic distribution for each pixel in the first group. These parameters are used to model the probability mass function for arithmetic coding. For the following groups, the prediction of the view and previously decoded groups are concatenated and provided to EM. We compensate for the reduced spatial resolution of decoded information by spatial upsampling using simple repeating to

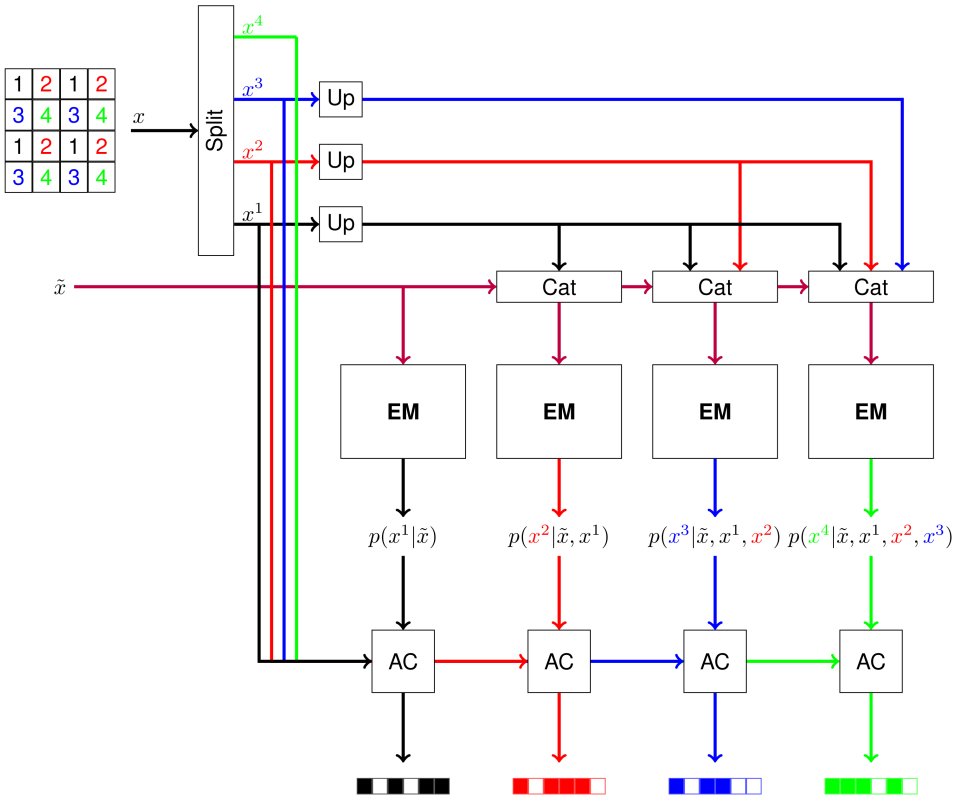


Figure 5.3: The encoding procedure of the proposed spatial autoregression with four groups. The prediction of the current view $\tilde{x}$ and previously decoded spatial groups are provided to the EM networks that estimate parameters of the probability mass function for each pixel in the current group. The bitstream of each group is obtained by encoding each group using computed probabilities and arithmetic coder.

generate a 2×2 tile. As the number of previously decoded groups varies, the input to EM has different sizes. Therefore, each EM network is modified to process new input by changing the very first layer according to the input size, and to obtain the output of the spatial size of the groups.

5.4 . Experiments

5.4.1 . Datasets

Training We trained our models on on 3323 LF images from *Flowers* dataset [130]. Raw LF images are decoded with Lytro Power Tool (LPT)[11] which demosaics Bayer sensor and calibrates lenslets. The resulting LFs have spatial size 541×376 and 14×14 angular resolution. As corner views suffer from strong vignetting artifacts, we use only central 8×8 set of views.

Testing For evaluation purposes we use 12 LF images, initially proposed in the

[11] https://github.com/kmader/lytro-power-tools, accessed on Nov. 10th, 2020.

First LF coding challenge [111], from *EPFL* dataset [110]. We use LPT to decode lenslet images as it is done for the training dataset. In addition, we apply gamma correction ($\gamma = 0.45$) and quantize LF images to 8 bits. For the convenient evaluation of methods that require that the spatial resolution is of a particular multiplier, we cropped LFs views to the spatial size of 320×512 pixels. Furthermore, we crop LFs in angular dimensions to 7×7, which has a central view.

5.4.2 . Training procedure

The training procedure starts by training the view synthesis method. We extract 100 images from the training set for validation purposes while the rest of the LFs are used for training. At each training iteration, training samples are randomly cropped to the spatial size 192×192, the angular position of the view to-be-estimated is randomly selected, while excluding the angular positions of the reference views (corners views or the middle view on the periphery of LF), and the samples are augmented by applying gamma correction with the gamma value randomly selected from the range $[0.4, 1.0]$. We observe the convergence of the model on the center views of the validation set, wherein we use the full spatial size and randomly select the gamma value from the range $[0.4, 0.5]$. We use ADAM optimizer [71] with default parameters and select the batch size of 10. After the convergence, we fix the parameters of the view synthesis network and train the entropy model network.

Similar to the previous training procedure, we randomly select LF patches of size $7 \times 7 \times 128 \times 128$, the positions of the reference views, and the target view. For the position of the reference views, the selection chose reference views at the boundary of a training patch (the maximum possible baseline) or at a boundary of a random quadrant of angular size 4×4. Then, we randomly perturb color channels to increase the training dataset's color diversity and finally apply gamma correction selected from the interval $[0.4, 1.0]$. We use a batch size of 16 and Adam optimizer [71] to update the weights.

5.4.3 . Scheme ablations

The main comparison aims at evaluating performance between the *Base* method and the proposed method that facilitates four spatial groups. In addition, we evaluate different aspects of the two schemes, such as the arrangement of reference views and single and multiple hierarchical levels, and finally, we select a hybrid scheme comprised of best practices. The performance of different variants of *Base* method, introduced in Chapter 5.3.4, is presented in Table 5.1, while the comparisons for the proposed method are presented in Table 5.2.

View arrangements. Beside evaluating native *Corner* arrangement where corner views are selected as reference views, we also examine *Cross* arrangement, which proved to be superior to the *Corner* arrangement in recent work [95]. The column marked as *Single* presents the performance of the proposed method where four reference views of the two arrangements are independently encoded and used

Table 5.1: The performance evaluation of variants of *Base* method presented in terms of bpp.

	Single		Hierarchical		
Sequence	Corner	Cross	Corner	Cross	Hybrid
Bikes	6.85	6.67	6.29	6.43	6.22
Danger	7.62	7.48	6.99	7.16	6.89
Flowers	7.56	7.39	6.88	7.08	6.79
Pillars	6.98	6.83	6.41	6.52	6.33
Vespa	7.09	6.59	6.18	6.31	6.10
Ankylosaurus	5.11	5.08	4.88	4.96	4.84
Desktop	7.37	7.10	6.67	6.85	6.62
Magnets	5.18	5.15	4.93	5.02	4.90
Fountain	7.10	6.66	6.24	6.32	6.13
Friends	6.34	6.42	6.10	6.25	6.04
Color Chart	7.40	6.60	6.11	6.29	6.00
ISO Chart	6.79	6.38	6.04	6.00	5.92
Average	6.78	6.53	6.14	6.27	6.06

to estimate all the other views. In terms of bitrate performance the *Cross* arrangement consistently outperform *Corner* arrangement when a single hierarchical level is used as presented in the first two columns of Table 5.1. A possible reason that contributes to the outcome is the narrower baseline between reference views which makes the rest of the LF views closer to the reference views in *Cross* arrangement compared to the scenario with *Corner* arrangement.

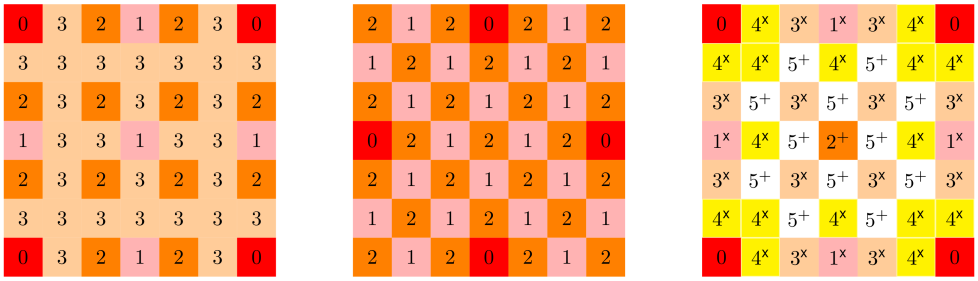


Figure 5.4: Hierarchical levels for prediction of views with Corner arrangement (left) and Cross arrangement (middle). At each level i, views can be predicted from decoded views from previous levels, i.e. $i_{prev} < i$. Prediction scenarios applied in Hybrid variant (right). The number denotes the prediction level i while the superscripts denote the arrangement of the reference views used for the prediction, i.e., x and $+$ denote Corner and Cross arrangements.

Hierarchical levels. In the view synthesis task, the whole LF is typically synthesized from a particular set of reference views. Note, this scenario corresponds to the *Single*, and it is the most efficient from a random access perspective for the given view synthesis module. On the other side, in lossless compression, as

Table 5.2: The performance evaluation of variants of the proposed method presented in terms of bitrate (bpp).

	Single		Hierarchical		
Sequence	Corner	Cross	Corner	Cross	Hybrid
Bikes	6.51	6.32	5.90	5.97	5.81
Danger	7.23	7.02	6.51	6.59	6.39
Flowers	7.21	6.96	6.42	6.53	6.31
Pillars	6.81	6.58	6.08	6.15	5.98
Vespa	6.49	6.21	5.88	5.91	5.79
Ankylosaurus	5.14	5.06	4.86	4.90	4.82
Desktop	6.84	6.63	6.24	6.30	6.15
Magnets	5.16	5.08	4.90	4.94	4.86
Fountain	6.56	6.37	6.01	6.01	5.90
Friends	6.07	5.90	5.54	5.60	5.45
Color Chart	5.93	5.65	5.43	5.45	5.35
ISO Chart	6.33	6.11	5.80	5.75	5.69
Average	6.36	6.16	5.80	5.84	5.71

views become available after decoding, it becomes possible to dynamically select reference views and leverage reduced baseline between views for improved prediction. However, a potential drawback could be reduced random access capabilities as later views depend on previously encoded/decoded views. We compare both cases, *Single* method mentioned before, and *Hierarchical* where, for *Corner* view arrangement, three levels are incorporated as depicted on the left part of Figure 5.4. In the first level, the method processes the middle views on the periphery of the LF and the center view. Then, a subset of views in each quadrant is estimated in the second level. Finally, the other views are predicted from their neighboring four views in the third level. As observed in Table 5.1 employing *Hierarchical* brings 0.72 bpp compared to *Single* scheme for *Corner* arrangement.

View arrangements in *Hierarchical* scheme When the number of hierarchical levels increases, we have noted improved performance for *Corner* arrangement. In the case of *Cross* arrangement, the trend is the same, but we observe smaller gains compared to *Corner*. The gains increase as much as to achieve an improved performance of the *Corner* arrangement compared to the *Cross* arrangement, which is the opposite behavior compared to *Single* scheme. The superiority of the *Corner* approach comes from a higher level of flexibility in selecting reference views. As we can note in Figure 5.4 (left) *Corner* arrangement supports three hierarchical levels with the majority of the views being predicted at the highest hierarchical level, i.e., level 3, where the baseline between the reference views is the narrowest resulting in highest quality reconstructs. In the case of *Cross* arrangement, Figure

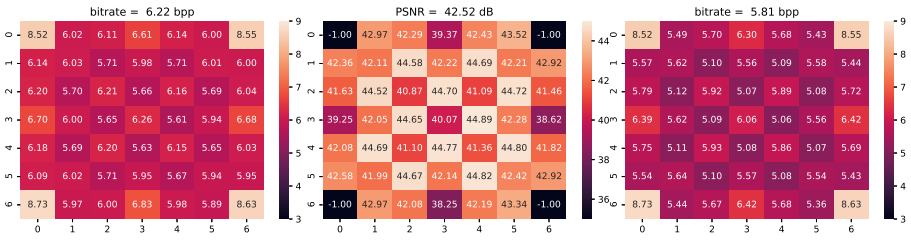
5.4 (middle), only two hierarchical levels are facilitated with most of the views included in one of the two. Here, almost half of the views (level 1) are estimated from the wide baseline, which reduces the overall quality. The lack of flexibility can also be attributed to a required odd angular size to select reference views. For instance, in an LF of 4×4 angular size, in order to utilize *cross* arrangement, it would be necessary to operate on LFs of 3×3 angular size. Nevertheless, as seen in the scenario with a single hierarchical level, the cross arrangement provides superior prediction.

The best practice By considering presented features, namely superiority of *Cross* arrangement at a single hierarchical level and improved flexibility of the *Corner* arrangement, we propose a *Hybrid* approach which allows to utilize the *Cross* arrangement in *Hierarchical* scheme effectively. The new selection is illustrated in Figure 5.4 (right), where the numbers denote hierarchical levels while the superscripts denote arrangements used for the prediction. E.g., "2^{+}" denotes that *Cross* arrangement was used to estimate this view and that it belongs to the second hierarchical level. The selection interleaves two arrangements across hierarchical levels allowing to practically navigate LF views so that fewer views are estimated with wider-baseline reference views while still leveraging superior prediction capabilities of *Cross* arrangement.

To compute the average bitrates in Table 5.1 and Table 5.2 the reference views are encoded independently using the Learned LossLess image Compression (L3C) method.

***Base* scheme compared to the proposed scheme**

Finally, we compare the performance of the proposed scheme and *Base* scheme. By comparing results presented in Table 5.1 and Table 5.2, we note that the different variants of the proposed scheme outperform their corresponding counter part in *Base* scheme. For the most promising variant, *Hierarchical-Hybrid* proposed scheme offers on average gains of 0.36 bpp.

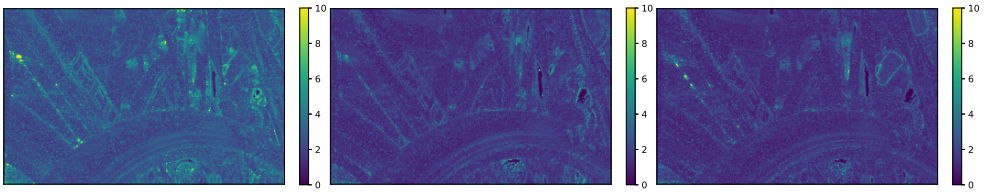


(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

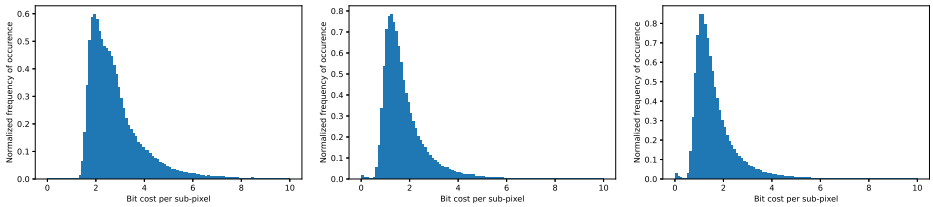
Figure 5.5: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Bikes*.

We also explore the gains in more detail by considering the contribution of every view in a LF image. Figure 5.5 shows per-view bitrate comparison for *Bikes*. The middle part of Figure 5.5 shows performance of view synthesis block operating in *Hierarchical-Hybrid manner*. As expected, the reconstruction quality positively

correlates with the decrease of the reference views baselines and increases towards higher hierarchical levels. The evaluation of bitrates in (a) and (c) in Figure 5.5 shows that more accurate prediction provides better context for entropy modeling. Furthermore, we observe considerable gains of the proposed scheme, especially with the increasing reconstruction quality. The results for the rest of the test LFs are available in Appendix B.2.



(a) Maps of bit costs (entropy) for each color channel. Average bit costs are 2.84, 1.77 and 1.65 bits per sub-pixel.



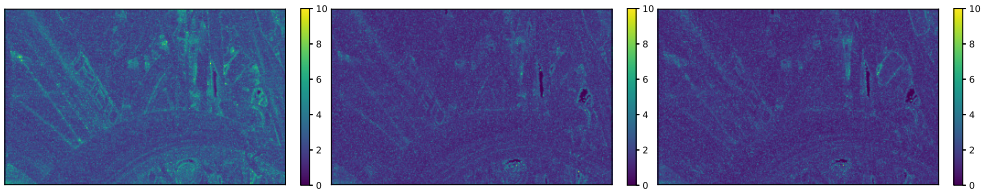
(b) Normalized histograms corresponding bit cost maps.

Figure 5.6: Intermediate results of probability distribution modeling using *Base* scheme for the center view of content *Bikes*.

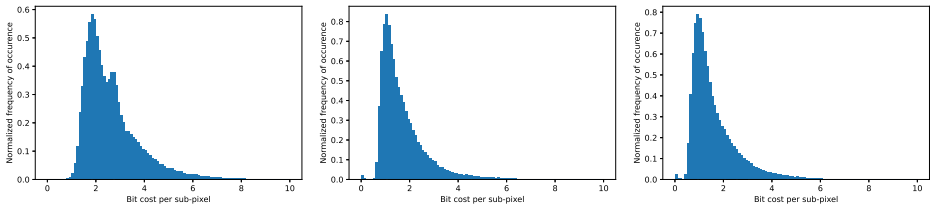
In Figure 5.6 and Figure 5.7 we show the distribution of bit costs for each color channel of center view, $(3, 3)$, in *Bikes* content, obtained with **Base** scheme and proposed scheme, respectively. Higher bit cost can be observed for the first color channel while the latter channels have considerably lower demands, which shows the effectiveness of the autoregressive model in representing the relations between color channels. It can be further noted that pixels around edges and fine objects demand more bits suggesting less accuracy in the entropy model.

Results in Figure 5.8 and Figure 5.9 show lower bit cost around edges and in fine-detail areas thanks to the improved prediction of view synthesis block. The improvement can also be observed in the histograms, where the distribution translates towards lower bit costs.

***Base* scheme with matched complexity of proposed scheme** As the proposed method facilitates four entropy models, it has four times more parameters compared to *Base* method employing a single entropy model. Therefore, in order to verify that the gains in the proposed method are achieved via grouping and auto-regressive modeling, we also train *Base* network with approximately the same number of parameters and denote it *Base-128* as the number of intermediate fil-

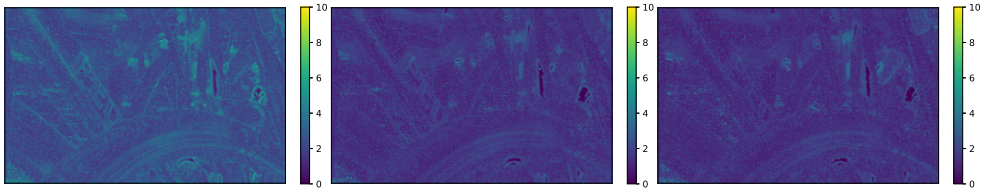


(a) Maps of bit costs (entropy) for each color channel. Average bit costs are 2.68, 1.73 and 1.64 bits per sub-pixel.

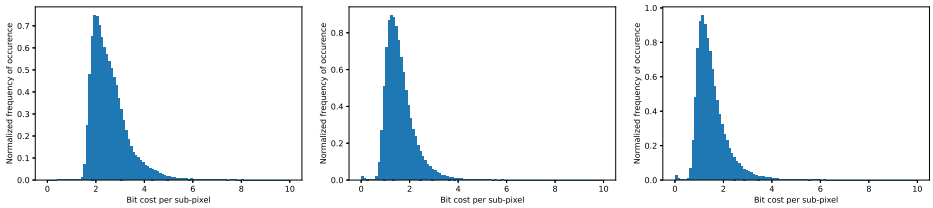


(b) Normalized histograms corresponding bit cost maps.

Figure 5.7: Intermediate results of probability distribution modeling using Proposed scheme for the center view of content *Bikes*.



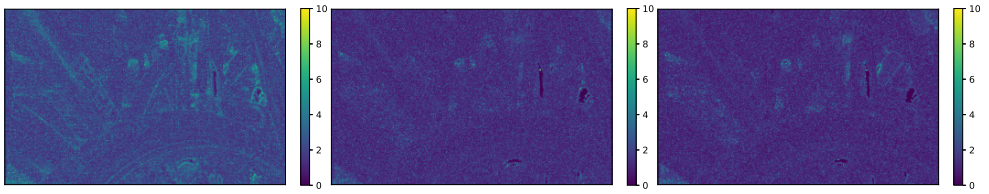
(a) Maps of bit costs (entropy) for each color channel. Average bit costs are 2.58, 1.60 and 1.52 bits per sub-pixel.



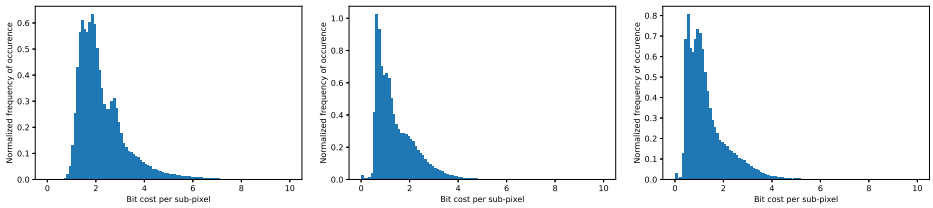
(b) Normalized histograms corresponding bit cost maps.

Figure 5.8: Intermediate results of probability distribution modeling using *Base* scheme for view (1, 2) of content *Bikes*.

ters is increased to 128. *Base-128* shows improved performance compared to *Base*, but still, it is inferior compared to the proposed scheme, which supports the benefit of utilizing the spatial autoregression model. The performance of *Base-128* is presented in Table B.1.



(a) Maps of bit costs (entropy) for each color channel. Average bit costs are 2.29, 1.45 and 1.35 bits per sub-pixel.



(b) Normalized histograms corresponding bit cost maps.

Figure 5.9: Intermediate results of probability distribution modeling using Proposed scheme for view (1, 2) of content *Bikes*.

5.5 . Results

5.5.1 . Compression performance

In order to evaluate the proposed method, we select and compare it to two groups of approaches: general lossless schemes for image and video compression, including both standard codecs and learning-based methods, and approaches designed explicitly for lossless compression of LFs.

Among general lossless schemes, we select HEVC [132]. In this case, a LF image is first reshaped into a pseudo-video sequence following a serpentine scan order and encoded in the lossless mode in addition to the Main-RExt profile. We use HM v16.22 implementation. Furthermore, we also evaluate the test LFs on FLIF codec [129] and JPEG XL [6] as they are typically selected in lossless compression literature.

L3C [87] and RC [88] represent learning-based methods for lossless image compression. RC method is the most relevant work as the approach is incorporated in the proposed method. Beside independent encoding of each view, we also consider coding LFs in the lenslet representation. As the angular and spatial information is interleaved on a 2D grid, image codecs can exploit to some extent the angular correlation in LF images.

Among the LF lossless coding approaches we select EPIC [93]. As *EPIC* was designed to operate independently on color channels in the YCbCr color space, we employ Reversible Color Transformation (RCT) [62] to input RGBs. In addition, we also evaluated the approach [125] with the publicly available software. However,

the results appear considerably higher compared to other methods.

Table 5.3: The performance evaluation in terms of bpp of image compression tools applied on each view separately.

	L3C	RC	FLIF	JPEG XL
Average	8.07	8.18	7.90	7.61

Table 5.3 shows the performance of the traditional and learned codecs applied independently on LFs views. This table represents a baseline for the scenario wherein the angular correlation is not exploited. We notice that RC and L3C obtain worse performance compared to standard approaches. Although the comparison in the respective works shows competitive or improved performance with standard methods, this seems not to be the case for our dataset. This might be due to the domain shift between the original training data of these methods and the lenslet test data. In addition, the results also suggest the presence of some structures in the extracted LF views not learned from traditional images.

Table 5.4: The comparison of the proposed method and available methods from literature in terms of bitrate (bpp).

	General lossless schemes					LF lossless schemes		
Sequence	HEVC	L3C	RC	FLIF	JPEG XL	CMS	EPIC-RCT	Proposed
Bikes	6.69	7.28	7.54	6.19	5.95	10.42	6.14	5.81
Danger	7.19	7.90	8.03	6.70	6.49	11.43	6.71	6.39
Flowers	7.08	8.00	8.27	6.84	6.49	10.96	6.67	6.31
Pillars	6.61	6.93	7.52	6.01	5.84	9.97	6.33	5.98
Vespa	6.64	7.12	7.30	6.10	6.03	10.01	5.98	5.79
Ankylosaurus	5.43	5.84	5.79	4.77	5.48	7.14	4.83	4.82
Desktop	6.76	7.18	7.56	6.22	6.33	9.46	6.26	6.15
Magnets	5.49	6.03	5.92	4.87	5.60	7.10	4.80	4.86
Fountain	6.71	7.48	7.35	6.14	6.14	10.15	6.03	5.90
Friends	6.10	6.50	6.89	5.47	5.38	8.46	5.69	5.45
Color Chart	6.00	6.57	6.57	5.35	5.69	7.46	5.37	5.35
ISO Chart	6.42	6.47	6.22	5.39	5.37	9.21	5.59	5.69
Average	6.43	6.94	7.08	5.84	5.90	9.32	5.87	5.71

Compression performance of the proposed scheme compared to state-of-the-art methods is presented in Table 5.4. First, we can observe consistently improved performance of L3C, RC, FLIF and JPEG XL, compared to scenario where they were applied independently to LF views. Even though the lenslet representation has different characteristics than natural images, improved performance compared

to independent views encoding suggests that both traditional and learned coding tools effectively exploit correlation in interleaved spatial and angular structures. Moreover, the gains of the standard coding tools are larger compared to L3C and RC, $> 28\%$ achieved by the traditional tools compared to $\sim 15\%$ achieved by the learned coding tools, and suggest a potential improvement in the performance by including LF data in training procedure. Then, we note that on average, proposed method outperforms FLIF, JPEG XL, EPIC by 2.23%, 3.22% and 2.73%, respectively. Interestingly, FLIF and JPEG XL perform quite well compared to the other two methods, which are specifically designed to exploit the correlation in LFs. It is highly likely that due to the small baseline between the views and small angular size, the lenslet format allows the efficient exploitation of spatial and angular similarities. HEVC is inferior compared to later approaches. This result aligns with the literature results.

A per-content comparison shows that the proposed method outperforms other methods on most sequences. The exceptions are less natural sequences such as *Ankylosaurus*, *Magnets*, and the two charts. We note various potential reasons for this behavior. Sequences like charts are mainly flat, which is a challenging content for geometry estimation. Moreover, the noise is strongly present, and the color constancy between the reference and target views is lacking. On the other side, object-rich sequences with diverse geometry content suit the proposed method exceptionally well, as it can be observed in the performance of *Bikes*, *Fountain* and *Friends*.

5.5.2 . Runtime

In Table 5.5, we report the encoding and decoding times of different codecs. The proposed method has, by construction, a symmetric processing workflow at encoder and decoder sides, which implies similar complexity at encoder and decoder, and execution time. The same observation holds for EPIC-RCT. These two approaches are especially efficient at encoding compared to conventional methods such as FLIF, JPEG XL and HEVC. On the contrary, the latter methods are designed for efficient decoding and show better execution time than the proposed method. Finally, we note improved processing time of the *Base* method insinuating the trade-off between the execution and the performance. Finally, we emphasize that the reported times represent a lower bound as they do not count the time needed to process the initial four reference views.

Table 5.5: The comparison of total LF encoding and decoding times presented in minutes.

Method	HEVC	FLIF	JPEG XL	EPIC-RCT	Proposed	Base
Encoding	6.47	0.62	0.34	0.13	0.17	0.14
Decoding	0.05	0.14	0.07	0.13	0.17	0.14

5.6 . Conclusion

This chapter proposes a learning-based method for lossless compression of LF images. The method consists of two learned functional blocks: a view synthesis block that predicts a current view and an entropy model which uses the prediction as a context to build probability distribution for arithmetic coding of the residual signal (difference between the input view and its prediction). An autoregressive model defines dependencies among color channels, with the latter channels being conditionally dependent on the prior channels. In addition, we introduce a spatial autoregressive model that operates on groups of pixels. Like the channel-wise autoregressive model, latter groups of pixels are conditionally dependent on the preceding groups. The proposed method outperforms state-of-the-art methods in terms of bitrate while maintaining low computational complexity.

Some results are associated with the following publication:

- M. Stepanov, M. U. Mukati, G. Valenzise, S. Forchhammer and F. Dufaux, "Learning-based lossless light field compression," *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1-6, doi:10.1109/MMSP53017.2021.9733637.

We conclude the chapter by explicitly answering initially posed questions in the Introduction:

1. How do recent traditional and learning-based image compression approaches for lossless compression behave on LF data? *We note that applying traditional image coding tools independently on each SAI is quite inferior to schemes designed to exploit correlation in LF images. Surprisingly, when the LF images are provided to these codecs in lenslet representation, the performance increases significantly, even providing competitive performance compared to state-of-the-art methods designed for LF structure. Still, it should also be taken into account that the test samples had somewhat limited angular size and that traditional coding tools are likely not to perform as well as when the angular size increases. Experiments also show that learned codecs are inferior to the traditional ones, which is likely due to the training dataset that considered only images.*

2. How can we effectively utilize view synthesis in lossless coding of LF images? *LF coding methods based on view synthesis have already shown vast potential. In the case of lossless compression, the decoded views are identical to the input views, allowing improved prediction performance compared to, e.g., lossy coding. Moreover, a hierarchical selection of the reference views allows to reduce the baseline between the reference views and obtain a better prediction.*

3. How can we factorize views in the LF for a more accurate estimation of probabilities for coding? *In our experiments, we observed that prediction is quite challenging not just due to occlusions and non-Lambertian surfaces but also due to the noise and lack of color constancy between different SAIs. Therefore, it is necessary to adopt a progressive encoding/decoding so that previously processed parts can be used to compensate for the prediction errors. We considered using four groups where each group relies on previously decoded groups to improve its estimation.*

4. How does lossless coding based on view synthesis and autoregressive modeling of probabilities compare to state-of-the-art methods? *Learned lossless LF codec based on view synthesis and autoregressive modeling offers state-of-the-art coding performance. Besides coding performance, the lossless codec offers a highly competitive execution time.*

6 - Distributed light field coding

6.1 . Introduction

In standard coding tools, the encoder side usually exploits the redundancy present in the input data. Generally, the encoder is highly complex and demands costly computational resources. Contrary to these schemes, in Distributed Source Coding (DSC) the correlation is exploited at the decoder side, which effectively lifts the complex computations from the encoder. From the LF acquisition perspective, DSC can thus release the burden of the camera processor while still guaranteeing efficient data transmission. DSC is based on the theoretical results of Slepian-Wolf, and Wyner-Ziv (WZ) theorems [26]. According to them, two correlated sources can be coded with a total rate lower bounded by their joint entropy (after quantization), even if only one of the two sources is available at the decoder.

Conventional video coding is designed as a hybrid block-based scheme including prediction, transformation, quantization, and entropy coding [132]. The inclusion of the prediction at the encoder side is the primary reason for the superior coding performance compared to transform-based coding. This framework fitted to a broadcast scenario provides efficient decoding at the cost of heavy computation at the encoder. On the contrary, there are scenarios where it is more desirable to have a power-efficient encoder and transfer most of the computation to the decoder side. These scenarios typically include low-power camera systems, for example, in wireless networks or multi-view video entertainment [101].

In this chapter, we present a distributed coding scheme developed on top of the latest state-of-the-art method in DMVC [117] to improve the estimation at the decoder side. More precisely, we replace the typically employed optical flow [79] or overlapped block motion compensation [51] to generate SI with a learning-based view synthesis approach, which estimates the scene geometry and inpaints occlusions, to obtain higher-quality estimates. We compare to distributed LF coding approaches based on optical flow to generate SI in two scenarios: PVS and SAI representation, motivated by DVC and DMVC, respectively. Furthermore, we show that a view synthesis approach that efficiently leverages the LF structure to synthesize intermediate views can provide competitive coding performance even if only a few key views are transmitted. This framework significantly reduces the computation requirements on the encoder side. Further on, we leverage deep learning approaches for better estimation of SI in the distributed coding scenario. More precisely, we improve the view synthesis performance by considering different arrangements of the reference view, and we propose a deep learning-based approach to estimate the residual signal.

In this chapter we explore following research questions:

- How can we employ DSC paradigm for LF data?
- Can we leverage deep learning methodologies to improve coding performance of DSC pipeline?
- How well view synthesis performs as the prediction block in DSC?
- How well can the residual signal be modeled using the deep learning methodology?
- How DLFC enhanced with deep learning-based functional blocks compares to state-of-the-art methods?

This chapter is organized as follows. We start with a background on DSC in Section 6.2. We mention related works in Section 6.3 including the coding of different visual modalities using DSC and deep learning-based view synthesis approaches. In Section 6.4, we explain our proposed variations for the view synthesis network and the architecture for residual signal modeling. We present the quantitative analysis of the proposed scheme in Section 6.6 where we compare the proposed scheme to state-of-the-art methods and the conventional coding tools. Finally, Section 6.7 concludes the chapter.

6.2 . Background

DSC denotes the coding of two or more correlated sources with a particular system design. Namely, each source is independently coded using a separate encoder, while received bitstreams are processed jointly at the decoder side to exploit statistical dependencies. This is illustrated in Figure 6.1. Two statistically dependent sources, X and Y are encoded with a particular encoder achieving rates $R_X \geq H(X)$ and $R_Y \geq H(Y)$, with $H(X)$ and $H(Y)$ denoting entropies of corresponding sources. With their independent decoding the total achievable rate would be $R = R_X + R_Y$. However, Slepian-Wolf (SW) theorem shows that with a joint decoding achievable rate can reach the joint entropy of the two sources $H(X, Y)$ even though the sources were separately encoded. SW theorem derives a lower bound on achievable rates for DSC scenario represented with a set of inequalities:

$$R_X \geq H(X|Y), R_Y \geq H(Y|X), R = R_X + R_Y \geq H(X, Y)$$

where $H(X|Y)$ is conditional entropy of X given Y.

Later on, WZ theorem extended this result for the lossy scenario when SI is available at the decoder side. A sequence generated by source X is encoded without input from SI Y while at the decoder, the SI Y is used to decode $\hat{X}$ which is a reconstruction of X with allowed distortion D. WZ theorem shows that the achievable rate of the WZ bitstream under allowed distortion D, $R^{WZ}_{X|Y}(D)$, is greater or equal to the achievable rate of the coding of X with SI available at

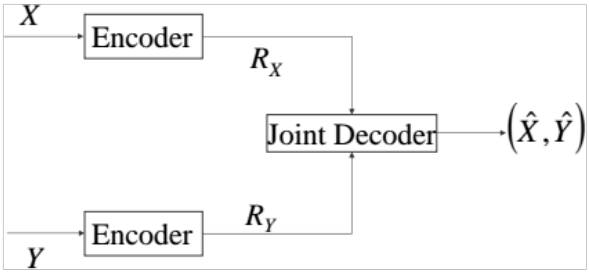


Figure 6.1: DSC pipeline in the scenario with two separate sources.

encoder side $R_{X|Y}(D)$. Moreover, it shows that $R_{X|Y}^{WZ}(D)$ reaches $R_{X|Y}(D)$ is the sources are jointly Gaussian, and the distortion is measured using Mean Squared Error (MSE).

Current video coding standards, e.g., MPEG schemes, recommend a highly efficient encoder and a simple decoder. This design is motivated by the applications the codec should serve; in broadcasting scenarios or for on-demand streaming scenarios, a video is compressed once and decoded many times. However, facilitating a complex encoder might not be feasible for some applications. Instead, the system should facilitate a low-complexity encoder and run heavy computations at the decoder. Examples of these systems include low-power sensor networks and wireless video surveillance cameras. Furthermore, in the systems with low complexity at both ends, the conventional and the distributed systems could be potentially combined so that the capture side utilizes the distributed coding while the receiver uses traditional coding with fast decoding capabilities. The capture would, in that scenario, transmit information to a heavy computational center which would decode captured information and re-encode using a standard codec.

In practical Distributed Video Coding (DVC) [43] schemes, a video sequence is divided into Group of Pictures (GOPs) whereas the first frame of each GOPs is referred to key frame, while the rest of GOPs consists of WZ frames. Key frames are encoded using traditional, hybrid coding schemes. WZ frames are on the other side, encoded using pixel-domain or transform-domain encoding characterized by considerably simpler encoding than traditional codecs. In the case of pixel-domain encoding, frame pixels are uniformly quantized and, in sufficiently large blocks, provided to the SW encoder. Transform-domain encoding applies a decorrelating transform, e.g., DCT, quantizes the coefficients, and after grouping, same-band coefficients sends the coefficients to the encoder. The SW encoder is based on rate-adaptive channel code, e.g., Low-Density Parity Check Accumulate (LDPCA) code, and generates so-called parity or syndrome bits. In addition, Cyclic Redundancy Check (CRC) of the quantized information is sent to help to decode. In the literature, transform-domain encoding exhibits superior performance compared to pixel-domain encoding. In either scenario, generating parity bits using channel codes is computationally much simpler than the prediction mechanism employed

in traditional encoders.

On the decoder side, key frames are decoded following the framework of the corresponding traditional, hybrid scheme. The decoding of each WZ frame starts by generating SI. Namely, SI generation block uses, e.g., motion estimation and compensation techniques to estimate a WZ frame based on previously decoded frames. Also, it estimates the parameters of a correlation model between WZ frame and its estimate. Typically, the Laplacian distribution is assumed as the correlation model. Based on these estimates and received syndrome bits, SW decoder tries to recover initial (quantized) coefficients. If the decoding is unreliable, the decoder asks for more parity bits and attempts the decoding again. The process continues until successful decoding. Afterward, a reconstruction block refines decoded information using the predicted information.

DSC has been explored extensively in the domain of video coding, notably with the development of DISCOVER [7] and VISNET II [8] codecs. Among the two, the latter leverages additional functionalities and shows consistent superiority compared to the former codec. These methods present similar or superior performance compared to H.264/AVC Intra, which has similar computational complexity, on scenes with low motion. On the other hand, they present lower performance than H.264/AVC No Motion, which has significantly higher complexity.

Besides the application in video coding, the distributed coding design is also suited for multi-view coding as it avoids relying on the communication between cameras. In the setups with a large number of cameras operating in power-constrained environments, DSC can effectively reduce the complexity of the encoder by eliminating the inter-camera dependency and frame buffering and shifting the prediction between neighboring views to the decoder side [35]. Based on these observations, DVC framework has been extended to Distributed Multi-view Video Coding (DMVC) [45]. With a similar aim, DSC has been applied to LFs in some preliminary works, e.g., [151, 1]. However, DSC of LFs has remained little explored till now.

6.3 . Related work

In this section, we mention the most relevant works to our approach. We consider approaches based DSC paradigm applied to video, multi-view data, and LFs. Furthermore, we briefly overview methodologies related to the SI generation block.

6.3.1 . DLFC approaches

[151] propose DSC for LF coding. They consider an array of low-cost cameras, whereas an individual coding of camera views is preferred compared to collecting and encoding all views jointly at a central computational node. Low-cost cameras with pixel-domain WZ encoder are interleaved with conventional cameras, which can provide good estimates of WZ views. WZ views are synthesized at a centralized

decoder using a geometry-based image rendering from the available key views. In the following work, [1] use transform-domain WZ coding to exploit better the spatial correlation and achieve higher RD performance.

Cong et al. [108] present DSC scheme for LF images. The scheme starts by downsampling LF views to QCIF resolution (176×144 pixels) and re-arranging views in a PVS. Then, an adaptive strategy skips the WZ decoding process if the synthesized view has a minimum quality to avoid transmitting bits for that particular view.

6.3.2 . DVC and DMVC approaches

Conversely, DVC and DMVC have received more attention. More precisely, novel approaches propose improving SI, as this plays a major role in the overall RD performance. The quality of generated SI can be improved by utilizing more adjacent frames [100] or multiple SI generation techniques [53, 54], which usually results in more than one estimate of SI. Maugey et al. [86] proposed three schemes to fuse the SI. Among the schemes, the fusion scheme utilizing the reciprocal of the residual and the reciprocal of vector magnitude as weights shows superior performance. Salmistraro et al. [117] propose a coding scheme for DMVC. They consider a horizontal, three-view scenario with video acquisition. The frames from lateral cameras are independently encoded using H.264/AVC codec, while the central-camera frames are processed either as key frames or WZ frames according to the GOPs structure. The scheme generates multiple predictions based on temporal and inter-view redundancies and employs a robust fusion method to merge likelihoods estimated from each SI.

6.3.3 . SI generation methods

As mentioned before SI generation block consists of two parts: prediction and residual modeling. Our prediction block utilizes a view synthesis technique which generates a view at a novel viewpoint from views given at different perspectives. Recently, view synthesis methods propose leveraging deep learning to generate high-quality views from a sparse set of input views. The overview of learning-based view synthesis methods has already been presented in Chapter 5.2.1, so we omit to repeat it here. Accurate Correlation Noise Modeling (CNM) is another essential aspect that influences the coding performance as it indicates the reliability of the prediction to an iterative decoder such as LDPCA. In DSC, the correlation noise is generally modeled by a Laplacian distribution. The authors in [17] explore the modeling of the correlation noise at different granularity levels and conclude that a higher granularity level translates to better RD performance, suggesting that the pixel-level and coefficient-level perform best in an offline mode for Pixel-Domain WZ and Transform-Domain WZ, respectively. In an online mode, the modeling is done adaptively based on the local intensity variation utilizing motion-compensated residuals at different granularity levels, e.g., frame-level, band-level, and coefficient-level. In [38], the estimated residual is divided into different classes

for each frequency band depending on the estimated residual energy for each block, and the Laplacian parameter is found using pre-calculated values in a lookup table. In [52], Previously Decoded Bands (PDBs) improve the noise model by classifying the subsequent residual into two categories. Additionally, a noise residue refinement step updates the noise residual after the decoding of each band. In [82], the residual frame is clustered into different classes using Fuzzy C Means based on the residual energy. Contrary to [52], it utilizes all the decoded frequency bands for improved noise modeling.

6.4 . Proposed method

This section describes the proposed DLFC scheme. We build upon previous work in DMVC [117] while modifying key functional blocks to achieve improved coding gains. More precisely, we propose to leverage a learning-based view synthesis approach, proposed by Navarro et al. [97], for the prediction of WZ views. Furthermore, we consider an improvement of the SI generation, whose quality directly correlates with the performance of the coding scheme. To this extent, we explore various modifications in the view synthesis scheme to obtain better predictions across different bitrates. Last but not least, we propose a deep learning scheme to estimate the uncertainty of our prediction.

In the following sections, we, first, give an overview of the DLFC scheme. Then, we describe a set of enhancements to view synthesis training for improved prediction. Next, we summarize the noise modeling in DLFC and propose a learning-based scheme to estimate it. We conclude the section with a description of the training procedure.

6.4.1 . Distributed LF compression

The proposed distributed LF coding scheme employs transform domain WZ coding with feedback channel [43].

The encoder is presented in Figure 6.2. It takes an LF image and extracts and divides views into two sets: key views and WZ views. We select four reference views of an LF image as key views according to one of the four arrangements shown in Figure 6.3 (b-e) and process them by a conventional coding tool, while the rest of the LF are processed using a computationally more efficient WZ encoder. First, each WZ view is transformed block-wise using the 4×4 DCT [112]. Then, the coefficients are quantized using one of eight proposed quantization matrices [7]. In the final step, the quantized coefficients are divided into bitplanes and independently encoded using a LDPCA encoder [140]. The computed syndrome bits of each bitplane are stored in the buffer together with 8-bit CRC.

At the decoder, key views are conventionally decoded and provided to the SI generation block. The role of the SI block is to estimate a WZ view, Y, as well as its corresponding residual signal, $\hat{R}$. The SI are then transformed using the 4×4 DCT, resulting in coefficients C_Y and $C_{\hat{R}}$ respectively. The noise mod-

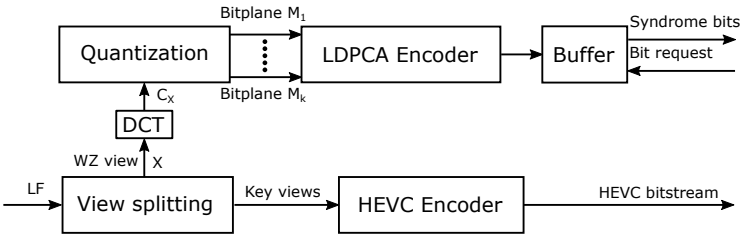


Figure 6.2: Block diagram of transform-domain WZ encoder.

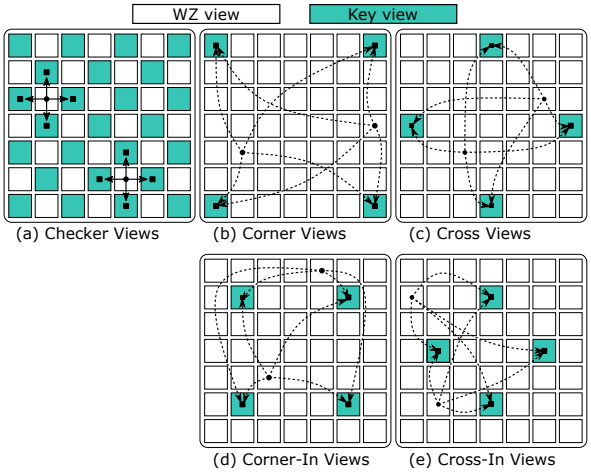


Figure 6.3: View splitting modes.

eling block considers Y as a noisy version of the original WZ view and utilizes residual coefficients $C_{\hat{R}}$ for CNM using the Laplacian distribution. The estimated distribution's parameters α_{CNM} and the prediction coefficients C_Y are provided to the soft input estimation block (and together with the information from the PDBs) used to calculate the bit-wise conditional probabilities for each bitplane (soft input). In order to decode bitplanes, the LDPCA decoder needs part of the accumulated syndrome bits from the encoder and the estimated soft input. Using the "message passing algorithm" [115] the decoder iteratively computes the source bits. The procedure stops upon convergence or a pre-defined number of iterations, and the decoder computes the syndrome bits from the estimated source bits. If the computed syndrome bits match the received syndrome stream, the Hamming distance of the two equal to zero, and pass the CRC checksum test, the decoding is considered successful. Otherwise, the decoder requests more bits from the encoder. After successfully decoding all bitplanes, the quantization intervals of a WZ view are obtained. Note, if the number of received syndrome bits is equal to the number of source bits, there is no compression gain, but successful decoding is guaranteed. In the final step, the WZ view is reconstructed using the maximum likelihood approach utilizing estimated Laplacian distribution and decoded quantization intervals [73]. The inverse DCT transforms the reconstructed view back to

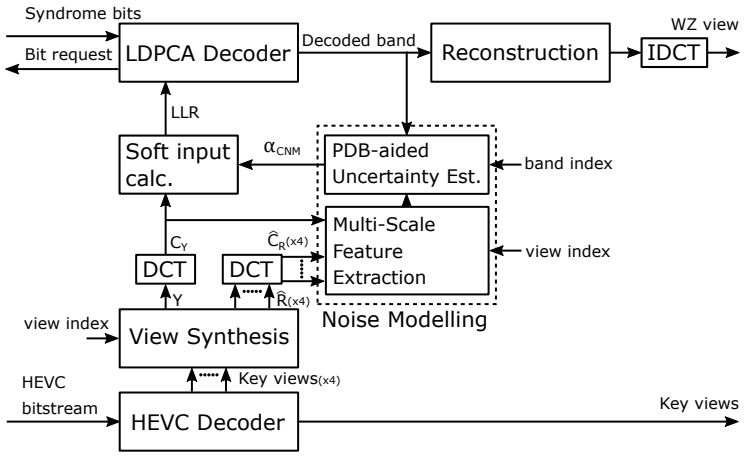


Figure 6.4: Block diagram of transform-domain WZ decoder.

the pixel domain.

6.4.2 . View synthesis

Baseline synthesis approach

For the sake of completeness, we briefly describe the view synthesis approach employed in the proposed DLFC scheme. We use the method [97] that has been already described in 5.3.1.

The view synthesis approach consists of three sequential networks: feature extraction, disparity estimation, and selection networks. The feature extraction network takes corner views (of an LF image) and the angular position of a novel view and extracts relevant information for the following stage. The disparity estimation network takes extracted features and the position of the novel view and estimates the disparity map of the novel view with respect to each corner view. Then, the corner views are warped following the estimated disparity maps and merged in the final estimation as a weighted sum with weights obtained by the selection network. The network is optimized using a two-part loss $\mathcal{L}_{l1-grad}$ which includes the L1 loss between the original image texture I and the synthesized image texture Y and the L1 loss of the gradients of the two textures:

$$\mathcal{L}_{l1-grad} = \|I - Y\|_1 + \frac{1}{2}\|\nabla I - \nabla Y\|_1. \tag{6.1}$$

The term in the loss function that compares gradients of the input view and the reconstructed one enforces similar gradients between two images allowing better preserving image textures.

Choice of reference views

In the coding of LF images using traditional coding tools, such as HEVC, much effort has been put into finding an optimal coding order, and it has been shown that the prediction from closer views provides better performance [141]. A typical configuration for view synthesis tasks includes a set of corner views in an LF image as they capture the widest field of view. We compare three more arrangements of reference views, as shown in Figure 6.3 (c-e), and select the one that provides the best prediction quality for SI generation.

Loss function

Furthermore, we evaluate two loss functions that could increase view synthesis's performance, especially with the decrease in the quality of reference views. More precisely, we consider a perceptual loss based on high-level feature maps of a deep neural network VGG utilized for the image classification task [128] and a loss which includes an uncertainty modeling of the prediction [69].

The early layers of the VGG network give a response highlighting low-level features of the input, while the deeper layers capture higher semantic information [41]. We assume that the inclusion of semantic reasoning will aid the view synthesis

network to generalize better in the case of the distorted input. We use pre-trained VGG-19, which is available in the PyTorch framework, and extract the activations from five layers as it is typically done in the literature [65][20] to compute the loss:

$$\mathcal{L}_{vgg} = \sum_{l}^{L} \|\Phi_l(I) - \Phi_l(Y)\|_1, \tag{6.2}$$

where Φ_l denotes the activations inferred from the layer l.

Kendall et al. [69] propose a loss function that considers the uncertainty in the prediction for the depth regression and semantic segmentation tasks. The loss function can be considered as learned attenuation as it penalizes the samples based on their prediction fidelity and provides a more robust estimation. Although our task does not explicitly regress depth, it depends on the estimated disparity maps at the intermediate levels. Moreover, our view synthesis task relies on the selection network to provide (soft) recommendations of the final prediction at the pixel level. Therefore, the robust estimation of the disparities should benefit the final prediction. We add a branch, which estimates uncertainty on a pixel level, to the original view synthesis network, and feed both estimates, the prediction, and the uncertainty, to a loss function defined as a negative logarithm of the likelihood of the Laplacian distribution. Note that it is also possible to select a Gaussian distribution. However, we choose the Laplacian as it is typically used to model the distribution of a residual signal. The Laplacian loss-based version of view synthesis approach is defined as follows:

$$\mathcal{L}_{laplacian} = -\frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{\alpha(n)}{2}\exp^{-\alpha(n)|I(n)-Y(n)|}\right), \tag{6.3}$$

where N is the total number of pixels in a batch, $\alpha(n)$ is the predicted Laplacian distribution parameter, and $I(n)$ and $Y(n)$ are ground truth and predicted pixel values, respectively.

6.4.3 . Correlation noise modeling

In an offline design process, the residual signal is used to model the correlation noise in the prediction of the WZ view. Typically, the Laplacian distribution offers a good fitting to the distribution of the correlation noise, where the distribution's parameter α_{CNM} should describe the reliability of the prediction. As the statistics of the correlation noise vary locally [17] estimating the distribution at the finer level is desirable. As reported in [17], the noise modeling at the finest level, i.e., pixel-level in the pixel-domain WZ or coefficient-level in the transform-domain WZ, offers optimal RD performance.

For example, the model parameter α_{CNM} of each coefficient (u, v) is defined inversely proportional to the absolute coefficients of residual signal $C_R(u, v)$ [17]:

$$\alpha_{CNM}(u, v) = \frac{\sqrt{2}}{|C_R(u, v)|}. \tag{6.4}$$

Due to the unavailability of the original WZ view at the decoder, the actual correlation noise cannot be used to model the distribution. Instead, the difference in the two predictions of the WZ view substitutes the actual residual signal, as the agreement in the two predictions represents the likelihood of the accuracy in the prediction. This approach can model well the correlation noise in prediction at the coarsest level. As we move towards the finer level, the noise modeling becomes unreliable due to an insufficient number of samples required for accurate modeling and the uncertainty in the residual estimation itself. Therefore, several methods have been proposed in the literature for robust correlation noise modeling, e.g., [17, 52].

For the noise modeling using the estimated residual signal, we used the approach described in [52]. The residual signal is estimated as a weighted average of the estimated intermediate residuals corresponding to the four corner views used at the input of the view synthesis method. The intermediate residuals $\hat{R}_i$ are calculated as follows:

$$\hat{R}_i(x,y) = Y(x,y) - W_i(x,y), \tag{6.5}$$

where Y is the predicted view, and W_i is the warped view corresponding to a view i from a set of reference views $\mathbf{I}$. The estimated individual residual noise signals are merged following:

$$\hat{R}(x,y) = \sum_{i \in \mathbf{I}} w_i^{\text{residual}}(x,y)\hat{R}_i(x,y), \tag{6.6}$$

where

$$\begin{aligned} w_i^{\text{residual}}(x,y) &= \frac{\log \prod_{j \in \mathbf{I} \setminus i} \left|\hat{R}_j(x,y)\right|}{\sum_{k \in \mathbf{I}} \log \prod_{j \in \mathbf{I} \setminus k} \left|\hat{R}_j(x,y)\right|} \\ &= \frac{\sum_{j \in \mathbf{I} \setminus i} \log \left|\hat{R}_j(x,y)\right|}{\sum_{k \in \mathbf{I}} \sum_{j \in \mathbf{I} \setminus k} \log \left|\hat{R}_j(x,y)\right|}. \end{aligned} \tag{6.7}$$

The level of uncertainty in the estimation process is represented with the degree of agreement of the warped key views, so that the SI with higher uncertainty should contribute less to the final residual. Thus, the reciprocal of noise value is better suited to model the contribution of the noise value. But, the sum of reciprocal value introduces a multiplication operation which becomes highly sensitive to changes in residual value. Therefore, we apply the natural logarithm function to achieve a more stable solution as proposed in Equation (6.7).

We have noted that the RD performance is still inferior compared to the case when the original residual is used for noise modeling in the offline process. We propose leveraging a learning-based approach to estimate the residual signal optimally using the predicted WZ view and the warped residuals. In [17], for the robust noise modeling, based on the local variation in the neighborhood, the variances

estimated from coarse-to-fine levels are assigned at the pixel level. The correlation between models across different bands is also exploited for improved modeling in [51]. We consider these approaches to design a network that can robustly estimate the residual signal.

Proposed network to model the residual signal

As our scheme is based on transform domain WZ, the residual is initially transformed to calculate α_{CNM}. The DCT transformation requires a signed residual as an input. As the absolute value of the transformed residual $|C_R|$ is utilized in Equation (6.4), we directly estimate $|C_R|$ using the network. In this way, we can calculate the absolute value of the transformed residual signal directly and simplify our prediction.

The proposed network estimates the absolute coefficients of the residual signal in two steps. These two parts are detailed in Tables 6.1 and 6.2, respectively. The first network extracts multi-scale spatial features from the synthesized view and the estimated residual signals. The statistics of the residual signal remain mostly constant across all the frequency bands. Utilizing them will help the network to generalize well across different datasets and frequency bands. Therefore, the first set of blocks of the network $\mathcal{F}_{INT}, \mathcal{F}_{MS}, \mathcal{F}_G$ are trained to learn common features across all the bands through weight sharing by utilizing 3D kernels with depth size of 1. It is also essential to consider the difference in the properties of the residual signals of different frequency bands. Therefore, we utilize another set of layers in the block $\mathcal{F}_S^b$ that is uniquely trained to process each frequency band b.

The block $\mathcal{F}_{INT}$ extracts some intermediate features F_{int} in the following way:

$$F_{int} = \mathcal{F}_{INT}\left(C_Y, C_{\hat{R}_1}, C_{\hat{R}_2}, C_{\hat{R}_3}, C_{\hat{R}_4}, P, Q\right), \tag{6.8}$$

where C_Y is the transformed coefficients of the predicted WZ view and $C_{\hat{R}_i}$ is the transform of the estimated residual corresponding to the cross-view i calculated using Equation (6.5). Additionally, the tensors P and Q consisting of the current view index p and q, respectively, are passed to this layer for the network to learn the view-position-dependent features. The stacking of signals results in a 3D input volume with seven channels. The output F_{int} is then passed to three parallel sets of convolutional layers, $\mathcal{F}_{MS}$, that learn to filter the intermediate features at multiple levels, i.e., with kernels of different receptive fields. These outputs are then concatenated and processed by $\mathcal{F}_G$. Finally, the features specific to each frequency band b are learned by $\mathcal{F}_S^b$:

$$F_s(b) = \mathcal{F}_S^b\left(\mathcal{F}_G\left(F_3, F_5, F_7\right)\right). \tag{6.9}$$

It should be noted that this network tries to learn the features without exploiting inter-band correlation. It is shown in [52, 82] that there exists some correlation in

Table 6.1: The network architecture of initial residual estimation. k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation.

	Name	k	In	Out	Depth	Act. f.
$\mathcal{F}_{INT}$	Input		7			
	conv0	$3 \times 3 \times 1$	7	16	16	ELU
	conv1	$3 \times 3 \times 1$	16	32	16	ELU
	conv2	$3 \times 3 \times 1$	32	32	16	ELU
	conv3	$3 \times 3 \times 1$	32	32	16	ELU
	conv4	$3 \times 3 \times 1$	32	32	16	ELU
	Output: F_{int}			32	16	
$\mathcal{F}_{MS}$	Input: F_{int}		32		16	
	conv0	$3 \times 3 \times 1$	32	32	16	ELU
	conv1	$3 \times 3 \times 1$	32	32	16	ELU
	conv2	$3 \times 3 \times 1$	32	16	16	ELU
	conv3	$3 \times 3 \times 1$	16	4	16	ELU
	Output: F_3			4	16	
	Input: F_{int}		32		16	
	conv0	$5 \times 5 \times 1$	32	32	16	ELU
	conv1	$5 \times 5 \times 1$	32	32	16	ELU
	conv2	$5 \times 5 \times 1$	32	16	16	ELU
	conv3	$5 \times 5 \times 1$	16	4	16	ELU
	Output: F_5			4	16	
	Input: F_{int}		32		16	
	conv0	$7 \times 7 \times 1$	32	32	16	ELU
	conv1	$7 \times 7 \times 1$	32	32	16	ELU
	conv2	$7 \times 7 \times 1$	32	16	16	ELU
	conv3	$7 \times 7 \times 1$	16	4	16	ELU
	Output: F_7			4	16	
$\mathcal{F}_G$	Input: Concatenate $[\mathcal{F}_3, \mathcal{F}_5, \mathcal{F}_7]$					
	conv0	$3 \times 3 \times 1$	12	32	16	ELU
	conv1	$3 \times 3 \times 1$	32	32	16	ELU
	conv2	$3 \times 3 \times 1$	32	32	16	ELU
	Output: F_g			32	16	
$\mathcal{F}_S^b$	Input: $F_g\,(b)$		32		-	
	conv0	3×3	32	32	-	ELU
	conv1	3×3	32	16	-	ELU
	conv2	3×3	16	4	-	ELU
	conv3	3×3	4	1	-	
	Output: $F_s\,(b)$			1	-	

the residual signals for different frequency bands. Hence, exploiting the correlation utilizing PDBs will improve the residual estimation process.

The second network is composed of two parts. The first part $\mathcal{D}$ processes the PDBs to exploit inter-band correlation. Instead of passing decoded bands to the network, the target residual C_R^q (the difference between the quantized coefficients of the WZ view and the coefficients of the prediction C_Y) of these bands are

computed and then provided to the block $\mathcal{D}$:

$$F_d(b) = \mathcal{D}^b\left(C_R^q \cdot M(b), b\right), \tag{6.10}$$

where $M(b)$ masks out the non-decoded bands in C_R^q. The features $F_d(b)$ and $F_s(b)$ are passed to the second part of this network $\mathcal{R}$ which makes the final prediction $\beta(b)$. The network is trained such that $\beta(b)$ represents the absolute coefficients of the residual which can be used to calculate $\alpha_{CNM}(b)$ for each band b in the following way:

$$\alpha_{CNM}(b) = \frac{\sqrt{2}}{\beta(b)} \tag{6.11}$$

The LDPCA decoder can only decode the coefficient of a WZ view up to some quantization level; therefore, it is intuitive to train a network for the quantized target residual C_R^q. In addition, the estimated residual plays a vital role in the reconstruction part as it is used along with the synthesized view and the decoded bands to find a maximum likelihood solution. We have observed that in this case, the actual residual signal C_R, i.e., the difference between unquantized coefficients of the WZ view and the coefficient of the prediction C_Y, results in the optimal reconstruction performance. Hence, two networks are trained for each residual signal.

The second network in the proposed scheme utilizes the quantized decoded bands. The statistics of decoded bands vary from one quantization index to another. To achieve the best coding performance, the networks are trained for each quantization index M independently. Each layer in the residual estimation network is followed by batch normalization.

6.5 . Experiments

In this section, we, first, define training and testing conditions. Then, we introduce ablation studies and present the obtained results. Finally, we describe anchors against which the best variant of the proposed methods is compared to.

6.5.1 . Datasets

Training. We use the Flowers dataset [130] which consists of 3343 images of plants. We select one hundred images for validation and the rest of the dataset for training.

To train view synthesis networks, at each training iteration, we randomly select the position of a target view, randomly crop training samples comprised of a set of reference views and the target view to the spatial size 192×192, and augment processed samples by applying gamma correction with the gamma value randomly selected from the range $[0.4, 1.0]$. We exclude the positions of the reference views when selecting the target view. We observe the convergence of the model on the

Table 6.2: The network architecture of refined residual estimation (aided by decoded bands). k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation. In this network each layer is followed by batch normalization.

	Name	k	In	Out	Depth	Act. f.
$\mathcal{D}^b$	Input		17			
	conv0	3×3	17	32	-	ELU
	conv1	3×3	32	64	-	ELU
	conv2	3×3	64	64	-	ELU
	conv3	3×3	64	32	-	ELU
	conv4	3×3	32	32	-	ELU
	conv5	3×3	32	1	-	ELU
	Output: $F_d(b)$			1	-	
$\mathcal{R}^b$	Input: Concatenate $[F_s(b), F_d(b)]$					
	conv0	3×3	2	32	-	ELU
	conv1	3×3	32	32	-	ELU
	conv2	3×3	32	16	-	ELU
	conv3	3×3	16	4	-	ELU
	conv4	3×3	4	1	-	
	Output: $\beta(b)$			1	-	

validation set wherein we use the full spatial size, select center views only, and randomly select the gamma value from the range $[0.4, 0.5]$. We use the ADAM optimizer with default parameters and set the batch size to 10.

In order to train the network for residual estimation, we need to provide the data in the transformed domain. A trained model for the view synthesis network predicts the WZ view of spatial size 192×192, which, after transformation, results in 48×48 spatial resolution. Therefore, the residual estimation network is trained with batches with a 48×48 block size for all the inputs. Considering the nature of the residual signal, we have used the Laplacian distribution as the loss function to train the residual estimation networks for coding and reconstruction using $\mathcal{L}_C$ and $\mathcal{L}_R$, respectively. The two loss functions are defined as follows

$$\mathcal{L}_C = \sum_b \log \beta_C(b) + \frac{|C_R(b)|}{\beta_C(b)}, \tag{6.12}$$

$$\mathcal{L}_R = \sum_b \log \beta_R(b) + \frac{\left|C_R^q(b)\right|}{\beta_R(b)}, \tag{6.13}$$

where β is the variance of the Laplacian distribution estimated at the coefficient level. The loss functions $\mathcal{L}_C$ and $\mathcal{L}_C$ reach their optimal minima when $\beta_C = \left|C_R^q\right|$ and $\beta_R = |C_R|$, respectively.

The networks are implemented and trained in Python using the PyTorch framework. Each view synthesis network is trained for 300 epochs, which takes around 15 hours on GeForce RTX 2080 Ti GPU. At the same time, the training of each

residual estimation network runs for 750 epochs and takes around 37 hours on Tesla V100 GPU.

Evaluation To analyze the RD performance, we utilize the EPFL dataset [110]. We select 8 LF images, illustrated in Figure 6.5, and decode them using LPT. Firstly, each view is zero-padded to the effective resolution of 376×544. This step is necessary because the spatial resolution needs to be a multiple of four as WZ encoding uses 4×4 DCT. The original resolution is 376×541 pixels. After transformation, each frequency band has an effective resolution of 94×136 pixels. Since the bitplanes for each frequency band are encoded one at a time by the LDPCA encoder, this results in a source code of length 12784 bits. We design LDPCA codes for this length following the procedure described in [140]. Only the luminance channel is used to report the performance.

The four key views are decoded using HEVC Intra decoder (HM reference software, v.16.22, with Range Extension (REext) mode and Main profile). The RD performance of distributed coding schemes is evaluated at four different RD profiles by selecting quantization matrices from [7] at quantization indices $M = [1, 4, 7, 8]$. To have the same quality key views and WZ views after the reconstruction, the QP parameter in HEVC is selected to match the quality of the reconstructed WZ view for each LF and quantization index, as specified in Table 6.3.

Table 6.3: Quantization parameters of the key views corresponding to four quantization indices $M = [1, 4, 7, 8]$ from the set in [7] to have consistent quality of reconstructed views.

Sequence	Q_1	Q_4	Q_7	Q_8
Bikes	41	29	25	22
Danger	41	30	25	22
Desktop	42	29	25	22
Flowers	40	30	25	22
Fountain	42	32	27	23
Friends	40	27	23	20
Pillars	38	28	23	21
Vespa	41	28	24	21

6.5.2 . Ablation studies

In this section, we analyze the performance of the view synthesis approach based on the variations proposed in Section 6.4.2 and select the approach that

(a) *Desktop*. (b) *Flowers*. (c) *Friends*. (d) *Vespa*.

Figure 6.5: Thumbnails of central views of added test LFs from the EPFL dataset [110].

generally performs best in terms of objective quality for the SI generation in the proposed DLFC scheme. We consider four arrangements of the reference views and evaluate the performance on variants of two datasets: California [66], and EPFL [110]. *EPFL-LPT* denotes a variant of the EPFL dataset obtained using LPT for decoding while the *EPFL-DAN* variant employs the decoding proposed by Dansereau et al. [27]. We use only one variant of the California dataset, denoted simply *California*, that uses LPT for the decoding. Employing different datasets and mentioned variants evaluates performance on different datasets compared to the training dataset and allows the evaluation of the impact of the decoding procedure. Next, we observe the behavior of the view synthesis methods trained on three different loss functions. In addition, the view synthesis network is trained on distorted reference views by minimizing the same set of loss functions. Last but not least, three variants of the residual signal estimator are compared.

View arrangements. In the first experiment, we compare the performance concerning the arrangements of the four reference views. For each of the four arrangements shown in Figure 6.3 (b-e), the view synthesis network is independently trained. Table 6.4 provides the quantitative analysis of the performance of the

Table 6.4: Performance evaluation of four arrangements for view synthesis task across three datasets in terms of PSNR (dB).

Dataset	*Corner*	*Corner-In*	*Cross*	*Cross-In*
California	38.20	38.64	39.07	38.90
EPFL-LPT	39.50	40.62	40.98	40.77
EPFL-DAN	30.65	32.48	32.17	32.65

view synthesis network for each arrangement of the reference views utilizing the three datasets described earlier. Overall, the *Cross* arrangements performed better across all the datasets. Moreover, since the view synthesis network is trained on LPT datasets, the *Cross* arrangement performs better on the *EPFL-LPT* and *California* datasets. Based on the superiority of *Cross-In* arrangement on the *EPFL-DAN* dataset, we deduce that this arrangement generalizes the LF structure better. Generally, it can be observed for the datasets decoded using LPT that significantly higher quality is achieved across different reference view arrangements than the dataset decoded using Dansereau's toolbox, i.e., *EPFL-DAN*. This comparison suggests that the trained models generalize well across different datasets but not across different LF decoding schemes. For the rest of the evaluation, we consider the *Cross* arrangement as the default arrangement for the proposed approach due to its superiority to the LPT decoded datasets. From Table 6.4, we observe that even though the inward variant of corner arrangement, *Corner-In*, improves performance compared to the original variant, this trend does not repeat in the case of *Cross* view arrangements. We explain this behavior by considering the similarity between reference views and the rest of the LF. Namely, by reducing the distance between

the reference views, the prediction quality of the in-between views should increase as the reference views are more similar in this case. Conversely, the quality of the extrapolated views degrades with an increase in their distances from the reference views. Therefore, it would be beneficial to find an optimal set of reference views for which the quality of synthesized in-between views increases while the quality of extrapolated views does not degrade considerably. Based on the results presented in Table 6.4 it can be noted that a "sweet spot" lies around *Cross* reference arrangement for datasets decoded using LPT and *Cross-In* reference arrangement for *EPFL-DAN* dataset.

Loss functions. Next, besides the originally proposed loss function, we explore two more loss functions as proposed in Section 6.4.2. From Table 6.5, it can be observed that $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ versions underperform compare to the original version $\mathcal{L}_{l1-grad}$ on LPT decoded datasets. On the other hand, the evaluation of the *EPFL-DAN* dataset suggests that some loss functions generalize better than others across different decoding schemes, e.g., $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$. This result motivates us to further explore these variants for the distorted inputs, which will be provided to the view synthesis network at the decoder of the proposed DLFC scheme.

Table 6.5: Performance evaluation of three loss functions for view synthesis task on Cross arrangement across three datasets in terms of PSNR (dB).

Dataset	$\mathcal{L}_{l1-grad}$	$\mathcal{L}_{vgg}$	$\mathcal{L}_{laplacian}$
California	39.07	38.46	38.12
EPFL-LPT	39.98	39.44	38.96
EPFL-DAN	32.17	32.43	32.49

Table 6.6 provides a quantitative evaluation in the case of distorted input views. We compare three loss functions in the *Cross* arrangement, selected due to the superior performance compared to the other arrangements 6.4, and on *EPFL-LPT* data as the best performance was achieved 6.5. Comparing Tables 6.5 and 6.6, we observe in the case of undistorted inputs that the original loss function $\mathcal{L}_{l1-grad}$ performs better compared to both $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ losses. In the case of distorted input views, we note the same behavior with a small exception in the case of the loss $\mathcal{L}_{laplacian}$ which seems to degrade relative quality between different quality levels less compared to the two other loss functions.

Although we can observe better generalization of $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ version across different datasets, these trends do not repeat on the distorted datasets. Therefore, we adopt the version of the network trained using the original loss function $\mathcal{L}_{l1-grad}$ in subsequent experiments.

Residual signal estimator. Next, we study the effect of utilizing different methods for estimating residual signals in the overall RD performance. The first variation utilizes weights calculated based on the four independent residual esti-

Table 6.6: Quantitative evaluation of view synthesis approach given distorted *Cross* arrangement reference views from the *EPFL-LPT* dataset in terms of PSNR (dB).

QP	$\mathcal{L}_{l1-grad}$	$\mathcal{L}_{vgg}$	$\mathcal{L}_{laplacian}$
27	37.96	37.71	36.95
32	35.78	35.56	35.12
38	32.61	32.41	32.30
45	28.78	28.63	28.74

mates obtained from each reference view to estimate the final residual signal. We denote this approach as *Cross-Weighted* when used alongside the *Cross* arrangement of the reference views. As another variation, we introduce *Cross-Ideal*, which utilizes the ideal residual signal to set the upper bound of the achievable performance. Figure 6.6 acknowledges the improvement achieved using the network-based approach *Cross-Net* to estimate the residual signal over *Cross-Weighted*. However, it can be inferred by looking at *Cross-Ideal* curves that even with accurate residual estimation, the performance can not surpass the upper bound. Given this situation, we can say that considerable improvement is achieved over *Cross-Weighted* using *Cross-Net*.

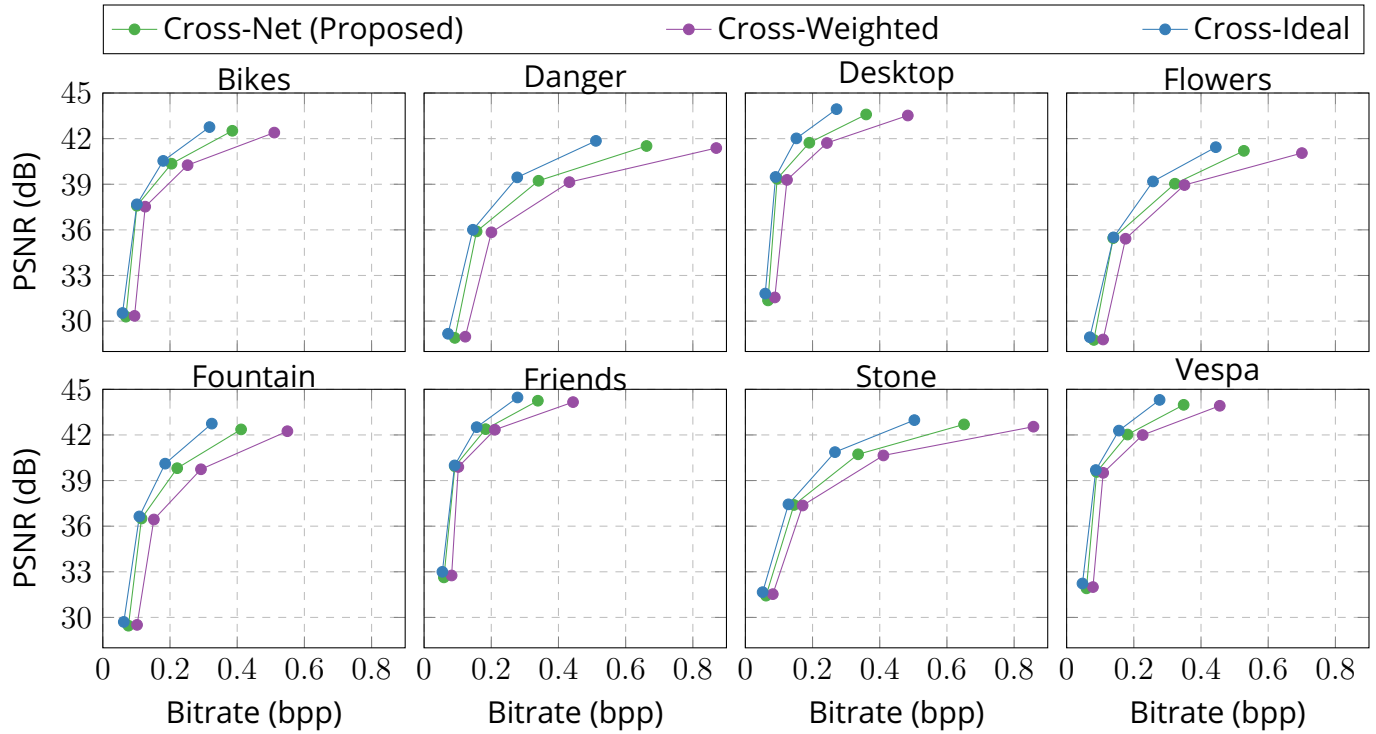


Figure 6.6: RD performance comparison between different variations of the proposed DLFC scheme utilizing three different residual estimation methods, at quantization indices $M = [1, 4, 7, 8]$, using PSNR as distortion metric.

6.6 . Results

In this section, we present the RD performance of the proposed scheme compared to relevant state-of-the-art schemes. We consider two DLFC schemes and three standard coding schemes. Besides RD performance, we visualize outcomes at different stages of the decoding procedure.

6.6.1 . RD performance

First, we compare a variant of the proposed scheme . The scheme employs the view synthesis method presented in Section 6.4.2 and uses the *Corner* arrangement of the reference views for the estimation of WZ views. Differently compared to the proposed approach, the residual signal is estimated as a weighted average of intermediate residuals following Equations 6.5, 6.6 and 6.7. This method will be denoted as DLFC-I in the following. The second DLFC scheme we compare the coding performance of the proposed scheme to is the method of Salmistraro et al. [117] (denoted later as Checker-MultiSI), which presents the state-of-the-art DMVC approach adapted for LF scenario. Here, the views are split in a checkerboard pattern, as shown in Figure 6.3-a, to utilize horizontal and vertical adjacent neighbors of a WZ view for its prediction. Contrary to DMVC, an additional angular dimension substitutes the temporal dimension. Compared with conventional coding schemes, we select HEVC-Intra as the first anchor to compress all the views independently. The same HEVC configuration is utilized for the key-views coding. Inspired by the comparison provided in [18], we compare our approach with HEVC-NoMotion, which is superior to the former approach because it exploits temporal redundancy like HEVC-Inter, but the motion search range is set to zero. The configuration provided in [18] has been used to configure the HEVC encoder for HEVC-NoMotion. The encoder is provided with the 1-D sequence of LF views as a pseudo video sequence, generated by following a serpentine scanning order. A relevant anchor to compare is the standard LF coding scheme provided by JPEG Pleno [122]. We compare only to MuLE, i.e., the transform-based mode of the reference software, as it has been shown that it is superior compared to the prediction-based mode of the JPEG Pleno reference software on lenslet data [106]. Section 3.4 briefly describes the MuLE codec.

Figure 6.7 plots the RD performance of the above-described schemes and the proposed method, utilizing PSNR as a distortion metric. It can be observed that all the variations of distributed coding significantly outperform HEVC-Intra due to the high quality of the synthesized views. The conclusion is evident by observing that the difference in performance reduces as the distortion increases. With a higher distortion, the compression artifacts become significant in the key views, due to which view synthesis can no longer exploit the common feature points in all the key views.

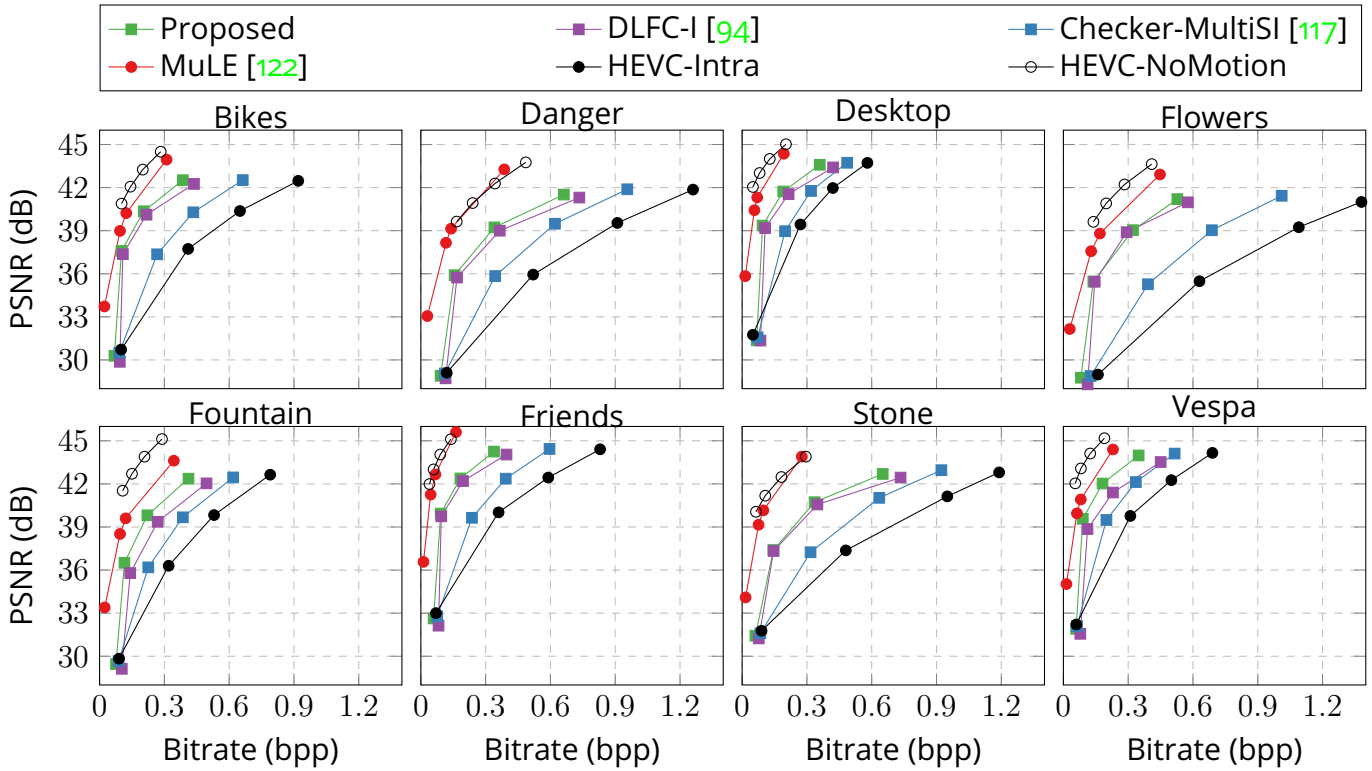


Figure 6.7: RD performance comparison of distributed and conventional coding schemes using PSNR as distortion metric at quantization indices $M = [1, 4, 7, 8]$, whereas, the quantization parameters specified in Table 6.3 are used for both HEVC plots.

Table 6.7 and Table 6.8 quantify the performance of the coding schemes in comparison to HEVC-Intra in terms of Bjøntegaard Delta [15]. It can be observed that the proposed scheme outperforms both distributed coding architectures. The higher RD performance of the proposed approach compared to the DLFC-I can be attributed to the quality gains in the view synthesis approach and the improvement in residual signal estimation using the network-based scheme. In the case of Checker-MultiSI, we would expect higher performance due to the availability of closer reference views for view synthesis. However, it requires half of the views to be encoded using HEVC-Intra, thus reducing the overall RD performance. Quantitatively, our approach achieves 0.96 dB and 4.02 dB gains in BD-PSNR and 17.45% and 46.66% reduction in BD-Rate compared to DLFC-I and Checker-MultiSI on average, respectively.

Table 6.7: Average coding performance in terms of BD-PSNR [dB] compared to HEVC-Intra.

	Proposed	DLFC-I [94]	Checker-MultiSI [117]	HEVC-NoMotion	MuLE [122]
Bikes	6.27	5.67	1.67	9.46	8.30
Danger	5.43	4.64	1.86	8.78	9.48
Desktop	3.78	3.04	0.58	8.81	7.83
Flowers	7.21	7.24	2.35	10.98	9.69
Fountain	5.40	3.52	1.59	10.33	8.06
Friends	5.34	4.75	1.19	9.91	9.52
Pillars	4.46	4.08	1.45	8.89	8.66
Vespa	4.56	2.85	1.42	9.01	7.03
Average	5.31	4.47	1.51	9.52	8.57

Table 6.8: Average coding performance in terms of BD-Rate [$\%$] compared to HEVC-Intra.

	Proposed	DLFC-I [94]	Checker-MultiSI [117]	HEVC-NoMotion	MuLE [122]
Bikes	-64.8	-57.8	-25.7	-84.2	-82.9
Danger	-59.4	-53.1	-27.2	-78.9	-85.3
Desktop	-43.9	-32.4	-7.3	-84.8	-83.4
Flowers	-71.0	-67.7	-32.3	-86.0	-86.2
Fountain	-54.6	-39.1	-22.2	-82.8	-79.8
Friends	-62.2	-54.6	-21.4	-90.5	-90.6
Pillars	-61.4	-58.3	-26.9	-88.1	-89.2
Vespa	-55.2	-37.4	-23.9	-85.8	-81.0
Average	-59.1	-50.1	-23.4	-85.1	-84.8

Overall, it can be observed that the distributed coding schemes achieve roughly $50\% - 65\%$ improvement in BD-Rate (Table 6.8) and 4.5 dB - 6.2 dB gains in

BD-PSNR (Table 6.7). Compared to HEVC-NoMotion and MuLE, we can observe the clear downside of using distributed coding schemes. Quantitatively, HEVC-NoMotion and MuLE achieve 4.18 dB and 3.52 dB gain in BD-PSNR (Table 6.7, and 66.09% and 57.34% (Table 6.8 reduction in BD-Rate, respectively, in comparison to our approach. On the other hand, these schemes involve computationally extensive operations at the encoder side suited for broadcasting applications.

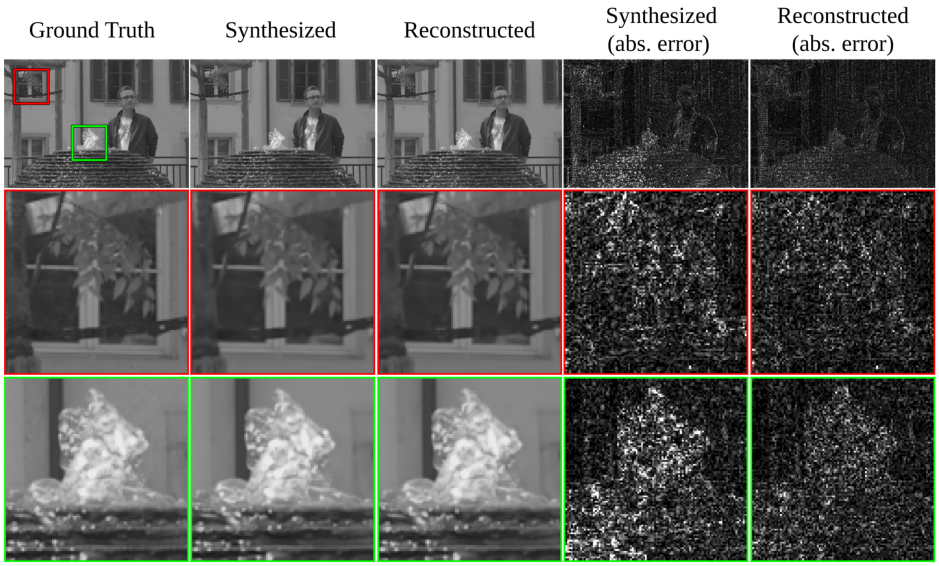
Although the compression performance of the distributed coding paradigm lacks the best conventional coding schemes, we emphasize that the application areas and goals are different, and we focus on the encoding complexity. Therefore, we discuss the performance of the proposed scheme in comparison to the conventional coding schemes in terms of encoding time. HEVC-Intra does have a complex encoding scheme, even though it does not exploit inter-view redundancy between the views. On the other hand, the other two schemes, HEVC-NoMotion and MuLE, require inter-view communication to exploit the redundancy at the encoder, resulting in increased complexity of the encoding architecture and additional overhead in the encoding time. For example, our measurements show that, on average, encoding the LF with the proposed method is 8 to 10 times faster than HEVC-Intra, depending on the quantization index, whereas it is 12 to 18 times faster compared to HEVC-NoMotion. In comparison to MuLE, our scheme is 20 to 30 times faster.

It is well-known that distributed coding schemes offer high-efficiency encoding by compromising on the simplicity of the decoder [43]. The major contributor to the decoding complexity in our implementation is the iterative LDPCA decoder. Although the iterative LDPCA decoder provides near-optimal performance, due to its iterative nature, it requires further work on speeding up the iterative decoding for real-time decoding applications. For instance, in the proposed scheme, decoding of a WZ view can be 300 to 1300 times slower than encoding it, depending on the quantization index. Neglecting that the implemented decoding solution is not optimized compared to the HEVC decoder, we note that the implementation of the proposed decoding scheme can be approximately three orders of magnitude slower.

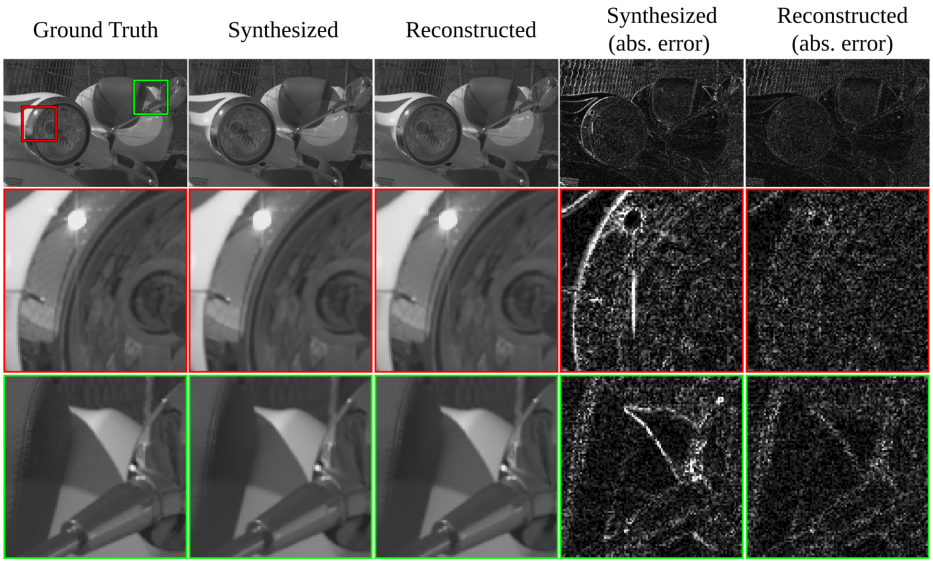
6.6.2 . Visual analysis

Figure 6.8 illustrates the outputs of the stages in the proposed decoding scheme. In the second column, we can note that the synthesized view provides accurate information about the WZ view in most of the regions. Still, higher errors can be observed in challenging areas such as non-Lambertian surfaces and occluded regions. At the same time, the errors in these areas in the reconstructed views are corrected as observed by the limited error magnitude, which is an outcome of utilizing successfully decoded WZ views for the final reconstruction.

6.7 . Conclusion



(a) Fountain



(b) Vespa

Figure 6.8: Visual comparison between the outputs of stages in the proposed decoding scheme to decode the central view of the two LF sequences i.e. *Fountain* and *Vespa* at quantization index $M = 8$. The ground truth image and corresponding zoomed patches are shown on the left. The synthesized and the reconstructed WZ view along with the corresponding absolute errors (range normalized to $0.00 - 0.04$) are shown in the next four columns. The zoomed patches are extracted from the highlighted regions in the ground truth images.

In this chapter, we have presented a novel approach for distributed LF compression. We proposed to improve SI generation block, i.e., both the prediction part and residual estimation part. The prediction part employs view synthesis, which complements the distributed coding paradigm as it enables generating high-quality novel views from a sparse set of key views. Using view synthesis effectively reduces

the number of key views that need to be coded and transmitted. We compare different arrangements of reference views and show that *Cross* arrangement offers more accurate prediction, which finally leads to an improvement of the overall RD performance. We also propose a deep learning-based architecture for the modeling of the residual signal that leverages PDB and employs common and specialized filters.

Our initial study shows that incorporating a deep learning-based view synthesis method into a distributed coding scheme improves coding performance compared to the HEVC Intra and state-of-the-art DLFC methods. We achieve further gains by integrating the proposed residual estimator network.

The presented results are associated with the following publication:

- M. U. Mukati, M. Stepanov, G. Valenzise, F. Dufaux and S. Førchhammer, "View Synthesis-based Distributed Light Field Compression," 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2020, pp. 1-6, doi:10.1109/ICMEW46912.2020.9105980.

- M. U. Mukati, M. Stepanov, G. Valenzise, S. Førchhammer and F. Dufaux, "Improved Deep Distributed Light Field Coding," *in IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 325-337, 2021, doi: 10.1109/OJCAS.2021.3073252.

We conclude the chapter by explicitly answering initially posed questions in the Introduction:

- How can we employ DSC paradigm for LF data? *DSC has been already used for single and multi-view video sequences, and these approaches, especially multi-view, already provide a good starting point for the application of DSC on LFs. The most obvious approach, and the one presented in this chapter, is to divide LF views in WZ and key views and to use depth-based or image-based rendering methods to predict the WZ views.*

- Can we leverage deep learning methodologies to improve coding performance of DSC pipeline? *We noted three functional blocks that can benefit from deep learning methods: the prediction of WZ views and the estimation of the residual information for the decoding and the reconstruction. Our experiments show that deep learning-based methodologies can provide highly accurate estimates at the outputs of these blocks and provide improvements in coding performance.*

- How well view synthesis performs as the prediction block in DSC? *Our experiments have shown that learning-based view synthesis brings significant benefits to the overall performance of a DSC scheme. It radically reduces the number of views that need to be coded as key views, thanks to high-quality predictions of the remaining views. Furthermore, it is highly flexible*

as it allows to select different reference views arrangements. For example, in our experiments, we have compared four arrangements and noted that the typically employed Corner arrangement does not always provide the best performance.

- How well can the residual signal be modeled using the deep learning methodology? *Our experiments have shown that by leveraging deep learning methodology for the prediction of the residual signal, it is possible to improve the overall coding gains compared to a hand-designed approach. Still, it might be possible to achieve further gains considering the lower coding performance than the upper bound set by an oracle estimate of the residual signal.*

- How DLFC enhanced with deep learning-based functional blocks compares to state-of-the-art methods? *The deep learning-enhanced DLFC scheme is superior to other state-of-the-art DLFC schemes and HEVC using Intra coding. On the other hand, the proposed scheme is still inferior to standard coding schemes which exploit inter-view correlation at the encoder side, such as HEVC or JPEG Pleno coding schemes. However, the complexity of these schemes is significantly higher at the encoder side, which, if deemed important, can be a turning point in the applicability of the distributed schemes.*

7 - Conclusions

7.1 . Overview and outcome

The thesis aims to improve coding performance using deep learning methods for LF contents. As a result, we have investigated, proposed, and evaluated novel coding solutions for unfocused plenoptic contents. In this section, we overview the work proposed in this thesis, analyze the objectives defined in the introductory chapter and conclude with limitations and following potential steps.

In Section 1.2 we have defined a set of particular objectives this thesis is looking to fulfill. The following overviews and discusses the outcomes of the proposed works and answers to the original objectives.

1. **Lossy LF coding.** *Explore the utility of end-to-end trained data-driven approaches for LF coding.*

   An end-to-end learning-based coding is an extremely appealing approach as it consists of a coding architecture with all its block optimized jointly. This approach presents strong leverage compared to traditional, hand-designed approaches. Chapter 4 introduces an end-to-end coding approach that takes a restructured LF image as an input, provides a compact representation of the input, and allows reconstructing the input from the representation. The restructured input and the codec architecture are designed to conform to each other. The codec employs computationally cheaper blocks, and it is optimized to reduce the rate-distortion cost. The codec is applied to plenoptic contents and shows superior performance to state-of-the-art coding anchors.

2. **Lossless LF coding.** *Investigate a design of the coding methodology and learning-based alternatives for conventional coding blocks.*

   A typical lossless coding scheme employs a predictive coding paradigm with a predictor and an entropy model. The goal of the predictor is to estimate samples of a single in the most accurate manner. In contrast, the entropy model estimates the probability distribution of the residual signal (the difference between original and predicted samples) that is needed for entropy coding. Considering this design, we considered the most promising learning-based methods for prediction and entropy modeling blocks. Chapter 5 presents a view synthesis methodology and its various operational configurations and an autoregressive model for entropy modeling and its improved variant that can compensate for prediction errors. We observed that view synthesis methods are extremely promising for prediction as they allow for

the transmission of a small number of views and reconstruction of the remaining views at high quality. Moreover, they are suited for hierarchical design and achieve, in this scenario, even improved performance. As for the entropy modeling, we considered recent state-of-the-art generative models as they can be employed readily to estimate the distribution of the following samples. When applied to plenoptic content, the proposed codec shows superior performance at reasonable costs to state-of-the-art coding methods.

3. **Distributed LF coding.** *Investigate learning-based alternatives to standard processing blocks in distributed schemes.*

   Distributed coding schemes promise to shift heavy processing from the encoder to the decoder side, as it is typically done in conventional approaches. There are two critical blocks for coding performance at the decoder side: prediction and correlation noise modeling of Wyner-Ziv views. Chapter 6 proposes a distributed LF coding scheme with the main blocks designed to leverage learning-based methods. We consider a learning-based view synthesis method for the prediction block and particular neural network architecture for the second block. Our analysis shows that both blocks demonstrated superiority compared to conventional methods and provided, overall, state-of-the-art coding performance.

7.2 . Limitations and future prospect

The work presented in this thesis can be improved as there are still challenges associated with the proposed coding schemes. In the following, we discuss possible directions how these challenges could be approached to:

- The coding solutions presented in Chapter 4 consider computationally less demanding convolutional layer instead of the 4D filters, which would naturally match the dimensionality of LF images. Moreover, more powerful entropy models could be employed to model better the representation to be encoded.

  It would also be interesting to explore the extension of the proposed scheme, which is optimized in an end-to-end fashion in its entirety. This direction would demand improving the base model so that it can operate efficiently across the entire bitrate range. As an initial step, the network's capacity can be increased to evaluate the saturation levels at high rates. Another direction could be to include an enhancement layer designed in the form of a neural network to encode the residual signal. The enhancement layer might have a simpler form designated for modeling MIs or EPIs, which might be an easier task than the modeling of SAIs.

  In addition, the proposed method was evaluated only on LFs captured with a plenoptic camera, while other types of LFs could be included as well. This

direction should be considered with some reservation considering the performance of similar approaches. Namely, the JPEG Pleno codec transform mode exhibits a significant drop in the coding performance as it cannot exploit the correlation between views in wide-baseline LFs. Also, the proposed method should be extended to support the coding of chroma components as it would allow comparison with a broader range of coding tools.

- The scheme proposed in Chapter 5 consists of two blocks that were trained sequentially. It would be interesting and potentially beneficial to train the entire scheme end-to-end.

  Another aspect that might be interesting to explore is different grouping configurations in the entropy model and their impact on performance and complexity.

  Regarding the limitations of evaluation conditions, it is vital to further evaluate the proposed method on the LFs with higher angular resolution. In the current experiments, traditional image-based codecs perform surprisingly well on the contents with the 7×7 angular resolution, suggesting a stronger correlation between MIs. At the same time, this is a very interesting observation as it demonstrates the high efficiency of traditional image codecs on non-natural still images. Furthermore, the coding method could be extended to support wide-baseline LFs.

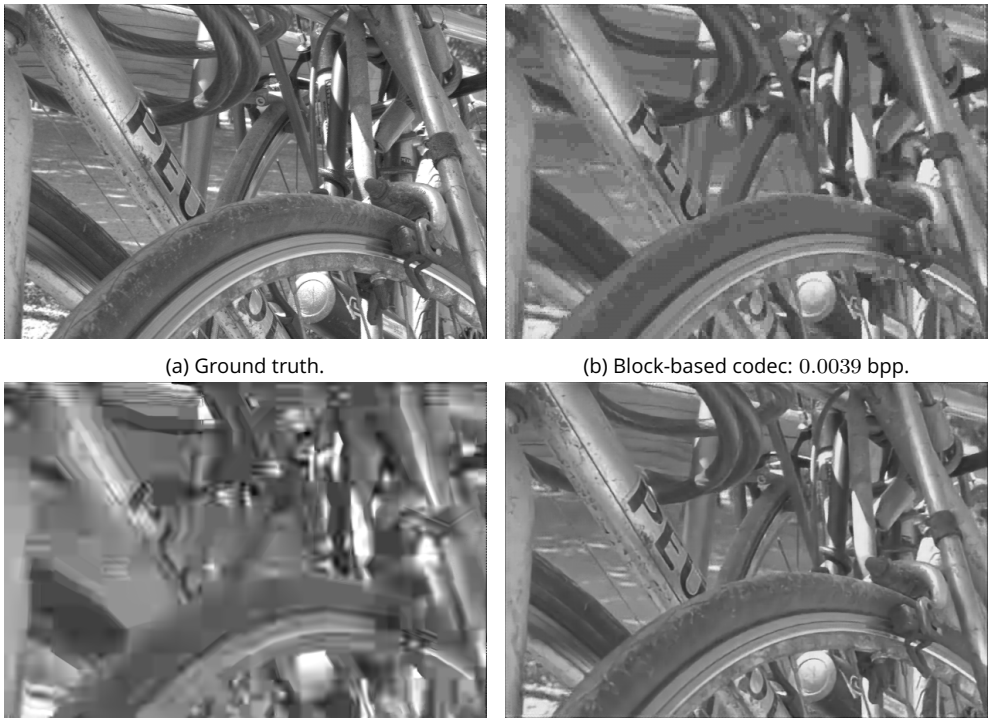
- The scheme proposed in Chapter 6 utilizes a feedback channel that signals the encoder to send more syndrome bits. This communication introduces great latency as costly decoding is executed in each iteration. Therefore, a possible improvement of the scheme could involve the elimination of the feedback channel, which requires an accurate estimation of the required number of syndrome bits at the encoder.

  A more critical limitation in the proposed work lies in its evaluation. Namely, the study considers LFs captured with a plenoptic camera, a significantly simplified scenario of a possible one. Therefore, future research should extend to LFs with a wide baseline and irregular sampling grids.

The main objective of the thesis was an investigation of deep-learning coding tools and deep-learning alternatives to critical functional parts of coding schemes for LF images. Further research is needed to include wide-baseline LFs and dynamic contents.

A - Learning-based Lossy Light Field Compression

A.1 . Visual evaluation



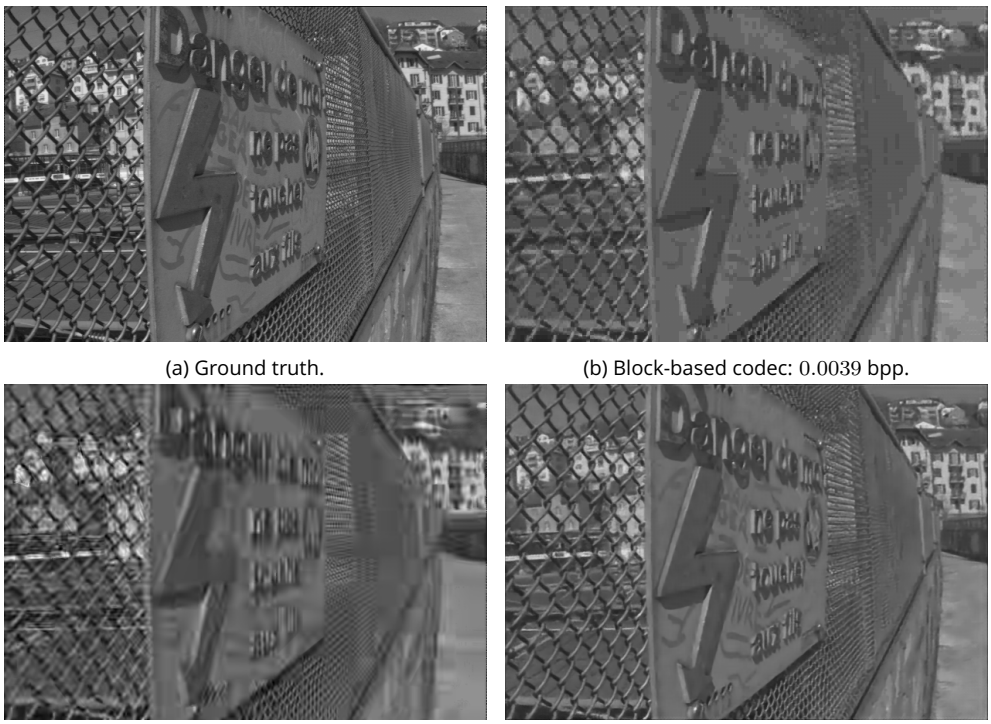
(a) Ground truth.

(b) Block-based codec: 0.0039 bpp.

(c) HEVC (x265): 0.003 bpp.

(d) Scalable codec: 0.0032 bpp.

Figure A.1: Visual evaluation of the central view of *Bikes* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d).



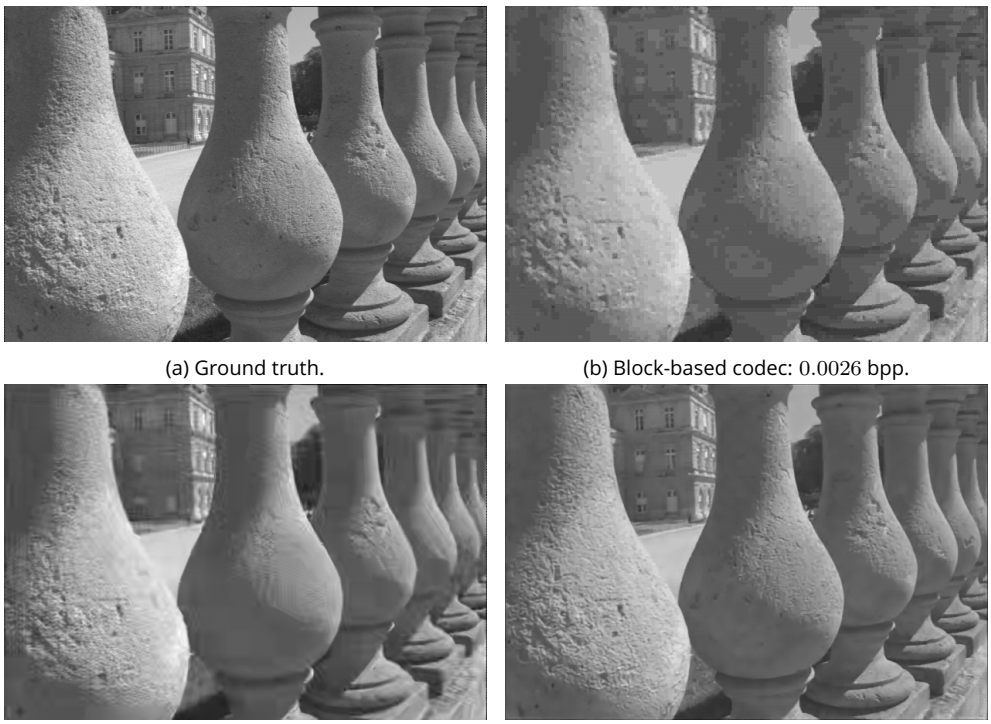
(a) Ground truth.

(b) Block-based codec: 0.0039 bpp.

(c) HEVC (x265): 0.0045 bpp.

(d) Scalable codec: 0.0035 bpp.

Figure A.2: Visual evaluation of the central view of *Danger* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d).



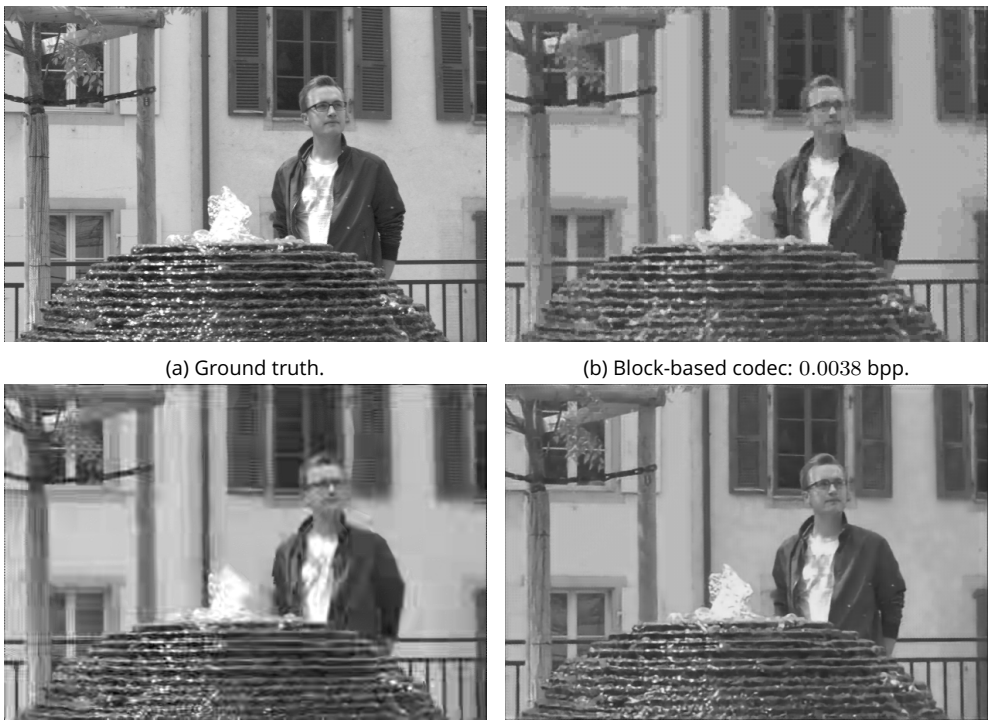
(a) Ground truth.

(b) Block-based codec: 0.0026 bpp.

(c) HEVC (x265): 0.0047 bpp.

(d) Scalable codec: 0.0029 bpp.

Figure A.3: Visual evaluation of the central view of *Pillars* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d).



(a) Ground truth.

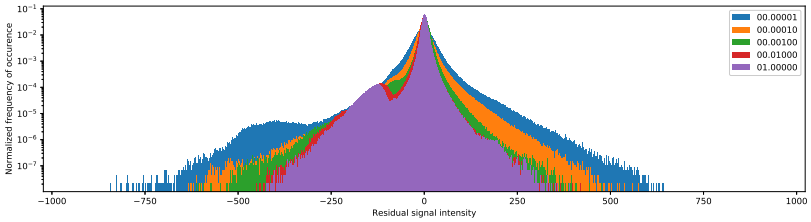
(b) Block-based codec: 0.0038 bpp.

(c) HEVC (x265): 0.0051 bpp.

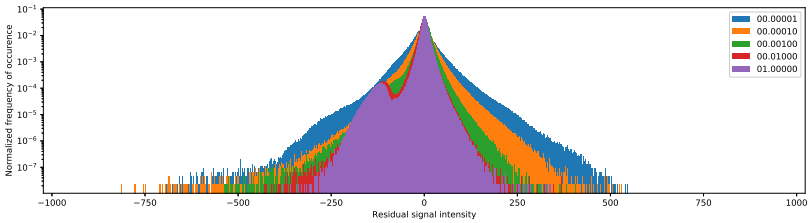
(d) Scalable codec: 0.0039 bpp.

Figure A.4: Visual evaluation of the central view of *Fountain* content (a) compressed with block-based approach (b), x265 (c) and scalable approach (d).

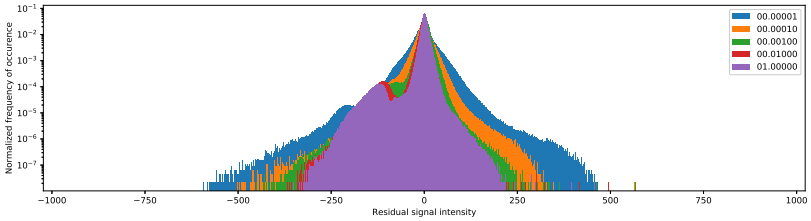
A.2 . Residual signals



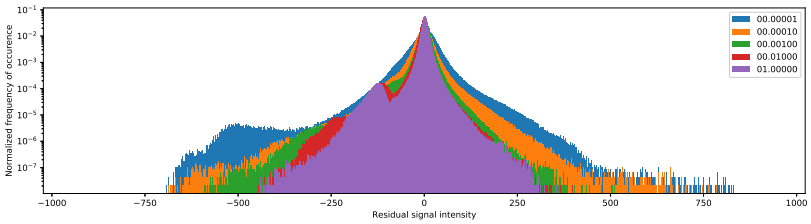
(a) *Bikes*.



(b) *Danger*.



(c) *Pillars*.

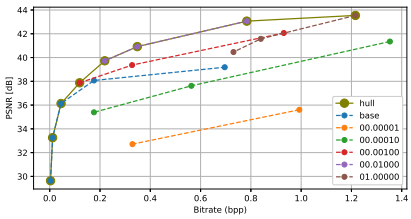


(d) *Fountain*.

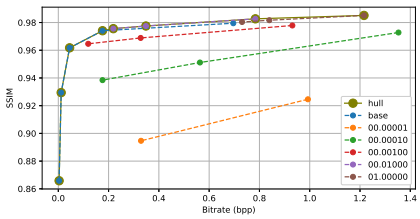
Figure A.5: Histogram of the residual signals obtained using various *Base* models.

A.3 . Convex hull generation

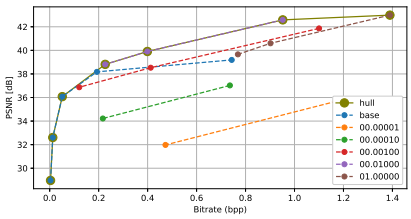
A.3.1 . Scalar Quantization



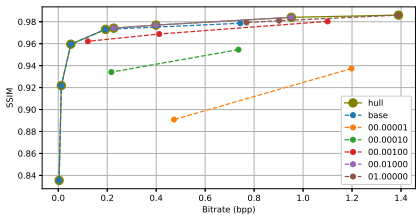
(a) *Bikes* - PSNR.



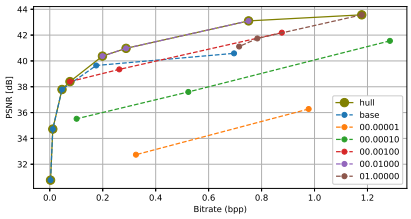
(b) *Bikes* - SSIM.



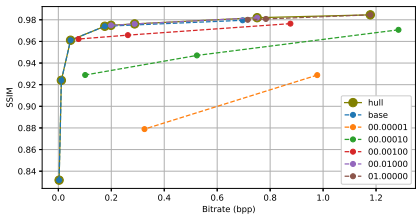
(c) *Danger* - PSNR.



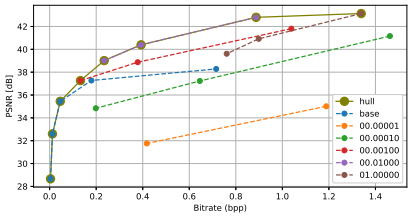
(d) *Danger* - SSIM.



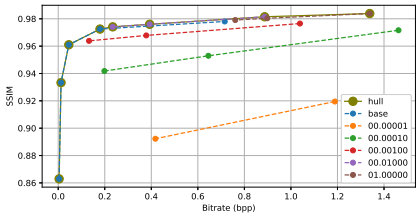
(e) *Pillars* - PSNR.



(f) *Pillars* - SSIM.



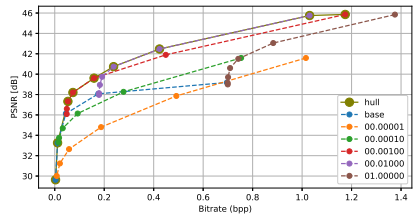
(g) *Fountain* - PSNR.



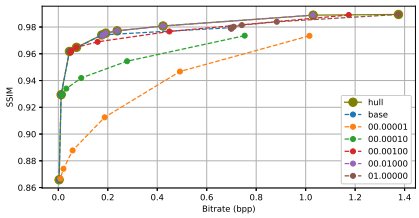
(h) *Fountain* - SSIM.

Figure A.6: Generation of the convex hull for the hybrid codec with scalar quantization-based enhancement layer in terms of PSNR and SSIM.

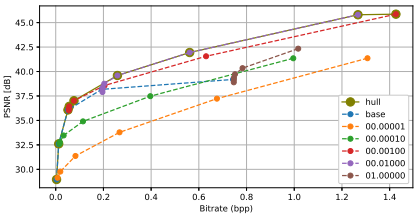
A.3.2 . Intra Coding



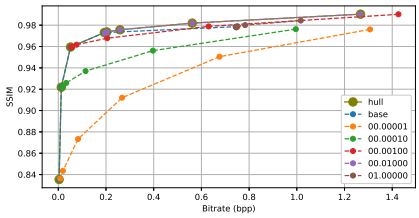
(a) *Bikes* - PSNR.



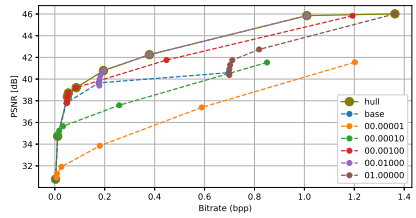
(b) *Bikes* - SSIM.



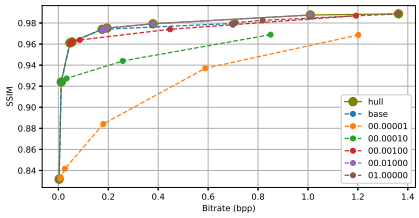
(c) *Danger* - PSNR.



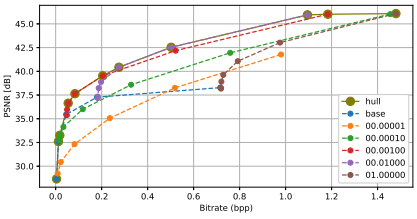
(d) *Danger* - SSIM.



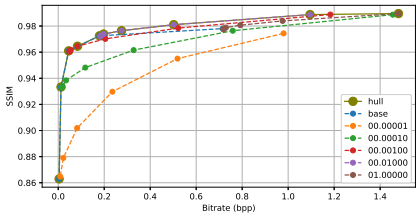
(e) *Pillars* - PSNR.



(f) *Pillars* - SSIM.



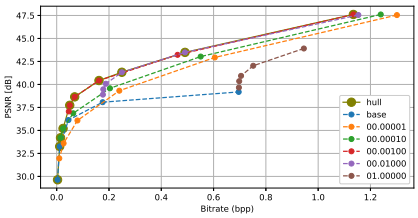
(g) *Fountain* - PSNR.



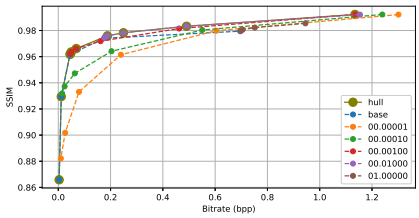
(h) *Fountain* - SSIM.

Figure A.7: Generation of the convex hull for the hybrid codec with HEVC Intra-based enhancement layer in terms of PSNR and SSIM.

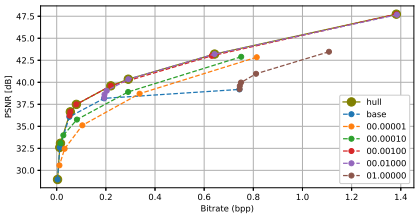
A.3.3 . Inter Coding



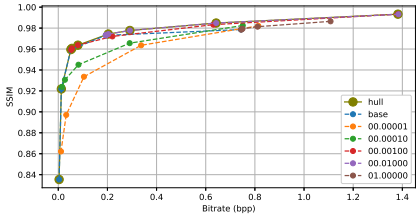
(a) *Bikes* - PSNR.



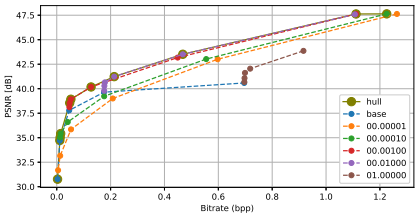
(b) *Bikes* - SSIM.



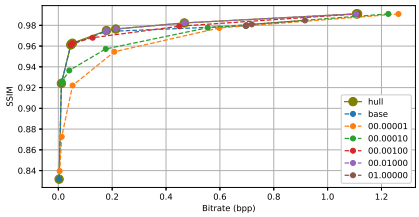
(c) *Danger* - PSNR.



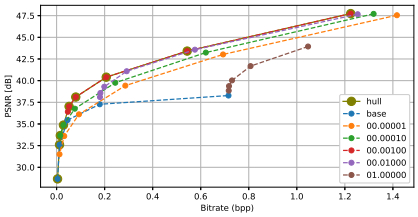
(d) *Danger* - SSIM.



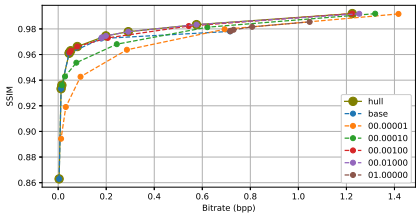
(e) *Pillars* - PSNR.



(f) *Pillars* - SSIM.



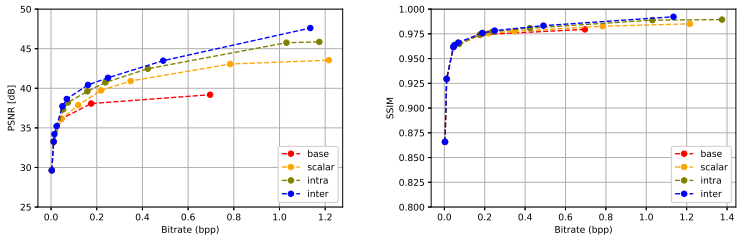
(g) *Fountain* - PSNR.



(h) *Fountain* - SSIM.

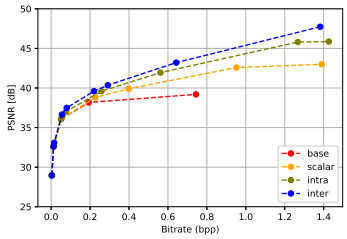
Figure A.8: Generation of the convex hull for the hybrid codec with HEVC Inter-based enhancement layer in terms of PSNR and SSIM.

A.4 . Comparison of enhancement layers

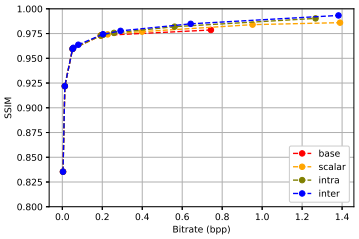


(a) *Bikes* - PSNR.

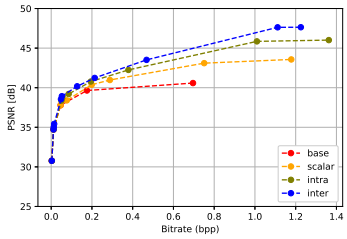
(b) *Bikes* - SSIM.



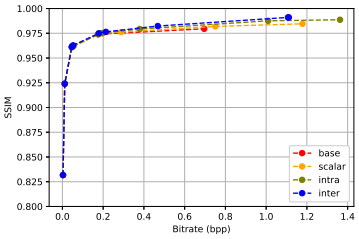
(c) *Danger* - PSNR.



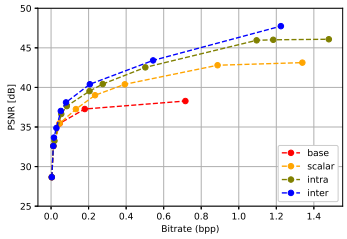
(d) *Danger* - SSIM.



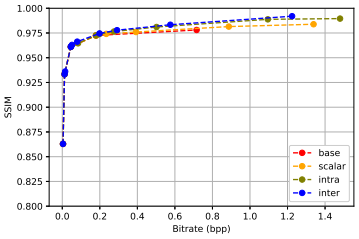
(e) *Pillars* - PSNR.



(f) *Pillars* - SSIM.



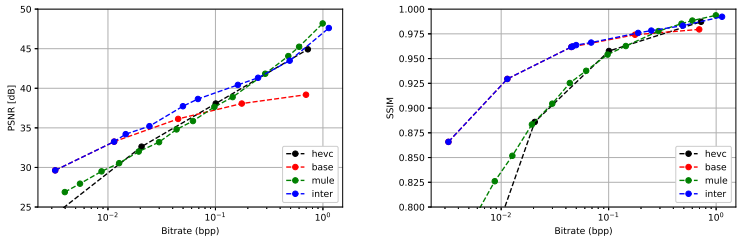
(g) *Fountain* - PSNR.



(h) *Fountain* - SSIM.

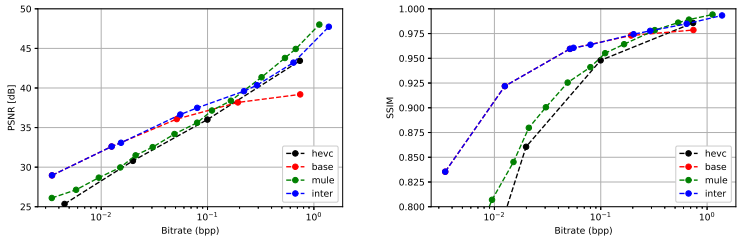
Figure A.9: RD comparison of hybrid codecs in terms of PSNR and SSIM.

A.5 . Comparison with state-of-the-art methods



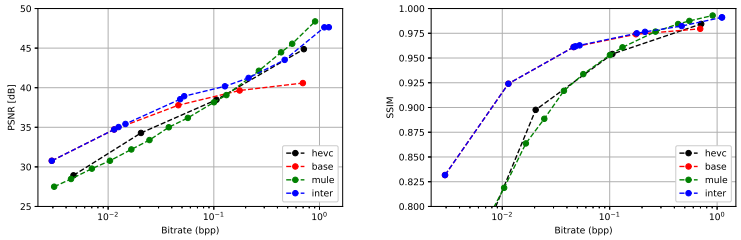
(a) *Bikes* - PSNR.

(b) *Bikes* - SSIM.



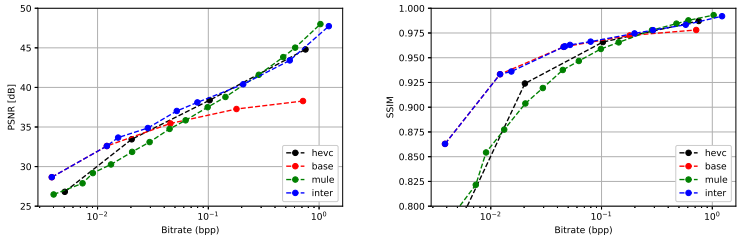
(c) *Danger* - PSNR.

(d) *Danger* - SSIM.



(e) *Pillars* - PSNR.

(f) *Pillars* - SSIM.



(g) *Fountain* - PSNR.

(h) *Fountain* - SSIM.

Figure A.10: RD comparison of the proposed hybrid codec and state-of-the-art methods in terms of PSNR (left) and SSIM (right).

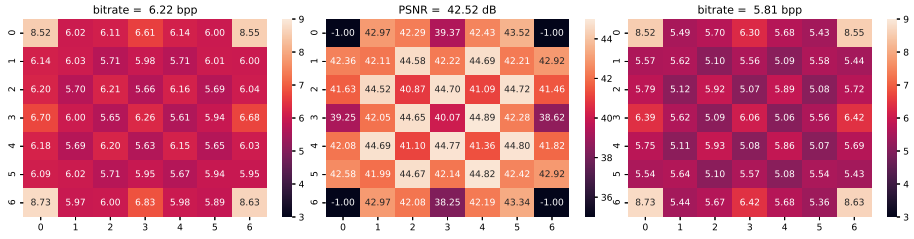
B - Learning-based Lossless Light Field Compression

B.1 . *Base-128* scheme

Table B.1: The performance evaluation in terms of bitrate (bpp) of variant *Base-128*.

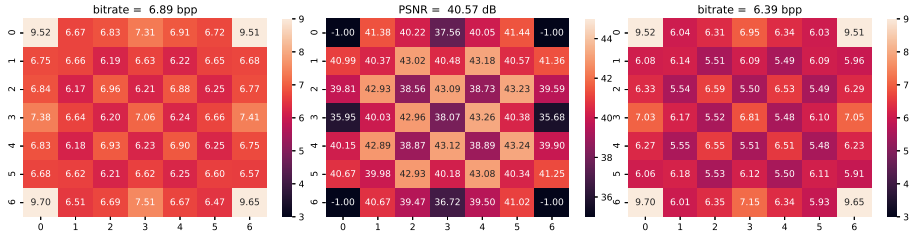
	Single		Hierarchical		
Sequence	Corner	Cross	Corner	Cross	Hybrid
Bikes	6.74	6.91	6.11	6.24	6.04
Danger	7.52	7.71	6.83	6.98	6.74
Flowers	7.51	7.74	6.75	6.94	6.66
Pillars	6.88	7.07	6.31	6.38	6.23
Vespa	7.05	7.14	6.08	6.21	5.99
Ankylosaurus	5.03	5.14	4.80	4.89	4.77
Desktop	7.25	7.34	6.59	6.75	6.52
Magnets	5.10	5.21	4.85	4.95	4.82
Fountain	7.02	7.14	6.17	6.25	6.07
Friends	6.26	6.41	5.84	6.00	5.77
Color Chart	7.27	7.24	5.93	6.25	5.83
ISO Chart	6.65	6.77	5.95	5.90	5.84
Average	6.69	6.82	6.02	6.15	5.94

B.2 . Comparison between proposed scheme and *Base*



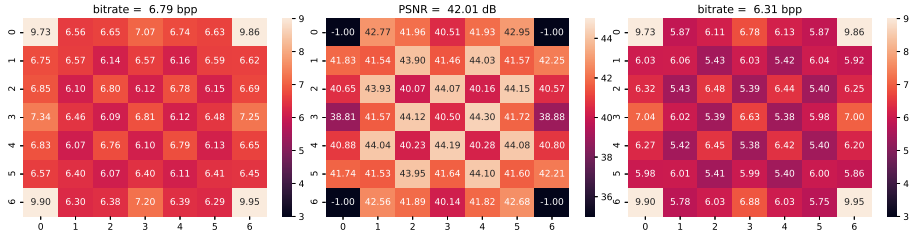
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.1: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Bikes*.



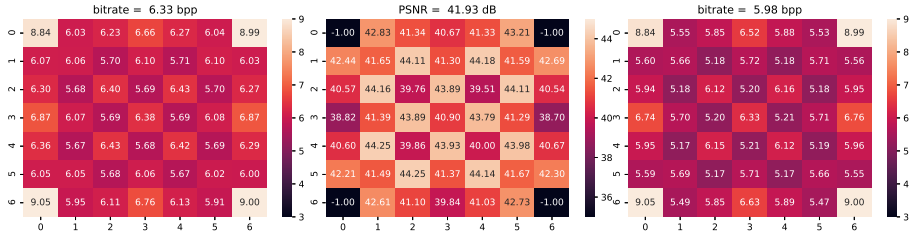
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.2: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Danger*.



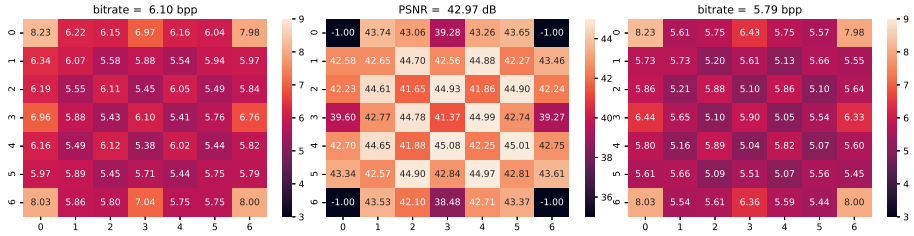
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.3: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Flowers*.



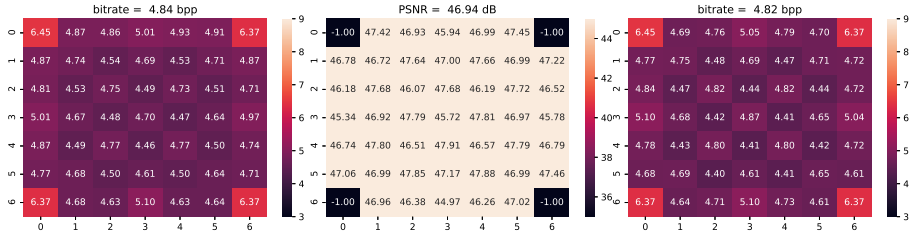
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.4: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Pillars*.



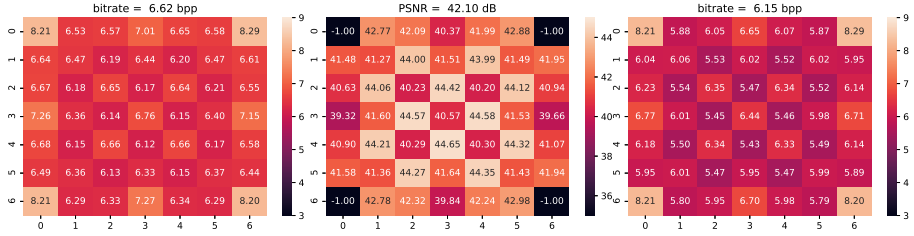
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.5: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Vespa*.



(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.6: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Ankylosaurus*.

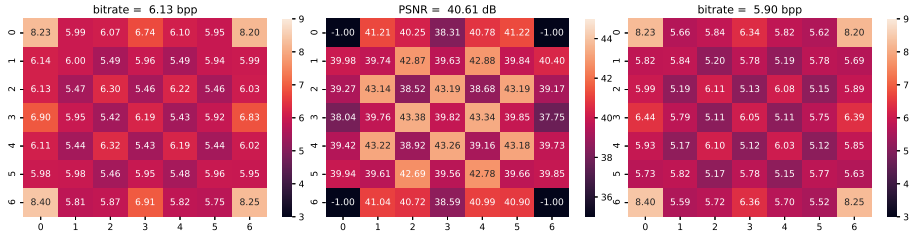


(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.7: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Desktop*.

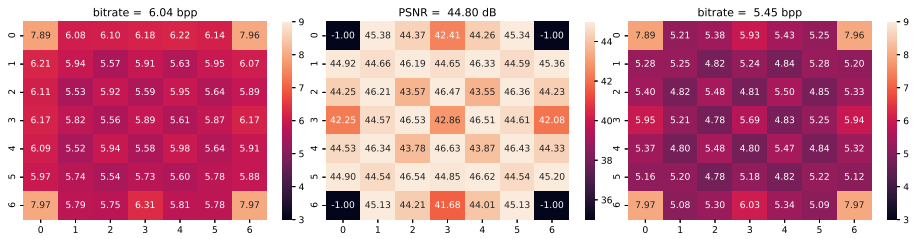
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.8: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Magnets*.



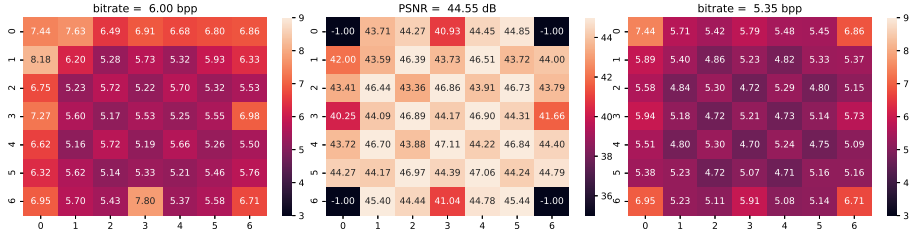
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.9: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Fountain*.



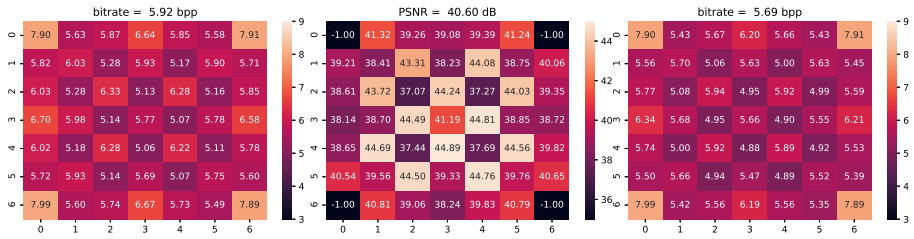
(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.10: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Friends*.



(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.11: Distribution of bitrates (top) and PSNRs (bottom) across views for content *Color Chart*.



(a) *Base* scheme. (b) Performance of view synthesis. (c) Proposed scheme.

Figure B.12: Distribution of bitrates (top) and PSNRs (bottom) across views for content *ISO Chart*.

C - Résumé en français

C.1 . Introduction

La tendance actuelle en matière de technologie d'imagerie est d'aller au-delà de la représentation Two-Dimensional (2D) du monde capturée par une caméra conventionnelle. Les exemples incluent la haute définition, la plage dynamique élevée et la vidéo à fréquence d'images élevée pour offrir une expérience utilisateur plus réaliste. La stéréo et la multivue augmentent l'expérience de la perception Three-Dimensional (3D) en simulant la profondeur et en se rapprochant de la présentation de l'expérience du monde réel. Enfin, la technologie Light Field (LF) nous permet de capturer des repères directionnels plus riches. Avec la disponibilité récente des caméras LF portatives, il est possible de capturer facilement une scène sous différentes perspectives en un seul temps d'exposition, ce qui permet de nouvelles applications. Par exemple, la synthèse rapide de nouvelles vues permet une transition en douceur à partir de différentes perspectives pour offrir une expérience visuelle plus naturelle. Des données LF riches échantillonnent les rayons lumineux d'une scène et permettent diverses manipulations après la capture. Les manipulations incluent un changement de perspective, la mise au point à différentes profondeurs de la scène et l'édition de la profondeur de champ, des champs étroits à la profondeur de champ étendue.

Alors que le nouveau modèle d'imagerie repousse les limites de l'immersion, de la qualité de l'expérience et de la photographie numérique, il génère d'énormes quantités de données exigeant des ressources de stockage et de bande passante importantes. Pour surmonter ces défis, les LF nécessitent le développement de schémas de codage efficaces. Bien qu'il existe des techniques standard efficaces de compression, ces méthodes ne sont pas aussi efficaces dans la nouvelle structure que dans le cas de l'imagerie traditionnelle. La présence croissante de données immersives et le manque de schémas de codage efficaces ont motivé le nombre élevé de travaux de recherche sur le codage LF, qui ont récemment abouti au développement d'une norme appelée JPEG Pleno.

L'apprentissage en profondeur est extrêmement efficace pour apprendre les caractéristiques les plus fines des données sous-jacentes. L'efficacité a été démontrée dans de nombreux domaines ces dernières années. De la régression aux tâches de classification, les approches d'apprentissage en profondeur surpassent rapidement les méthodologies conventionnelles, qui ont été conçues avec précision et réflexion sur de nombreuses années. Néanmoins, il reste encore de la place pour s'améliorer et pour explorer de nouveaux domaines. Cela vaut particulièrement dans le cas du traitement LF, où l'apprentissage en profondeur a apporté une nouveauté dans diverses applications de traitement, y compris l'estimation de la profondeur, la super-résolution spatiale, la super-résolution angulaire et la compression.

Cette thèse vise à explorer des approches basées sur l'apprentissage profond pour la compression LF. Nous proposons un schéma de codage hybride qui combine une approche de compression basée sur l'apprentissage avec un schéma de codage vidéo traditionnel pour la compression avec perte d'images LF et une méthode basée sur l'apprentissage pour la compression LF sans perte. De plus, nous étudions le paradigme de codage source distribué pour la compression LF.

C.2 . Motivations

Les sous-sections suivantes motivent chaque partie du travail, y compris les méthodologies basées sur l'apprentissage pour la compression LF avec perte, sans perte et distribuée.

C.2.1 . Compression avec perte basée sur l'encodeur automatique

Les auto-encodeurs sont des réseaux de neurones formés pour reproduire leur entrée à leur sortie. Ils se composent d'un encodeur qui crée une représentation et d'un décodeur qui reconstruit l'entrée à partir de la représentation. Habituellement, une contrainte est définie, ce qui empêche d'apprendre à reproduire parfaitement l'entrée et oblige à apprendre les caractéristiques essentielles des données. Cette conception a été récemment proposée pour la compression d'image avec perte. Un nouveau composant est ajouté à l'auto-encodeur, qui modélise la distribution de probabilité de la représentation. La distribution de probabilité estimée permet de calculer le coût en bits de la représentation. Le coût en bits est minimisé avec la distorsion de reconstruction, ce qui permet d'apprendre les paramètres d'un codec avec perte qui fonctionne en un point unique sur la courbe débit-distorsion. Le cadre d'apprentissage est attrayant car il propose d'apprendre les paramètres de l'ensemble du système de codage de bout en bout. En revanche, les codecs traditionnels nécessiteraient de concevoir manuellement le cadre et d'optimiser indépendamment différents blocs.

Les codecs d'image basés sur l'encodeur automatique ont atteint des performances de pointe en quelques années seulement, ce qui est un développement fantastique compte tenu du parcours des outils de codage d'image standard. Naturellement, le développement a motivé l'exploration de cette architecture pour le codage d'images LF. Avant notre travail, les encodeurs automatiques n'étaient pas explorés pour le codage LF.

C.2.2 . Modèles autorégressifs profonds pour une compression sans perte

Les modèles autorégressifs prennent en compte les échantillons précédents d'une séquence pour modéliser les échantillons actuels. Récemment, ces modèles ont été intégrés dans des méthodes d'apprentissage profond pour modéliser la distribution d'images naturelles. Par rapport, par exemple, aux réseaux antagonistes génératifs, qui sont également des modèles génératifs, les modèles autorégres-

sifs modélisent explicitement la distribution des données sous-jacentes. De plus, comme la distribution conjointe est factorisée en un produit de distribution conditionnelle, elle fournit des vraisemblances traitables. Les méthodes d'apprentissage en profondeur basées sur des modèles autorégressifs tels que PixelRNN et PixelCNN [99] ont démontré des résultats de pointe pour la modélisation d'images naturelles. La traçabilité de la vraisemblance estimée et les performances supérieures font de ces approches des candidats parfaits pour la compression sans perte. Ces méthodes ont déjà été proposées pour la tâche de codage d'image sans perte, ce qui se traduit par des performances compétitives par rapport aux outils de codage standard.

Compte tenu du potentiel de ces modèles, une question évidente sur leur utilité se pose. Nous considérons d'abord ces modèles pour la compression sans perte d'images LF.

C.2.3 . Codage source distribué en profondeur

Le codage de source distribué est un paradigme de codage non conventionnel qui permet une flexibilité dans la distribution des ressources de calcul (complexité) entre un codeur et un décodeur. Il a été proposé de pousser la plus grande complexité de calcul du côté du décodeur pour fonctionner sur des systèmes d'acquisition avec des ressources limitées. Ce paradigme a déjà été proposé pour le codage LF, mais le domaine était considérablement inexploré par rapport au travail effectué, par exemple, sur le codage vidéo distribué.

Motivés par la perspective de faire avancer la recherche sur le codage LF distribué et les capacités de modélisation élevées des méthodes d'apprentissage en profondeur, nous envisageons d'améliorer les blocs critiques dans les schémas de codage distribués.

C.3 . Objectifs et contributions

La capacité de transmission et les ressources de stockage limitent la quantité d'informations à conserver. Le problème est encore exagéré pour le contenu LF en raison de la grande quantité d'informations. La compression avec perte permet de préserver les caractéristiques les plus importantes dans des conditions de transmission limitées. La compression sans perte facilite les mécanismes de prédiction pour obtenir efficacement une représentation compacte du signal d'entrée. Enfin, le codage de source distribué fonctionne efficacement avec des ressources de calcul limitées du côté de l'acquisition. Cette thèse vise à explorer des méthodes d'apprentissage profond pour le codage efficace de contenus plénoptiques pour ces tâches.

Nous observons les limitations suivantes dans les schémas de codage traditionnels pour le codage LF:

1. Les schémas de codage traditionnels reposent sur des blocs de traitement conçus à la main sur la base de certaines heuristiques observées.

2. Chaque bloc fonctionnel est généralement optimisé indépendamment, ce qui limite les performances du pipeline de traitement global.

3. Le manque d'approches basées sur les données peut découvrir des indices délicats et comprendre les relations entre eux.

Dans cette thèse, les objectifs suivants sont fixés :

1. **Codage LF avec perte.** Explorer l'utilité des approches entraînées de bout en bout axées sur les données pour le codage LF.

2. **Codage LF sans perte.** Étudier une conception de la méthodologie de codage et des alternatives basées sur l'apprentissage pour les blocs de codage conventionnels.

3. **Codage LF distribué.** Étudier des alternatives basées sur l'apprentissage aux blocs de traitement standard dans les schémas distribués.

Nous définissons des objectifs à un niveau élevé, tandis que des objectifs détaillés et des questions de recherche sont présentés au début de chaque chapitre décrivant une méthodologie particulière.

Les apports de la thèse sont décrits ci-dessous :

1. Proposition d'un schéma de codage de bout en bout pour la compression LF avec perte.

2. Proposition d'un schéma de codage hybride pour la compression LF avec perte.

3. Introduction d'un modèle autorégressif pour le codage LF.

4. Conception d'un schéma de codage de compression LF sans perte.

5. Évaluation d'une méthode de synthèse de vues basée sur l'apprentissage dans le scénario de codage conventionnel et distribué.

C.4 . Aperçu de la thèse

La thèse est organisée comme suit. Le chapitre 2 présente les LFs et les tâches importantes de l'imagerie LF. Le chapitre 3 donne un aperçu des technologies de codage LF et inclut les solutions adoptées dans la récente norme JPEG Pleno. Les chapitres 4, 5 et 6 décrivent respectivement les schémas proposés pour la compression d'image avec perte, sans perte et distribuée LF. Le dernier chapitre, Chapitre 7, conclut la thèse par les résumés des travaux proposés et les perspectives.

Bibliography

[1] A. Aaron, P. Ramanathan, and B. Girod. Wyner-Ziv coding of light fields for random access. In *Workshop on Multimedia Signal Processing*, pages 323–326. IEEE, 2004. 92, 93

[2] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 1, 1991. 21

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*, pages 294–302, 2004. 32

[4] W. Ahmad, R. Olsson, and M. Sjöström. Interpreting plenoptic images as multi-view sequences for improved compression. In *International Conference on Image Processing (ICIP)*, pages 4557–4561. IEEE, 2017. 40

[5] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson. Shearlet transform-based light field compression under low bitrates. *IEEE Transactions on Image Processing*, 29:4269–4280, 2020. 42

[6] J. Alakuijala, R. van Asseldonk, S. Boukortt, M. Bruse, I.-M. Comșa, M. Firsching, T. Fischbacher, E. Kliuchnikov, S. Gomez, R. Obryk, et al. JPEG XL next-generation image compression architecture and coding tools. In *Applications of Digital Image Processing XLII*, volume 11137, page 111370K. International Society for Optics and Photonics, 2019. 84

[7] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret. The DISCOVER codec: Architecture, techniques and evaluation. In *Picture Coding Symposium (PCS)*. IEEE, 2007. 11, 92, 94, 104

[8] J. Ascenso, C. Brites, F. Dufaux, A. Fernando, T. Ebrahimi, F. Pereira, and S. Tubaro. The VISNET II DVC codec: architecture, tools and performance. In *European Signal Processing Conference*, pages 2161–2165. IEEE, 2010. 92

[9] J. Ascenso, P. Akayzi, M. Tesolina, A. Boev, and E. Alshina. Performance evaluation of learning based image coding solutions and quality metrics. Technical report, ISO/IEC JTC 1/SC29/WG1 N85013, San Jose, USA, 2019. 49

[10] P. Astola and I. Tabus. WaSP: Hierarchical Warping, Merging, and Sparse Prediction for Light Field Image Compression. In *European Workshop on*

Visual Information Processing (EUVIP), pages 1–6, 2018. doi: 10.1109/EUVIP.2018.8611756. 41, 45

[11] P. Astola, L. A. da Silva Cruz, E. A. da Silva, T. Ebrahimi, P. G. Freitas, A. Gilles, K.-J. Oh, C. Pagliari, F. Pereira, C. Perra, et al. JPEG Pleno: Standardizing a coding framework and tools for plenoptic imaging modalities. *ITU Journal: ICT Discoveries*, 2020. 46

[12] N. Bakir, W. Hamidouche, O. Déforges, K. Samrouth, and M. Khalil. Light field image compression based on convolutional neural networks and linear approximation. In *International Conference on Image Processing (ICIP)*, pages 1128–1132. IEEE, 2018. 43

[13] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 60

[14] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018. 50, 51, 52, 58, 60, 61, 62

[15] G. Bjøntegaard. Calculation of average PSNR differences between RD-curves. *VCEG-M33*, 2001. 111

[16] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987. 28

[17] C. Brites and F. Pereira. Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding. *Transactions on Circuits and Systems for Video Technology*, 18:1177–1190, 2008. 93, 98, 99

[18] C. Brites and F. Pereira. Distributed video coding: Assessing the HEVC upgrade. *Signal Processing: Image Communication*, 32:81–105, 2015. 109

[19] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod. Light field compression using disparity-compensated lifting and shape adaptation. *IEEE Transactions on Image Processing*, 15(4):793–806, 2006. 36, 38

[20] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision (ICCV)*, pages 1511–1520. IEEE, 2017. 98

[21] Z. Cheng, P. Akyazi, H. Sun, J. Katto, and T. Ebrahimi. Perceptual quality study on deep learning based image compression. In *International Conference on Image Processing (ICIP)*, pages 719–723. IEEE, 2019. 49

[22] R. Conceição, M. Porto, B. Zatt, and L. Agostini. LF-CAE: Context-adaptive encoding for lenslet light fields using HEVC. In *International Conference on Image Processing (ICIP)*, pages 3174–3178. IEEE, 2018. 40

[23] C. Conti, J. Lino, P. Nunes, L. D. Soares, and P. L. Correia. Spatial prediction based on self-similarity compensation for 3D holoscopic image and video coding. In *International Conference on Image Processing (ICIP)*, pages 961–964. IEEE, 2011. 41

[24] C. Conti, P. Nunes, and L. D. Soares. Light field image coding with jointly estimated self-similarity bi-prediction. *Signal Processing: Image Communication*, 60:144–159, 2018. 41

[25] C. Conti, L. D. Soares, and P. Nunes. Dense light field coding: A survey. *IEEE Access*, 8:49244–49284, 2020. 5, 35, 39

[26] T. M. Cover and J. A. Thomas. Elements of information theory. *John Wiley & Sons, Inc.*, page 409, 1991. 89

[27] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1027–1034, 2013. 27, 46, 53, 70, 105

[28] D. G. Dansereau, O. Pizarro, and S. B. Williams. Linear volumetric focus for light field cameras. *ACM Trans. Graph.*, 34(2):15–1, 2015. 53

[29] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni. A 4D DCT-based lenslet light field codec. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 435–439. IEEE, 2018. doi: 10.1109/ICIP.2018.8451684. 36, 37, 38, 65

[30] S. M. de Faria, J. N. Filipe, P. M. Pereira, L. M. Tavora, P. A. Assuncao, M. O. Santos, R. Fonseca-Pinto, F. Santiago, V. Dominguez, and M. Henrique. Light field image dataset of skin lesions. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3905–3908. IEEE, 2019. 31

[31] G. De Oliveira Alves, M. B. De Carvalho, C. L. Pagliari, P. G. Freitas, I. Seidel, M. P. Pereira, C. F. S. Vieira, V. Testoni, F. Pereira, and E. A. B. Da Silva. The JPEG Pleno light field coding standard 4D-transform mode: How to design an efficient 4D-native codec. *IEEE Access*, 8:170807–170829, 2020. doi: 10.1109/ACCESS.2020.3024844. 5, 36, 45

[32] E. Dib, M. Le Pendu, and C. Guillemot. Light field compression using Fourier disparity layers. In *International Conference on Image Processing (ICIP)*, pages 3751–3755. IEEE, 2019. 42

[33] M. Domański, K. Klimaszewski, M. Kurc, A. Luczak, O.Stankiewicz, and K.Wegner. Ftv: Poznam laboratory - a test light field sequence from poznan university of technology. Technical report, ISO/IEC JTC 1/SC29/WG1 M35071, Poland, Warsaw, 2014. 31

[34] M. Domański, K. Klimaszewski, M. Kurc, R. Ratajczak, O. Stankiewicz, and K. Wegner. Super-multi-view light-field images from poznan university of technology. Technical report, ISO/IEC JTC 1/SC29/WG1 M36566, Poland, Warsaw, 2015. 31

[35] F. Dufaux, W. Gao, S. Tubaro, and V. A. Distributed video coding: trends and perspectives. *EURASIP Trans. IVP*, 2010. 92

[36] T. Dumas, A. Roumy, and C. Guillemot. Autoencoder based image compression: Can the learning be quantization independent? In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1188–1192. IEEE, 2018. 61

[37] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens. JPEG Pleno: Toward an Efficient Representation of Visual Reality. *IEEE MultiMedia*, 23(4):14–20, 2016. doi: $10.1109/\mathrm{MMUL}.2016.64$. 30

[38] G. R. Esmaili and P. C. Cosman. Correlation noise classification based on matching success for transform domain Wyner-Ziv video coding. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 801–804. IEEE, 2009. 93

[39] Fraunhofer HHI. Specifications for high-density camera array (HDCA) data sets. Technical report, ISO/IEC JTC 1/SC29/WG1 M75008, Sydney, Australia, 2017. 31

[40] J. Garrote, C. Brites, J. Ascenso, and F. Pereira. Lenslet light field imaging scalable coding. In *European Signal Processing Conference (EUSIPCO)*, pages 2150–2154. IEEE, 2018. 36, 38

[41] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016. 97

[42] T. G. Georgiev and A. Lumsdaine. Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, 19(2), 2010. 27, 28

[43] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, 2005. 91, 94, 112

[44] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Conference on Computer Graphics and Interactive Techniques*, pages 43–54, 1996. 32

[45] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li. Free viewpoint switching in multi-view video streaming using wyner-ziv video coding. In *Visual Communications and Image Processing*, volume 6077, pages 298–305. SPIE, 2006. 92

[46] H. P. Hariharan, T. Lange, and T. Herfet. Low complexity light field compression based on pseudo-temporal circular sequencing. In *International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5. IEEE, 2017. 40

[47] P. Helin, P. Astola, B. Rao, and I. Tabus. Sparse modelling and predictive coding of subaperture images for lossless plenoptic image compression. In *2016 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2016. doi: 10.1109/3DTV.2016.7548953. 46

[48] P. Helin, P. Astola, B. Rao, and I. Tabus. Minimum description length sparse modeling and region merging for lossless plenoptic image compression. *Journal of Selected Topics in Signal Processing*, 11(7):1146–1161, 2017. doi: 10.1109/JSTSP.2017.2737967. 46

[49] R. S. Higa, R. F. L. Chavez, R. B. Leite, R. Arthur, and Y. Iano. Plenoptic image compression comparison between JPEG, JPEG2000 and SPITH. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications*, 3(6):1–6, 2013. 37

[50] J. Hou, J. Chen, and L.-P. Chau. Light field image compression based on bi-level view compensation with rate-distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):517–530, 2019. doi: 10.1109/TCSVT.2018.2802943. 43

[51] X. Huang and S. Forchhammer. Improved side information generation for distributed video coding. In *IEEE Workshop on Multimedia Signal Processing*, pages 223–228. IEEE, 2008. 89, 100

[52] X. Huang and S. Forchhammer. Cross-band noise model refinement for transform domain Wyner-Ziv video coding. *Image Communication*, 27(1), 2012. ISSN 0923-5965. 94, 99, 100

[53] X. Huang, C. Brites, J. Ascenso, F. Pereira, and S. Forchhammer. Distributed video coding with multiple side information. In *Picture Coding Symptosium (PCS)*, pages 1–4. IEEE, 2009. 93

[54] X. Huang, L. L. Rakêt, H. Van Luong, M. Nielsen, F. Lauze, and S. Forchhammer. Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow. In *IEEE Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2011. 93

[55] X. Huang, P. An, L. Shen, and R. Ma. Efficient light field images compression method based on depth estimation and optimization. *IEEE Access*, 6:48984–48993, 2018. 42

[56] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben. JPEG-1 standard 25 years: Past, present, and future reasons for a success. *Journal of Electronic Imaging*, 27(4):040901, 2018. 36

[57] K. Imaeda, K. Isechi, K. Takahashi, T. Fujii, Y. Bandoh, T. Miyazawa, S. Takamura, and A. Shimizu. LF-TSP: Traveling salesman problem for HEVC-based light-field coding. In *Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019. 40

[58] ISO/IEC JTC 1/SC29/WG1. Grand challenge on light field image compression. Technical report, Doc. M72022, Geneva, Switzerland, 2016. 30

[59] ISO/IEC JTC 1/SC29/WG1. JPEG Pleno: Call for proposals on light field coding. Technical report, Doc. N74014, Geneva, Switzerland, 2017. 30

[60] ISO/IEC JTC 1/SC29/WG1. JPEG Pleno: Light field coding common test conditions. Technical report, Doc. N81022, Vancouver, Canada, 2018. 54

[61] ISO/IEC JTC 1/SC29/WG1. JPEG Pleno: Use cases and requirements for light field quality assessment v1.0. Technical report, Doc. N92019, Online, 2021. 33

[62] ITU. ITU-T Recommendation T.812: Information technology – JPEG 2000 image coding system: An entry level JPEG 2000 encoder,. Standard, International Telecommunication Union, Geneva, CH, 2007. 46, 84

[63] J.-S. Jang, S. Yeom, and B. Javidi. Compression of ray information in three-dimensional integral imaging. *Optical Engineering*, 44(12):127001, 2005. 36, 37

[64] X. Jin, H. Han, and Q. Dai. Plenoptic image coding using macropixel-based intra prediction. *IEEE Transactions on Image Processing*, 27(8):3954–3968, 2018. 41

[65] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 98

[66] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35 (6):1–10, 2016. 43, 70, 71, 105

[67] H.-H. Kang, D.-H. Shin, and E.-S. Kim. Compression scheme of sub-images using Karhunen-Loève transform in three-dimensional integral imaging. *Optics communications*, 281(14):3640–3647, 2008. 36, 37

[68] E. C. Kaya and I. Tabus. Corner view disparity estimation for lossless light field compression. In *European Light Field Imaging (ELFI) Workshop*, pages 1–4, 2019. 47

[69] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017. 97, 98

[70] M. S. Khan Gul, T. Wolf, M. Bätz, M. Ziegler, and J. Keinert. A high-resolution high dynamic range light-field dataset with an application to view synthesis and tone-mapping. In *International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6, 2020. doi: 10.1109/ICMEW46912.2020.9105964. 31

[71] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 78

[72] A. Kolesnikov and C. H. Lampert. PixelCNN models with auxiliary variables for natural image modeling. In *International Conference on Machine Learning*, pages 1905–1914. PMLR, 2017. 71, 72

[73] D. Kubasov, J. Nayak, and C. Guillemot. Optimal reconstruction in Wyner-Ziv video coding with multiple side information. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 183–186. IEEE, 2007. 95

[74] M. Le Pendu, C. Guillemot, and A. Smolic. A Fourier disparity layer representation for light fields. *IEEE Transactions on Image Processing*, 28(11): 5740–5753, 2019. 42

[75] M. Levoy and P. Hanrahan. Light field rendering. In *Conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996. 21, 22, 32

[76] 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 Journal of Selected Topics in Signal Processing*, 11(7):1107–1119, 2017. doi: 10.1109/JSTSP.2017.2725198. 40

[77] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag. Coding of focused plenoptic contents by displacement intra prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1308–1319, 2016. doi: 10.1109/TCSVT.2015.2450333. 41

[78] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag. Scalable coding of plenoptic images by using a sparse set and disparities. *IEEE Transactions on Image Processing*, 25(1):80–91, 2016. doi: 10.1109/TIP.2015.2498406. 41

[79] C. Liu. *Beyond pixels: Exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 89

[80] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng. Pseudo-sequence-based light field image compression. In *International Conference on MultimediaE & Expo Workshops (ICMEW)*, pages 1–4. IEEE, 2016. doi: 10.1109/ICMEW.2016.7574674. 40

[81] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2009. 24

[82] H. V. Luong, L. L. Raket, X. Huang, and S. Forchhammer. Side information and noise learning for distributed video coding using optical flow and clustering. *IEEE Transactions on Image Processing*, IP-21:4782–4796, 2012. 94, 100

[83] K. Mader. lytro-power-tools. https://github.com/kmader/lytro-power-tools, 2018. 46, 70

[84] M. A. Magnor, A. Endmann, and B. Girod. Progressive compression and rendering of light fields. In *Vision, Modeling and Visualization*, pages 199–204. Citeseer, 2000. 36, 38

[85] M. W. Marcellin and D. S. Taubman. JPEG2000: Image compression fundamentals, standards, and practice. *International Series in Engineering and Computer Science, Secs*, 642, 2002. 36, 37

[86] T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu. Fusion schemes for multiview distributed video coding. In *European Signal Processing Conference (EUSIPCO)*, pages 559–563. IEEE, 2009. 93

[87] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool. Practical full resolution learned lossless image compression. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10629–10638, 2019. 71, 84

[88] F. Mentzer, L. V. Gool, and M. Tschannen. Learning better lossless compression using lossy compression. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6638–6647, 2020. 71, 72, 74, 75, 84

[89] A. Miyazawa, Y. Kameda, T. Ishikawa, I. Matsuda, and S. Itoh. Lossless coding of light field camera data captured with a micro-lens array and a color filter. In *International Workshop on Advanced Image Technology (IWAIT)*, pages 1–4, 2018. doi: 10.1109/IWAIT.2018.8369695. 46

[90] J. R. Monteiro, M. N. Rodrigues, M. S. Faria, and J. P. Nunes. Optimized reference picture selection for light field image coding. In *European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 40, 41, 44

[91] R. J. Monteiro, P. J. Nunes, N. M. Rodrigues, and S. M. Faria. Light field image coding using high-order intrablock prediction. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1120–1131, 2017. 41

[92] M. U. Mukati and S. Forchhammer. EPIC: Context adaptive lossless light field compression using epipolar plane images. In *Data Compression Conference (DCC)*, pages 43–52, 2020. doi: 10.1109/DCC47342.2020.00012. 46

[93] M. U. Mukati and S. Forchhammer. Epipolar plane image-based lossless and near-lossless light field compression. *IEEE Access*, 9:1124–1136, 2021. doi: 10.1109/ACCESS.2020.3047073. 46, 84

[94] M. U. Mukati, M. Stepanov, G. Valenzise, F. Dufaux, and S. Forchhammer. View synthesis-based distributed light field compression. In *International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020. 110, 111

[95] M. U. Mukati, M. Stepanov, G. Valenzise, S. Forchhammer, and F. Dufaux. Improved deep distributed light field coding. *IEEE Open Journal of Circuits and Systems*, 2:325–337, 2021. doi: 10.1109/OJCAS.2021.3073252. 78

[96] F. Murgia and D. Giusto. A comparison of raw light field lossless data compression algorithms. In *Telecommunications Forum (TELFOR)*, pages 1–4, 2016. doi: 10.1109/TELFOR.2016.7818796. 46

[97] J. Navarro and N. Sabater. Learning occlusion-aware view synthesis for light fields. *Pattern Analysis and Applications*, pages 1–16, 2021. 71, 72, 94, 97

[98] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. *Light field photography with a hand-held plenoptic camera*. PhD thesis, Stanford University, 2005. 5, 22, 23, 24

[99] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, volume 48, pages 1747–1756. PMLR, 2016. 18, 71, 76, 139

[100] M. Ouaret, F. Dufaux, and T. Ebrahimi. Fusion-based multiview distributed video coding. In *International Workshop on Video Surveillance and Sensor Networks*, pages 139–144. ACM, 2006. 93

[101] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp. Distributed video coding: selecting the most promising application scenarios. *Signal Processing: Image Communication*, 23(5):339–352, 2008. 89

[102] F. Pereira, C. Pagliari, E. da Silva, I. Tabus, H. Amirpour, M. Bernardo, and A. Pinheiro. ISO/IEC JTC 1/SC29/WG1: JPEG Pleno: Light field coding common test conditions v3.3. Technical report, Doc. N84025, Brussels, Belgium, 2019. 30, 33

[103] M. P. Pereira, G. Alves, C. L. Pagliari, M. B. de Carvalho, E. A. da Silva, and F. Pereira. A geometric space-view redundancy descriptor for light fields: Predicting the compression potential of the JPEG Pleno light field datasets. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2017. 30, 51

[104] C. Perra. On the coding of plenoptic raw images. In *Telecommunications Forum (TELFOR)*, pages 850–853. IEEE, 2014. 37

[105] C. Perra. Lossless plenoptic image compression using adaptive block differential prediction. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1231–1234. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178166. 46

[106] C. Perra, P. Astola, E. A. B. da Silva, H. Khanmohammad, C. Pagliari, P. Schelkens, and I. Tabus. Performance analysis of JPEG Pleno light field coding. In *Applications of Digital Image Processing XLII*, volume 11137, pages 402 – 413. International Society for Optics and Photonics, SPIE, 2019. 109

[107] C. Perwass and L. Wietzke. Single lens 3D-camera with extended depth-of-field. In *Human Vision and Electronic Imaging XVII*, volume 8291, pages 45–59. SPIE, 2012. 22

[108] H. PhiCong, S. Perry, and X. HoangVan. Adaptive content frame skipping for Wyner–Ziv-based light field image compression. *Electronics*, 9(11):1798, 2020. 93

[109] S. Reed, A. van den Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, Y. Chen, D. Belov, and N. de Freitas. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, volume 70, pages 2912–2921. PMLR, 2017. 71, 72

[110] M. Rerabek and T. Ebrahimi. New light field image dataset. In *International Conference on Quality of Multimedia Experience (QoMEX)*, 2016. 7, 30, 53, 78, 104, 105

[111] M. Rerábek, T. Bruylants, T. Ebrahimi, and F. Pereira. *ICME 2016 Grand Challenge: Light-field image compression - Call for proposals and evaluation procedure*, 2016. 78

[112] I. E. Richardson. *H.264 and MPEG-4 video compression: Video coding for next-generation multimedia*. John Wiley & Sons, 2004. 94

[113] O. Rippel and L. Bourdev. Real-time adaptive image compression. In *International Conference on Machine Learning*, pages 2922–2930. PMLR, 2017. 50

[114] D. Rüefenacht, A. T. Naman, R. Mathew, and D. Taubman. Base-anchored model for highly scalable and accessible compression of multiview imagery. *IEEE Transactions on Image Processing*, 28(7):3205–3218, 2019. 38, 39

[115] W. Ryan. An introduction to LDPC codes. *CRC Handbook CSPRS*, 2004. 95

[116] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. PixelCNN++: A PixelCNN implementation with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations (ICLR)*, 2017. 71, 76

[117] M. Salmistraro, J. Ascenso, C. Brites, and S. Forchhammer. A robust fusion method for multiview distributed video coding. *EURASIP Journal on Advances in Signal Processing*, 2014(1):1–16, 2014. 89, 93, 94, 109, 110, 111

[118] J. M. Santos, P. A. A. Assunção, L. A. da Silva Cruz, L. Távora, R. Fonseca-Pinto, and S. M. M. Faria. Lossless light-field compression using reversible colour transformations. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2017. doi: 10.1109/IPTA.2017.8310154. 46

[119] J. M. Santos, P. A. Assunção, L. A. da Silva Cruz, L. M. Távora, R. Fonseca-Pinto, and S. M. Faria. Lossless coding of light field images based on minimum-rate predictors. *Journal of Visual Communication and Image Representation*, 54:21–30, 2018. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2018.03.003. 46

[120] J. M. Santos, P. A. A. Assunção, L. A. d. S. Cruz, L. M. N. Távora, R. Fonseca-Pinto, and S. M. M. Faria. Lossless compression of light fields using multi-reference minimum rate predictors. In *Data Compression Conference (DCC)*, pages 408–417, 2019. doi: 10.1109/DCC.2019.00049. 46

[121] J. M. Santos, L. A. Thomaz, P. A. A. Assunção, L. A. d. S. Cruz, L. Távora, and S. M. M. de Faria. Lossless coding of light fields based on 4D minimum rate predictors. *IEEE Transactions on Image Processing*, 31:1708–1722, 2022. doi: 10.1109/TIP.2022.3146009. 46

[122] P. Schelkens, P. Astola, E. A. Da Silva, C. Pagliari, C. Perra, I. Tabus, and O. Watanabe. JPEG Pleno light field coding technologies. In *Applications of Digital Image Processing XLII*, volume 11137. International Society for Optics and Photonics, 2019. 5, 45, 109, 110, 111

[123] I. Schiopu and A. Munteanu. Macro-pixel prediction based on convolutional neural networks for lossless compression of light field images. In *International Conference on Image Processing (ICIP)*, pages 445–449. IEEE, 2018. doi: 10.1109/ICIP.2018.8451731. 46, 47

[124] I. Schiopu and A. Munteanu. Deep-learning-based macro-pixel synthesis and lossless coding of light field images. *APSIPA Transactions on Signal and Information Processing*, 8, 2019. doi: 10.1017/ATSIP.2019.14. 47

[125] I. Schiopu, M. Gabbouj, A. Gotchev, and M. M. Hannuksela. Lossless compression of subaperture images using context modeling. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2017. doi: 10.1109/3DTV.2017.8280403. 46, 84

[126] N. P. Sgouros, D. P. Chaikalis, P. G. Papageorgas, and M. S. Sangriotis. Omnidirectional integral photography images compression using the 3D-DCT. In *Digital Holography and Three-Dimensional Imaging*. Optical Society of America, 2007. 36, 37

[127] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim. EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4748–4757, 2018. 47

[128] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 97

[129] J. Sneyers and P. Wuille. FLIF: Free lossless image format based on maniac compression. In *International Conference on Image Processing (ICIP)*, pages 66–70. IEEE, 2016. 72, 84

[130] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to synthesize a 4D RGBD light field from a single image. In *International Conference on Computer Vision (ICCV)*, pages 2243–2251, 2017. 70, 77, 102

[131] Stanford Computer Graphics Laboratory. The (new) Stanford light field archive. Technical report, http://lightfield.stanford.edu/, 2004. 31

[132] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) standard. *Transactions on Circuits and Systems for Video Technology*, 22:1649–1668, 2012. 71, 84, 89

[133] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998. 38

[134] I. Tabus and P. Helin. Microlens image sparse modelling for lossless compression of plenoptic camera sensor images. In *European Signal Processing Conference (EUSIPCO)*, pages 1907–1911, 2017. doi: 10.23919/EUSIPCO.2017.8081541. 46

[135] I. Tabus and E. Palma. Lossless compression of plenoptic camera sensor images. *IEEE Access*, 9:31092–31103, 2021. doi: 10.1109/ACCESS.2021.3059921. 46

[136] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017. 50

[137] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016. 50

[138] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2004. doi: 10.1109/CVPR.2004.1315006. 22

[139] G. Valenzise, A. Purica, V. Hulusic, and M. Cagnazzo. Quality assessment of deep-learning-based image compression. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2018. 49

[140] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive codes for distributed source coding. *Signal Processing*, 86(11):3123–3130, 2006. 94, 104

[141] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao. Data formats for high efficiency coding of Lytro-Illum light fields. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 494–497. IEEE, 2015. 40, 97

[142] I. Viola and T. Ebrahimi. Quality assessment of compression solutions for icip 2017 grand challenge on light field image coding. In *International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2018. 30, 41

[143] I. Viola, M. Řeřábek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi. Objective and subjective evaluation of light field image compression algorithms. In *Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2016. 30, 44

[144] I. Viola, H. P. Maretic, P. Frossard, and T. Ebrahimi. A graph learning approach for light field image compression. In *Applications of Digital Image Processing XLI*, volume 10752, page 107520E. International Society for Optics and Photonics, 2018. 42

[145] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 33

[146] S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4D light fields. In *Vision, Modeling and Visualization*, volume 13, pages 225–226. Citeseer, 2013. 22, 31

[147] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers*, pages 765–776, 2005. 22

[148] X. Wu and N. Memon. Context-based, adaptive, lossless image coding. *Transactions on Communications*, 45(4):437–444, 1997. 46

[149] Z. Zhao, S. Wang, C. Jia, X. Zhang, S. Ma, and J. Yang. Light field image compression based on deep learning. In *International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. 43

[150] T. Zhong, X. Jin, L. Li, and Q. Dai. Light field image compression using depth-based CNN in intra prediction. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8564–8567. IEEE, 2019. ISBN 978-1-4799-8131-1. doi: 10.1109/ICASSP.2019.8682820. 51

[151] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. In *Workshop on Statistical Signal Processing*, pages 30–33. IEEE, 2003. 92

[152] M. Ziegler, J. Keinert, N. Holzer, T. Wolf, T. Jaschke, R. op het Veld, F. S. Zakeri, and S. Foessel. Immersive virtual reality for live-action video using camera arrays. In *International Broadcasting Convention (IBC)*, pages 1–8, 2017. 22

[153] M. Ziegler, R. op het Veld, J. Keinert, and F. Zilly. Acquisition system for dense lightfield of large scenes. *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2017. 31