1 2 9 0

UNIVERSIDADE Ð COIMBRA

João Miguel Pereira da Silva Santos

Efficient Lossless Coding of Light Fields for Medical Imaging Systems

Tese no âmbito do Doutoramento em Engenharia Electrotécnica e de Computadores, Ramo de Especialização em Telecomunicações, orientada pelo Professor Doutor Luís Alberto da Silva Cruz, e co-orientada pelo Professor Doutor Sérgio Manuel Maciel de Faria e pelo Professor Doutor Pedro António Amado de Assunção, e apresentada ao Departamento de Engenharia Electroténica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

October 2021

Faculty of Sciences and Technology

Department of Electrical and Computer Engineering

Efficient Lossless Coding of Light Fields for Medical Imaging Systems

João Miguel Pereira da Silva Santos

Thesis in the scope of the PhD in Electrical and Computer Engineering, Specialisation in Telecommunications, supervised by Professor Luís Alberto da Silva Cruz, and with the co-supervision of Professor Sérgio Manuel Maciel de Faria and Professor Pedro António Amado de Assunção, and presented to the Department of Electrical and Computer Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

Coimbra, October 2021

1 2 9 0

UNIVERSIDADE Ð COIMBRA

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT), Portugal, under PhD Grant SFRH/BD/114894/2016, project DermoPleno (PEst-OE/EEI/LA0008/2016), project PlenoISLA (POCI-01-0145-FEDER-028325), the Advanced Computing Project (CPCA/A00/6782/2020) through the MACC Bob cluster, and by FCT/MCTES through national funds and when applicable co-funded by EU funds under the project UIDB/EEA/50008/2020.

I am and always will be the optimist. The hoper of far-flung hopes, and the dreamer of improbable dreams.

(The Doctor)

Acknowledgments

At long last, one of the hardest challenges I ever had to endure comes to its end. The fruitful conclusion of this thesis would not have been possible without the support of many people and institutions, to whom I am greatly indebted.

Firstly, I would like to extend my deepest gratitude to my supervisors, Prof. Luís Cruz, Prof. Sérgio Faria, and Prof. Pedro Assunção, their tireless guidance, encouragement, and knowledge were invaluable for the completion of this thesis. I thank them for the trust placed in me and for being role models of leadership and altruism. Thanks should also go to Prof. Luís Távora whose expertise and discussions helped me a great deal, especially concerning the optical side of Light Fields. I would also like to extend my deepest gratitude to Lucas Thomaz who have become both an unofficial supervisor and a cherished friend. His availability, dedication, and guidance (along with his love for the Oxford comma) were inestimable for the results achieved in the last years of this work.

I would like to extend my sincere thanks to the host institution, *Instituto de Telecomunicações*, which has provided excellent working conditions, both in terms of facilities and computing resources, that allowed me to completely focus on my work. I acknowledge, as well, the financial support provided through projects DermoPleno (PEst-OE/EEI/LA0008/2016) and PlenoISLA (POCI-01-0145-FEDER-028325).

I gratefully acknowledge the financial support of *Fundação para a Ciência e a Tecnologia* (FCT), Portugal, through the PhD Grant SFRH/BD/114894/2016, the Advanced Computing Project CPCA/A00/6782/2020 through MACC Bob cluster, and by FCT/MCTES through national funds and when applicable co-funded by EU funds under the project UIDB/EEA/50008/2020.

A fruitful working environment is much more than just its material conditions. In my 10 plus years at Instituto de Telecomunicações I was fortunate to meet some wonderful people, André Guarda, Francisco Cunha, João Carreira, João Parracho, José Filipe, Luís Lucas, Pedro Pereira, Ricardo Monteiro, Rúben António, and Rui Lourenço, that made this experience more enjoyable, through their friendship and support, both inside and outside the laboratory. A special mention should go to João Carreira and Luís Lucas whose absolutely fantastic technical capabilities gave us the Calyp raw video player and, more importantly, our local computing cluster.

I am extremely grateful to my family, my parents Zélia and Rui, and my brother Filipe. Their love and unwavering support are beyond anything I could ever expect. I hope to be able to reciprocate the affection and patience they have shown me, and always make them proud.

To my girlfriend, Diana, whose love and support have been my anchor throughout these years. I am grateful for her patience. I hope to prove worthy of her kindness, now and in the future. May we continue to grow closer together, making the best out of everything including adversities, as we managed to during the pandemic.

Finally, I would like to acknowledge everyone who has contributed in any way to this work or provided me with any assistance over the years.

To all, my sincere and profound

Thank You!

Abstract

The adoption of light field technology is often hindered by the large amount of data necessary to represent such information. In particular, applications with strict quality requirements call for compressed representations that can preserve the integrity of the original data captured from the visual scene. Such is the case of medical images which, due to technical and legal reasons, need to be archived with no quality loss. In this context, this thesis investigates efficient algorithms for lossless compression of light fields, in particular those acquired in a medical setting, using the MRP encoder as the basis framework for new solutions. The research work carried out within the scope of this thesis has produced three main contributions.

The first one comprises pre-processing methods to ease the compression of light fields using standard-compliant and other state-of-the-art image/video lossless codecs available in the literature. A simulation study is presented, using generic image and video codecs, including MRP, HEVC, JPEG-LS, JPEG 2000, and CALIC, to assess the coding of light fields with various data arrangements. The results of this study show that encoding light field in a pseudo-video sequence arrangement is an efficient approach. The pre-processing algorithms developed in this research are shown to greatly improve the compression efficiency of four-dimensional transform based encoders, such as MuLE, specially in the cases where they are less efficient in comparison with the other standard light field encoder WaSP, *i.e.*, in light fields with higher disparity.

The second contribution consists in new algorithms, based on the MRP encoder, to efficiently exploit the redundancies present in the four-dimensional light field structure. These algorithms, named as 4D-MRP, DT-4D-MRP, and M-MRP, employ four-dimensional prediction and data partitioning to fully exploit the inherent spatial and angular redundancies of the four-dimensional light field representation. The proposed algorithms largely surpass previous state-of-the-art lossless encoders, including those specifically tailored for light field coding.

The third contribution proposes a hierarchical codec optimised for light fields that can trade-off compression efficiency for other functionalities like angular scalability and random access. The enhanced capabilities of the this encoder, named as H-MRP, are expected to help the faster adoption of this technology, by allowing compatibility with legacy displays and easier navigation through light field viewpoints. The H-MRP encoder is highly configurable, and is able to surpass state-of-the-art methods both in terms of compression efficiency and its random access capabilities.

Overall, the advances in lossless light field coding presented in this thesis demonstrate that the proposed methods achieve the best compression efficiency in comparison to other solutions available in the literature. Therefore these contributions establish a new state-of-art in lossless light field coding.

Keywords: Light Field Coding, Lossless Coding, Medical Images Compression, Efficient Light Field Representation formats, Minimum Rate Predictors

Resumo

A adopção da tecnologia *light field* é dificultada pela grande quantidade de dados necessários para a sua representação. Em particular, aplicações com requisitos de qualidade rigorosos exigem formatos de representação comprimidos que possam preservar a integridade dos dados originais capturados de uma cena. Este é o caso das imagens médicas que, por razões técnicas e legais, necessitam de ser arquivadas sem perda de qualidade. Neste contexto, esta tese investiga algoritmos eficientes para compressão sem perdas de *light fields*, em particular aqueles adquiridos em ambientes médicos, utilizando o codificador MRP como base para as novas soluções propostas. O trabalho de investigação desenvolvido no contexto desta tese resultou em três contribuições principais.

A primeira envolve métodos de pré-processamento que facilitam a codificação de *light fields* utilizando codificadores de imagem/vídeo genéricos do estado da arte disponíveis na literatura. Um estudo experimental é apresentado, onde se usam codecs genéricos de imagem e vídeo, incluindo MRP, HEVC, JPEG-LS, JPEG 2000 e CALIC, para avaliar a codificação de *light fields* com vários arranjos de dados. Os resultados deste estudo mostram que codificar *light fields* reorganizados numa pseudo-sequência de vídeo é uma abordagem eficiente. Os algoritmos de pré-processamento desenvolvidos nesta pesquisa demonstraram melhorar bastante a eficiência de compressão de codificadores baseados em transformadas tetradimensionais, como o MuLE, especialmente nos casos onde são menos eficientes quando comparados com o outro codificador padrão de *light field*, WaSP, *i.e.*, em *light fields* com maior disparidade.

A segunda contribuição consiste em novos algoritmos, baseados na abordagem do MRP, para explorar de forma eficiente as redundâncias presentes na estrutura tetradimensional dos *light fields*. Estes algoritmos, denominados de 4D-MRP, DT-4D-MRP e M-MRP, utilizam predição e partição tetradimensional dos dados para explorar de forma abrangente as redundâncias espaciais e angulares inerentes à representação tetradimensional de *light* *fields*. Os resultados obtidos com os algoritmos propostos ultrapassam largamente aqueles dos anteriores codificadores sem perdas do estado da arte, incluindo os especificamente adaptados para a codificação de *light fields*.

A terceira contribuição propõe um codec optimizado para a codificação hierárquica de *light fields* que pode sacrificar a eficiência de compressão por outras funcionalidades como a escalabilidade angular e o acesso aleatório. É esperado que as capacidades avançadas deste codificador, denominado H-MRP, ajudem a adopção mais rápida desta tecnologia, permitindo a compatibilidade com ecrãs antigos e navegação mais fácil dos diversos pontos de vista presentes no *light field*. O codificador H-MRP é altamente configurável, e é capaz de superar métodos do estado da arte quer em termos de eficiência de compressão, quer nas suas capacidades de acesso aleatório.

Em geral, os avanços na codificação sem perdas de *light fields* apresentados nesta tese demonstram que os métodos propostos atingem a melhor eficiência de compressão em comparação com outras soluções disponíveis na literatura. Por conseguinte, estas contribuições estabelecem um novo estado da arte em matéria de codificação sem perdas de *light fields*.

Palavras-chave: Compressão de *Light Fields*, Compressão sem Perdas, Compressão de Imagens Médicas, Formatos Eficientes para Representação de *Light Fields*, *Minimum Rate Predictors*

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

4D-MRP	Four-Dimensional Minimum Rate Predictors	76
BD	Bjøntegaard delta	15
bpp	bits-per-pixel	18
CALIC	Context based Adaptive Lossless Image Codec	19
CC	class coefficients	46
CMS	Context Modeling of Subaperture images	35
CP	class partition	46
DCT	discrete cosine transform	24
DICOM	Digital Imaging and Communications in Medicine	3
DPCM	differential pulse code modulation	29
DT-4D-MRP	Dual-Tree 4D Minimum Rate Predictors	80
EPFL	École Polytechnique Fédérale de Lausanne	16
EPI	epipolar plane image	13
EPIC	Context Adaptive Compression of Epipolar Plane Images	37
EPIP	EPI-based Predictor	38
ESP	EPI Slope based Predictor	37
GAP	Gradient-Adjusted Predictor	37
H-MRP	Hierarchical Minimum Rate Predictors	94
HCI	Heidelberg Collaboratory for Image Processing	16
HDCA	high density camera array	9
HEVC	High Efficiency Video Coding	15
HM	HEVC reference software	87
ISO	International Organization for Standardization	3
ITU	International Telecommunication Union	3
JPEG	Joint Photographic Experts Group	2

JPEG-LS	Lossless and Near-lossless Compression of Continuous-tone Still Images	19
JPEG-Pleno CfP	Call for Proposals of the JPEG-Pleno	58
JPEG-Pleno CTC	Common Test Conditions of the JPEG-Pleno	7
LF	light field	1
LOCO-I	LOw COmplexity LOssless COmpression for Images	32
LRBF	low reliability black frame	14
LZMA	Lempel-Ziv-Markov chain algorithm	28
M-MRP	Multi-reference Minimum Rate Predictors	83
MDL	minimum description length	34
MI	micro image	1
MLA	micro-lenses array	9
MP-CNN	Macro-pixel Predictive Convolutional Neural Network	32
MRP	Minimum Rate Predictors	4
MSE	mean square error	24
MuLE	Multidimensional Light Field Encoder	24
MV-HEVC	Multiview High Efficiency Video Coding	95
NC	number of classes	46
OMP	Orthogonal Matching Pursuit	34
OOMP	Optimised Orthogonal Matching Pursuit	27
PR	prediction residuals	46
PSMP-CNN	Prediction using Synthesized MPs based on Convolutional Neural Network	33
PSNR	peak signal to noise ratio	18
PVS	pseudo-video sequence	50
RA	random access	35
RAP	random access penalty	19
RAR	random access region	96
REP-CNN	Residual-Error Predictive Convolutional Neural Network	33
RKV	representative key view	95
RLE	run length encoding	46
SAI	sub-aperture image	1
SBE	Spiral-blackend	50
SH	sequence header	46
SKINL2	light field image dataset of skin lesions	17
SMC	Sparse Modelling Compression	29
SRRC	Sparse Relevant Regressors and Contexts	30
ThV	threshold values	46
TV	total variation	45

VBS	variable block size	43
VVC	Versatile Video Coding	19
WaSP	Hierarchical Warping, Merging, and Sparse Prediction	24

CHAPTER 1

Introduction

Contents

This thesis presents an investigation on efficient lossless coding of light field (LF) images. The research work is focused on encoding solutions to be used in applications where the original image data is required to be perfectly reconstructed from the compressed data, *e.g.* medical applications. Multiple coding approaches were investigated and the main findings are presented throughout this thesis, ranging from compliant coding techniques to new algorithms that can efficiently exploit the inherent data redundancies of LFs, contributing to advance the current state-of-the-art.

1.1 Context and Motivation

The light field imaging concept and fundamentals, initially proposed by Lippmann in 1908 [1], promise to provide superior immersive multimedia experiences for users and content producers. Applications of this technology range from the entertainment industry, such as virtual and augmented reality [2], 3D movies and television [3] to more critical tasks using computer vision in medical imaging [4–6], for instance. These advanced applications are only possible due to the characteristics of LFs, which provide richer visual information when compared to traditional 2D images, by capturing not only the light intensity, but also the directionality of light rays reaching the camera sensors. Such extra information enables a panoply of post-processing operations, like the extraction of depth maps [7–9], 2D image rendering with different viewing perspectives [10], or post-capture image refocusing [11].

Developments in LF acquisition systems technology are expected to make LF imaging available to a broader range of users, both industrial and domestic. Since consumer-grade devices, such as the Lytro cameras (discontinued) and Raytrix (`https://www.raytrix.de/`), appeared in the market, this technology has started to receive more attention from the research community. LF acquisition devices capture significantly more visual information of the scene than regular cameras, since besides light intensity from each point in the scene, they also record information related to the direction of the light rays travelling towards the sensor, as mentioned before. Accordingly, each sample acquired by the image sensor corresponds to the intensity of one (or more) light-ray coming from a point in the visual scene with a certain direction. This results in a sampled LF that has the form of an array of micro images (MIs), each one corresponding to the light captured through a single micro-lens, where the spatial position of each sample implicitly encodes directional information. After adequate calibration and rectification, different views, *i.e.* sub-aperture images (SAIs), of the visual scene can be obtained. Other LF acquisition set-ups based on arrays of cameras, and therefore with different sized apparatus, geometries, and imaging characteristics, have also been developed mainly for research purposes, like the camera array used to capture part of the LF data available at "The (New) Stanford Light Field Archive" [12].

Due to its ability to capture the intensity and direction of light rays [13], LF sensor data can be processed *a posteriori*, enabling several post-processing operations that can extract diverse image-related information and render images with different characteristics. Amongst the operations that can be performed on the raw data, one can mention extraction of depth maps [9, 14] and rendering of images with different focal planes, depth-of-field, or viewing perspectives [15–17].

LF images can be used for specialised purposes, such as medical diagnosis as described in the literature: in [18], LF images are used for diagnosis and control of Kaposi's sarcoma; in [19], this technology is proposed to enhance imaging in urological oncology; another application is in eye cataract diagnosis [20], where the possibility of image refocusing can help the detection and evaluation of cataract volume and opacity; in [21], a study is undertaken on the performance and possible applications of plenoptic cameras in clinical settings. Other important LF imaging applications are in industry, with uses ranging from classification of fruit [22] and quality control in manufacturing [23] to optical measurement of surface properties to volumetric recording of prised sculptures and other cultural heritage artefacts [24]. Recognising the relevance and potential widespread use of LF imaging in multiple application fields (*e.g.,* entertainment, medical, robotics, computer vision and education [25]), the Joint Photographic Experts Group (JPEG)-Pleno initiative has put significant research and standardisation efforts on the development of LF compression methods, raging from lossy, to near-lossless, and lossless [26].

The coding methods proposed in this thesis are intended to be applied in compression of LF dermoscopy images captured in a professional health environment [27]. For instance, in [28] it is shown that dermoscopy provides more accurate classification of skin lesions than naked eye examination. However, such classification can be further improved using computer vision and machine learning techniques [29]. As shown in [21], LF imaging enable the reconstruction of 3D surfaces, thus providing extra depth features, *i.e.,* height and three-dimensional shape, that can improve current classification algorithms [5, 6]. For instance, in [30] the authors found that skin surface depth information has relevant features for melanoma discrimination.

Nevertheless, the flexibility and new functionalities provided by LFs come at the cost of huge amounts of storage capacity and bandwidth required to handle all the data, which can be a potential deterrent to the development of LF imaging technologies. Much like what is done with common 2D images, data compression can be used to alleviate this problem. In a recent work [31], it was found that, to a certain extent, lossy compression may not produce significant impact on the visual quality of images obtained through LF computational methods, such as refocusing and extended focus. However, in application fields with stringent accuracy requirements, such as computer vision in industry – where high-precision depth measurements may be necessary – and medical applications (*e.g.*, LF dermoscopy) – where all relevant features captured in LF data must be preserved with high-fidelity – LF lossless compression is necessary to fulfil such requirements.

For instance, in professional health environments, due to medical and legal reasons [32] the results of medical imaging scans need to be kept and archived for several years. Additionally, the archiving process needs to guarantee that the scans are archived with the same quality as that used during the diagnostic process. For instance, the level of representation fidelity is a relevant requirement not only for maintaining the patients' records, but also for judicial proceedings whenever malpractice or other legal issues are raised in regard to medical procedures. As a consequence, databases for medical images archiving are facing a quasi-exponential growth in their contents [32], leading to increasingly higher storage costs.

Transmission and compression of imaging in clinical context are regulated by Digital Imaging and Communications in Medicine (DICOM) [33–35] international standard. DICOM is one of the most widespread healthcare standards, supported by most medical imaging devices. The use of standards is particularly important in medical imaging [36], as it assures that images can be interchangeably used and shared between the various institutions like hospitals and imaging centres. DICOM integrates common International Organization for Standardization (ISO) and International Telecommunication Union (ITU) image compression standards, both reversible (lossless) and irreversible (lossy). The standard defines an encapsulated format archive, where the compression information is included in the bitstream syntax. Despite allowing a number of coding algorithms to be used, the standard makes no assumptions or recommendations on which encoders should be applied and in which applications [34]. This is especially valid for the irreversible encoding, as there is still an open debate whether lossy compression should be used in the context of medical imaging, specially when images are used for diagnostic purposes. Regulatory bodies in the UK, EU, USA, Canada, and Australia allow the use of lossy compression for medical images. However, the ultimate decision of using irreversible compression is left to the institutions and imaging technical staff [37]. Nevertheless, reversible compression, whenever possible, is recommended by several regulatory bodies, such as the Royal Australian and New Zealand College of Radiologists [38].

1.2 Thesis Objectives and Contributions

This doctoral work aimed to advance the state-of-the-art in LF lossless coding for applications which do not tolerate distortions, such as medical imaging and high-precision vision measurement. In order to fulfil its purpose, three major objectives were defined:

1. **Research pre-processing algorithms for light field compression and representation arrangements beyond the raw uncompressed RGB format, enabling efficient compression:**

   This objective was two fold. The first goal deals with finding suitable arrangements to encode the LF data using generic state-of-the-art encoders. Although these encoders are not well suited to deal with the inherent redundancies present in LFs, accomplishing this goal allows to promote early adoption of this technology. The second goal is the development of pre-processing algorithms that can facilitate data compression operations for this type of images. The research developed under this objective resulted in two journal papers (J4 and J5, see the list of publications below) and two conference papers (C1 and C5).

2. **Research efficient prediction-based lossless coding methods adapted to light fields:**

   This objective aims at developing new techniques that are capable of exploiting the structural redundancy generated by the light field acquisition process. The goal is to develop efficient coding algorithms, based on the Minimum Rate Predictors (MRP) encoder, that are able to surpass other state-of-the-art techniques. The conclusions of this objective resulted in a journal paper (J2) and three published conference papers (C3 and C4).

3. **Research methods for light field lossless coding with enhanced capabilities:**

   This research investigated the possibility of adding partial data access functionalities to LF lossless encoding, using angular scalability and random access. The focus was on achieving good compromises between compression efficiency and capabilities of decoding partial LF information. This part of the work resulted in a journal paper (J1).

The research work presented in this thesis was carried out by the author between September 2016 and October 2021 at University of Coimbra and at the Instituto de Telecomunicações - Leiria branch. The developed work has been presented in international conferences, published in peer reviewed journals, and has resulted in some contributions to the JPEG-Pleno standardisation action. The work resulted in the following list of publications:

Journal Publications

J1. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Hierarchical Lossless Coding of Light Fields with Improved Random Access, Signal Processing: Image Communication, vol. 105, pp. 116687-116687, 2022, doi: 10.1016/j.image.2022.116687.

J2. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Coding of Light Fields based on 4D Minimum Rate Predictors, IEEE Transactions on Image Processing, vol. 31, pp. 1708-1722, 2022, doi: 10.1109/TIP.2022.3146009.

J3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Coding of Light Field Images based on Minimum-Rate Predictors, Journal of Visual Communication and Image Representation, Vol. 54C, pp. 21 - 30, April, 2018, ISSN 1047-3203, doi: 10.1016/j.jvcir.2018.03.003.

J4. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Performance evaluation of light field pre-processing methods for lossless standard coding, IEEE COMSOC MMTC Communications - Frontiers, Vol. 12, No. 4, pp. 44 - 49, July, 2017.

Peer Reviewed Conference Publications

C1. João M. Santos, Lucas A. Thomaz, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Sérgio M. M. de Faria, Disparity compensation of light fields for improved efficiency in 4D transform-based encoders, IEEE International Conference on Visual Communications and Image Processing (VCIP), Macau, China, December, 2020, pp. 112-115, doi: 10.1109/VCIP49819.2020.9301829.

C2. Lucas A. Thomaz, João M. Santos, Pekka Astola, Sérgio M. M. de Faria, Pedro A. A. Assunção, Murilo B. de Carvalho, Eduardo A. B. da Silva, Ioan Tabus, Márcio P. Pereira, Gustavo Alves, Fernando Pereira, Vanessa Testoni, Pedro Freitas, Ismael Seidel, Visually Lossless Compression of Light Fields, EURASIP European Light Field Imaging Workshop (ELFI), Borovets, Bulgaria, June, 2019.

C3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Rate-Complexity Trade-Off in Minimum Rate Predictors Light Field Lossless Encoding, EURASIP European Light Field Imaging Workshop (ELFI), Borovets, Bulgaria, June, 2019.

C4. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless compression of Light Fields using multi-reference Minimum Rate Predictors, Data Compression Conference (DCC), Snowbird, Utah, United States, March, 2019, pp. 408-417, doi: 10.1109/DCC.2019.00049.

C5. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Light-Field Compression Using Reversible Colour Transformations, International Conference on Image Processing Theory, Tools and Applications IPTA, Montreal, Canada, pp. 1-6, November, 2017, doi: 10.1109/IPTA.2017.8310154.

JPEG Documents

D1. Lucas A. Thomaz, João M. Santos, Pedro A. A. Assunção, Sérgio M. M. Faria, “WG1M84001-CTQ-JPEG Pleno Light Field ES4.1 IT-LEIRIA partial replacement report”, 84th JPEG Meeting, Brussels, Belgium.

D2. Lucas A. Thomaz, João M. Santos, Pedro A. A. Assunção, Sérgio M. M. Faria, “WG1M83038 - C&Q-JPEG Pleno Light Field ES4.1 IT-LEIRIA report”, 83rd JPEG Meeting, Geneve, Switzerland.

D3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. N. Távora, Rui Fonseca-Pinto, Sérgio M. M. Faria, “WG1M82036 - Lossless compression of Light Fields using multi-reference Minimum Rate Predictors”, 82nd JPEG Meeting, Lisbon, Portugal.

1.3 Thesis Outline

The remainder of this document is organised in six chapters as follows: Chapter 2 introduces the fundamentals pertaining to LFs that provide the essential background for the remaining chapters. Chapter 3 starts by exposing introductory concepts of lossless coding, followed by a literature review on relevant state-of-the-art lossless encoding of LF data. A description of the MRP algorithm is also presented. Chapter 4 describes the research focused on LF compression methods that are conforming with generic codecs, and pre-processing algorithms designed to improve the compression efficiency of existing encoders. Chapter 5 presents new techniques developed to efficiently exploit the redundancies arising from the LF structures. Chapter 6 proposess a LF encoder with angular scalability and random access capabilities. Finally, Chapter 7 presents the conclusions and future research directions of this thesis.

CHAPTER 2

Light Field Fundamentals

Contents

This chapter outlines the fundamental concepts pertaining to light fields (LFs), which provide the necessary background for the contributions of this thesis. Section 2.1 introduces the plenoptic function, which characterises the light rays passing at each point in space, as well as the considerations leading to its 4D simplification. Section 2.2 briefly discusses the two most common LF acquisition apparatus, namely the plenoptic camera and the array of cameras, and their variations. The same section describes the processing chain needed to convert the raw lenslet images into the common 4D representation. The characteristics of the most common arrangements of the 4D LF data, which have a huge impact in the coding efficiency, are presented in Section 2.3. Finally, Section 2.4 describes the methodology used to evaluate the proposed methods, taking into account the guidelines set by Common Test Conditions of the JPEG-Pleno (JPEG-Pleno CTC), and defines the modifications used in this work to account for the lossless compression case.

2.1 The 4D Representation of the Plenoptic Function

Light fields are sampled representations of the plenoptic function, from the Latin *plenus*, *i.e.*, complete/full, and the Greek *optikos*, *i.e.*, pertaining to vision. This function represents the intensity $\mathcal{L}$ of light rays at every point x, y, z in space, travelling in the θ, ϕ direction, with the wavelength λ, at time instant τ [39]. This translates into a seven-dimensional function:

$$\mathcal{L} = P\left(x, y, z, \theta, \phi, \lambda, \tau\right). \tag{2.1}$$

However, in practical systems, full representation of the plenoptic function is not necessary, thus a reduced dimension version is normally used. This simplified representation requires less computational effort for its acquisition, due to its inherent smaller data size. The sampled plenoptic function can be captured by several methods [13]: Omnidirectional imaging, Depth-enhanced imaging, Point cloud imaging, Holographic imaging, and Light Field imaging. Several different LF data representations have been proposed in the literature, each with distinct characteristics [40]. This thesis deals with the common 4D representation of LFs, which reduces the seven dimensions of the plenoptic function to four, based on the following three assumptions:

1. The spectral distribution of each ray (a function of the wavelength λ) can be simplified to the three RGB channels normally captured by cameras;
2. The transmission medium does not introduce attenuation, so the radiance along a light ray path remains constant. Therefore the scene information can be captured by projection onto any selected surface and the explicit z dimension can be eliminated;
3. The scene is static during the acquisition interval thus, the time dimension τ can be disregarded.

Under these assumptions, the 7D plenoptic function can be simplified to a 4D function

$$\mathcal{L}_{4D} = L(t, s, v, u) : \mathbb{R}^4 \longmapsto \mathbb{R}^3, \tag{2.2}$$

where $L(t, s, v, u)$ maps each position (t, s, v, u) to a three-dimensional value, representing the RGB colour components. A 4D parametrisation of a light field, where a light ray is defined by two pairs of coordinates defined by its intersection with the (v, u) and (t, s) planes, is shown in Figure 2.1. Light rays originated in the same scene, with different directions, converge to the same focal point in the (v, u) plane, but are are uniquely identified by the coordinates in the (t, s) plane. In other words, for each point in (v, u) plane the camera captures the intensity of light rays with different directions (t, s). In this context, (t, s) represent the angular coordinates, and (v, u) represent the spatial coordinates in the 4D LF. The two planes represented in Figure 2.1 are discretised allowing for only a finite number of viewpoints and light rays to be recorded.

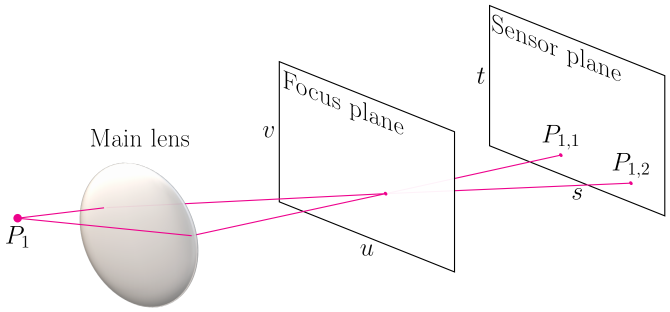


Figure 2.1: Simplified example of the 4D LF parametrisation used in this work.

2.2 Light Field Acquisition

Conventional cameras cannot capture the 4D LF structure described in the previous section, as their sensors can only capture a two-dimensional[1] approximation of a scene. Several processes for capturing a LF exist resorting to object side coding or sensor side coding, for instance, some of which are described by Zhou in [13]. The combination of the usual 2D information captured by conventional camera sensors, in combination with the angular information resulting from the various perspectives, allows for the construction of the previously described 4D LF structure.

Two main approaches for the acquisition of LFs are used in this thesis. The first approach uses plenoptic cameras, which are modified conventional cameras that use a micro-lenses array (MLA) in front of the main camera sensor to capture the extra information. The second one uses conventional cameras, arranged either in a fixed array or using a robotic arm to displace a single camera. Due to its characteristics this approach is called high density camera array (HDCA).

2.2.1 Plenoptic Cameras

Plenoptic cameras are single sensor devices developed for the acquisition of LFs, using a MLA placed between the main lens and the sensor to capture both spatial and angular information. In this type of cameras, a trade-off between angular and spatial resolution is present, related to the main sensor resolution and the MLA grid resolution. Each micro-lens of the array allows the camera to capture the angular information of a scene, instead of the conventional 2D image. Two main types of plenoptic cameras can be found: the unfocused camera [41], also known as plenoptic 1.0, and the focused camera [42], also known as plenoptic 2.0. Figure 2.2 shows a schematic representation of each of these types of cameras. Despite the differences in the acquisition processes, the LF can always be represented in terms of the 4D data structure described in Section 2.1.

The main difference between the two types of plenoptic cameras is the way images are formed in the sensor. In the unfocused camera, shown in Figure 2.2a, the main lens is focused at the

[1] In this analysis the colour of a captured image is not considered. If the colour information is considered the output is a three-dimensional image, *i.e.*, width, height, and colour.

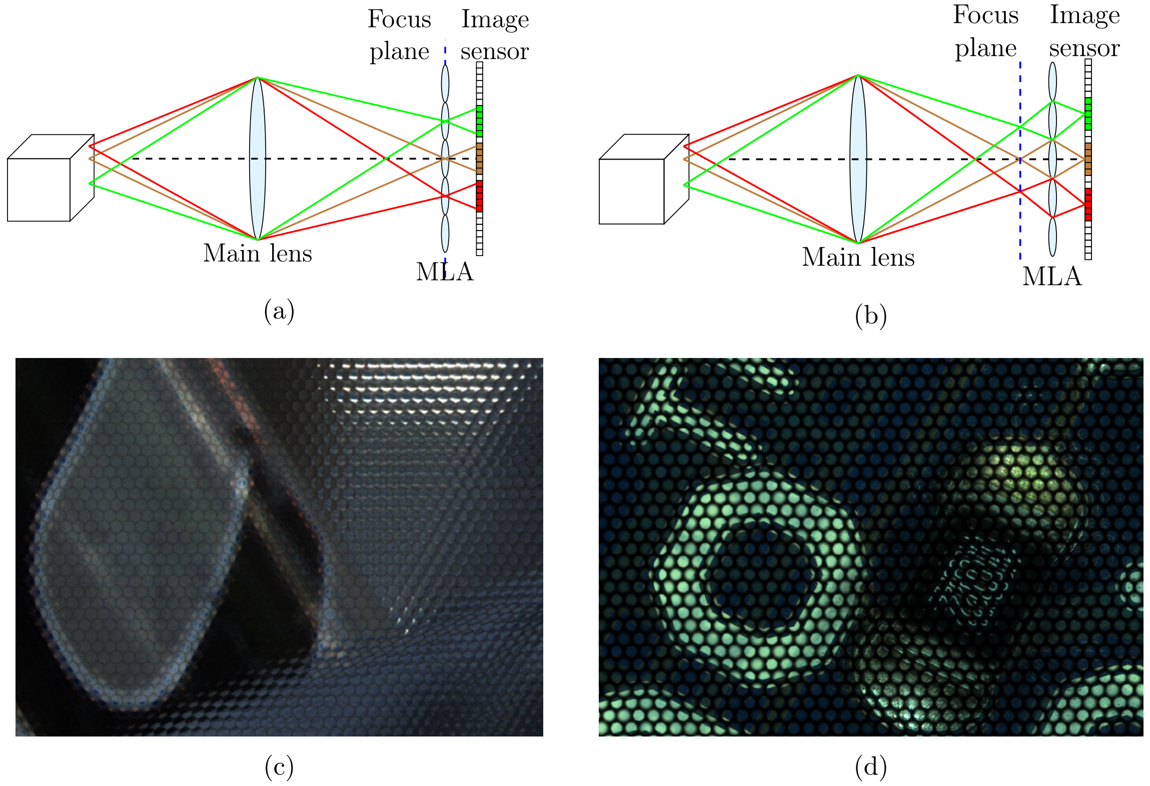
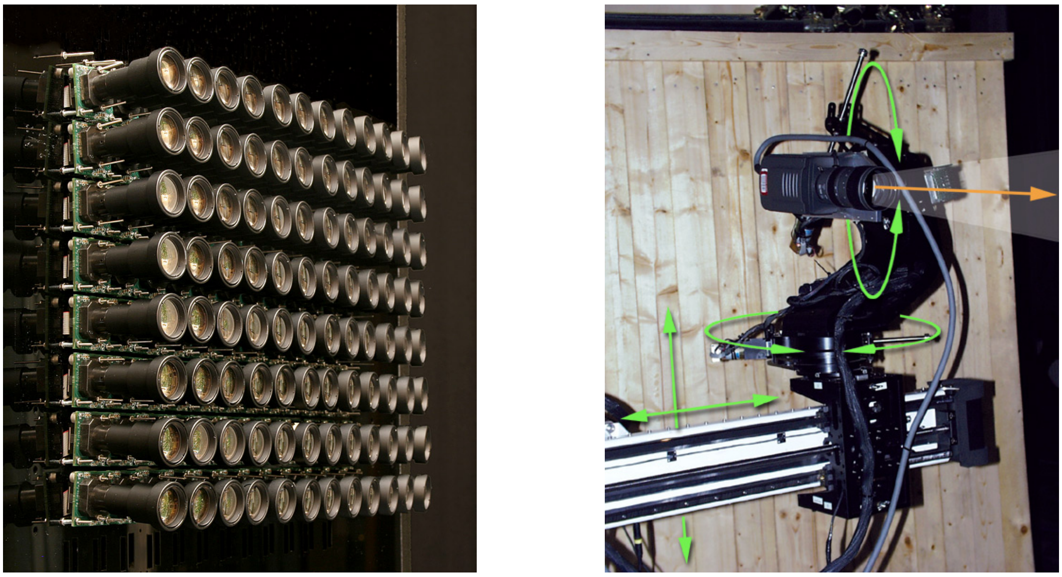


Figure 2.2: Example of plenoptic acquisition systems: (a): unfocused camera, (b): focused camera (MLA focused on the real image), (c): detail of an image captured with an unfocused camera [43], and (d): detail from an image captured with focused camera.

MLA, which are placed at a focal length distance of the main sensor. Consequently, in this arrangement the micro-lenses are focused at optical infinity. Due to the previous considerations, each micro-lens captures only angular information. Thus, each pixel in the micro image (MI) corresponds to a different viewpoint, which limits the spatial resolution of the LF. In the focused plenoptic camera, shown in Figure 2.2b, the micro-lenses are focused on the image produced by the main lens [42], which allows for the creation of focused MIs. There are two types of focused plenoptic cameras, differing in the relative position of the main lens focus plane, which can be in front or behind the MLA. Thus, the micro-lenses are either focused on the real image, as shown in Figure 2.2b, or on the virtual image, respectively. This approach allows for higher spatial resolution rendering, albeit with some sacrifice on the angular resolution. An example of an image acquired with each of these cameras is shown in Figures 2.2c and 2.2d, for unfocused and focused cameras, respectively.

2.2.2 High Density Camera Array

The high density camera array LF acquisition approach follows the process used to capture stereo or multiview video scenes, using multiple side-by-side cameras. However, instead of using an acquisition apparatus with linear horizontal camera placement, in the case of to 4D LF acquisiton, two-dimensional arrays of cameras are generally used. Figure 2.3 shows two examples of HDCA acquisition apparatus, Figure 2.3a shows a camera array, and Figure 2.3b shows a camera mounted on a gantry. The gantry apparatus allows to emulate a wide array of cameras distributions using a single camera.



(a)

(b)

Figure 2.3: Example of a HDCA acquisition systems from the Stanford University: (a): camera array [44], (b): camera gantry [45].

Camera arrays can be configured to capture LFs with various characteristics. The distribution of the cameras allows to adapt for a parallel or convergent camera arrangements, and the baseline between cameras can be adjusted to a particular application, for instance. The angular and spatial resolution of HDCA LF, are limited by the number of cameras and the camera sensor resolution, respectively. The gantry acquisition apparatus generally have the same characteristics of the camera arrays. Nevertheless, the camera placement is more flexible in this case, as the cameras are not constrained to a pre-defined grid. In turn, this allows a more fine definition of the distance between acquisition points. However, as it relies in only one camera, each perspective is, necessarily, acquired at a different time interval, which means that time-varying conditions (like lighting intensity variations) may induce distortions in subsequent LF processing.

2.2.3 Light Field Processing Chain

Following the description of the LF acquisition methods it is important to introduce the LF processing chain for plenoptic cameras. For these cameras, a set of pre-processing operations are needed to convert the raw lenslet format acquired by the camera to the conventional 4D LF described in Section 2.1. The most common approach to convert the raw lenslet data, resulting from unfocused cameras, is the one proposed by Dansereau in [46], as implemented by the LF toolbox [47]. In its call for proposals, the JPEG-Pleno standardisation action adopted the LF toolbox as the decoding step for the raw plenoptic images. Consequently, it was also adopted in this thesis, with some adaptions resulting from the lossless nature of the proposed algorithms. The diagram of the reference workflow, as described in [26], is shown in Figure 2.4.

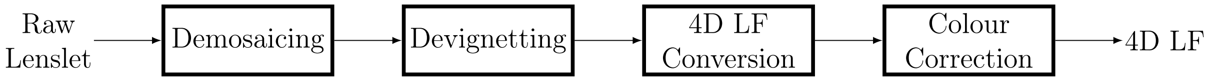
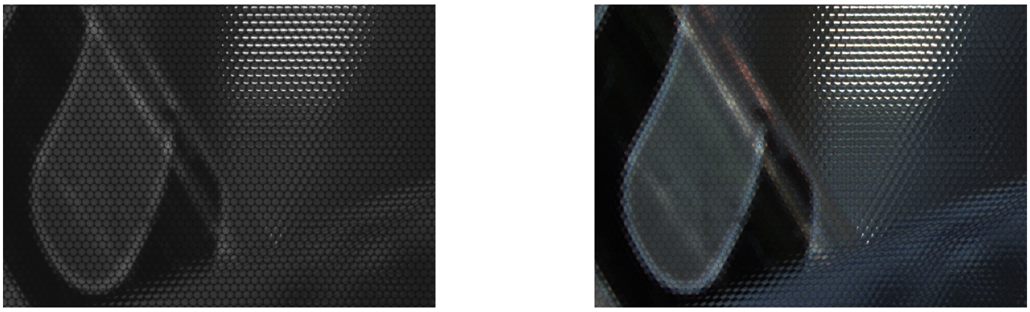


Figure 2.4: Raw lenslet processing chain.

The processing steps of this reference workflow can be described as follows:

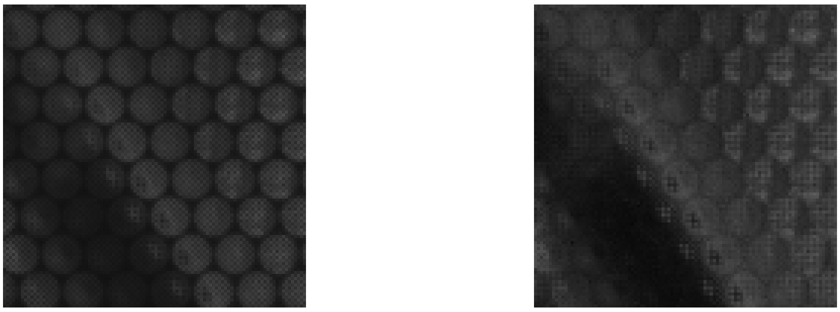
- **Demosaicing**: converts the colour samples, resulting from the camera colour filter array, to a full colour image, *e.g.*, an RGB image, as depicted in Figure 2.5.



(a) (b)

Figure 2.5: Detail of a LF: (a) before the demosaicing (b) after the demosaicing.

- **Devignetting**: compensates the reduction of image brightness towards its corners due to the lens optics, as shown in Figure 2.6. The vignetting is more apparent in plenoptic cameras LF due to the characteristics of the MLA.



(a) (b)

Figure 2.6: Detail of a LF: (a) before the devignetting (b) after the devignetting.

- **4D LF Conversion**: converts the raw lenslet data into the 4D LF described in Section 2.1, as can be seen in Figure 2.7. This conversion is described in [46], and requires some metadata from the acquiring camera. The number of pixels in the resulting 4D structure is expanded.

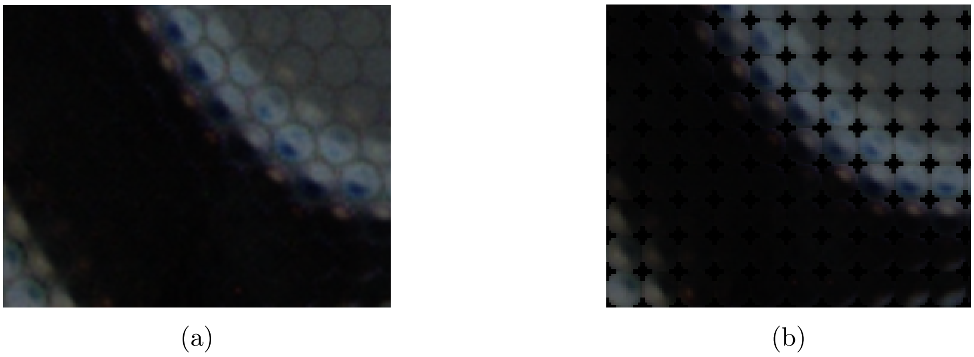


Figure 2.7: Detail of a LF: (a) before the 4D conversion (b) after the 4D conversion.

- **Colour correction**: corrects issues related to colour and illumination, making the captured LF more life like, as shown in Figure 2.8.

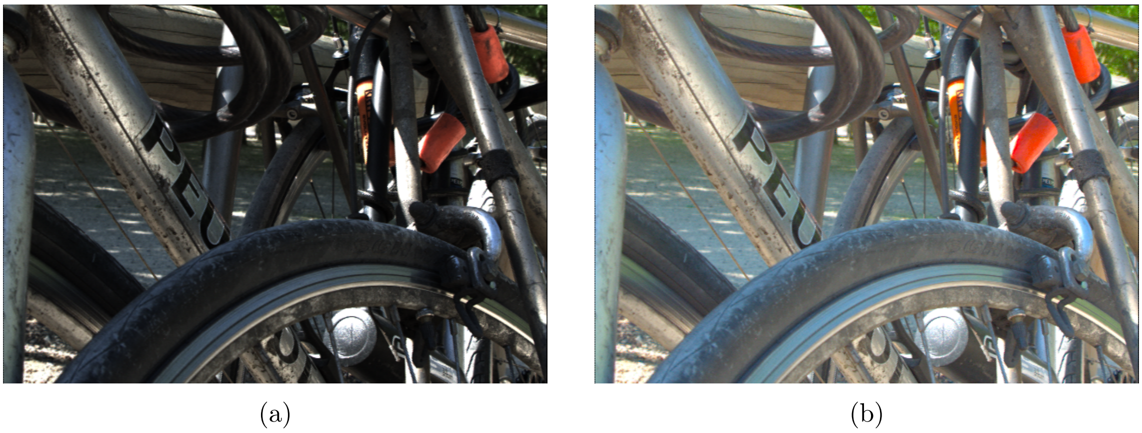


Figure 2.8: Central viewpoint of a 4D LF: (a) without colour correction (b) with colour correction.

2.3 Light Field Data Arrangements

As can be inferred from the previous sections, due to their characteristics and capturing processes, light fields present two types of correlations or redundancies. The first are the spatial redundancies, *i.e.*, between neighbouring pixels in the $v \times u$ dimensions, also seen in conventional images. The second are related to the capturing of different viewpoints by the LF, which are called angular redundancies, and can be found between neighbouring pixels in the $t \times s$ dimensions. These types or redundancies are particularly relevant when considering the possible arrangements of the 4D LF data, namely due to their impact on possible applications and coding efficiency. In this thesis three arrangements are considered: micro image, sub-aperture image, and epipolar plane image. For the purpose of this section, the LF dimensions are $T \times S \times V \times U$.

The MI, or lenslet, are the images acquired by each individual micro-lens. An example of a full LF in a lenslet representation and a zoomed-in detail is shown in Figure 2.9. In this case, each MI has 15×15 pixels, and the lenticular structure of the MLA is clearly discernible. Additionally, due to the 4D reconstruction operation described in Section 2.2.3, each MI has black pixels in its corners. These are pixels that were computed with low reliability, as they are not present in the original raw lenslet image[2]. Mathematically, a given MI can be defined by fixing (v, u) in Equation 2.2, resulting in $V \times U$ MIs each with $T \times S$ pixels, and is denoted as:

$$\mathcal{L}^{v,u}(t, s) : \mathbb{R}^2 \longmapsto \mathbb{R}^3, \tag{2.3}$$

where the mapping to $\mathbb{R}^3$, in this and the following equations, means that the equation returns the RGB colour component values for the set coordinates.

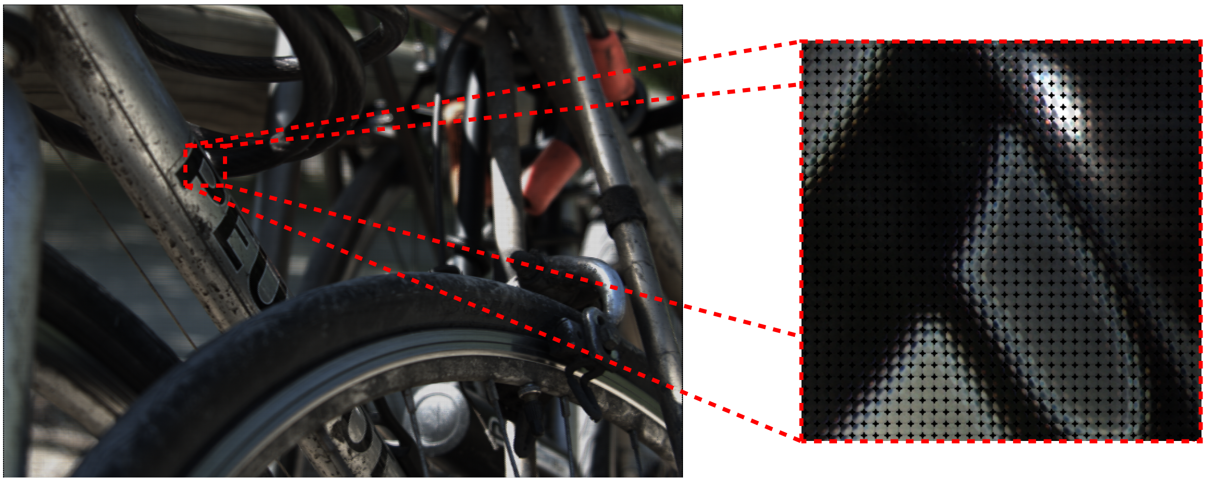


Figure 2.9: Example of the lenslet LF representation of the Bikes image from [43].

The most common representation format of a 4D LF is the SAI, also referred to as viewpoint. Each SAI represents the acquired scene observed from a particular perspective. An example of an SAI, and the full representation of the 4D LF in an SAI array format, can be seen in Figure 2.10. Figure 2.10b has darker SAIs in the corners, which are called low reliability black frames (LRBFs), due to the existence of low reliability pixels in the MI. Mathematically, a given SAI can be defined by maintaining (t, s) fixed in Equation 2.2, resulting in $T \times S$ SAIs each with $V \times U$ pixels, and is denoted as:

$$\mathcal{L}^{t,s}(v, u) : \mathbb{R}^2 \longmapsto \mathbb{R}^3. \tag{2.4}$$

Finally, the EPI representation is formed through the intersection of a specific pair (t, v) or (s, u) in a specific row or column of SAIs in the LF array. This can be interpreted as the concatenation of rows or columns taken from each SAI in the selected row or column of the LF array. Figure 2.11 shows an example of an EPI and a representation of how these images are formed. As shown in Figure 2.11c, these images contain a set of slanted lines. This information is very useful, as the slope of these lines is inversely proportional to the depth of the corresponding objects [48]. For instance, a vertical line indicates that an object is located at an 'infinite' distance from the camera, and therefore such object has no disparity between the SAIs. Mathematically, a given

[2] Low reliability pixels only exist on LFs acquired with a plenoptic camera.

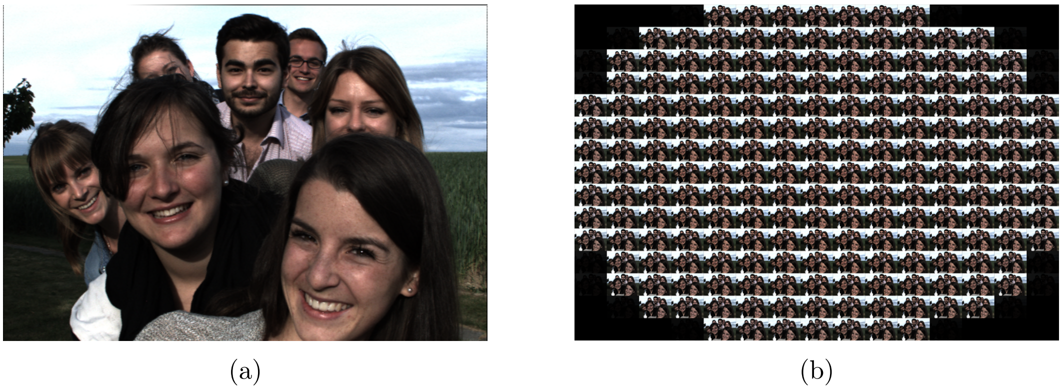


Figure 2.10: Example of: (a) the SAI representation, and (b) the full array of SAIs using the Friends image from [43].

EPI can be defined by maintaining (t, v) or (s, u) fixed in Equation 2.2, resulting in either $T \times V$ or $S \times U$ EPIs each with $S \times U$ or $T \times V$ pixels, respectively, and is denoted as:

$$\mathcal{L}^{t,v}(s,u) : \mathbb{R}^2 \longmapsto \mathbb{R}^3 \quad \text{or} \quad \mathcal{L}^{s,u}(t,v) : \mathbb{R}^2 \longmapsto \mathbb{R}^3. \tag{2.5}$$

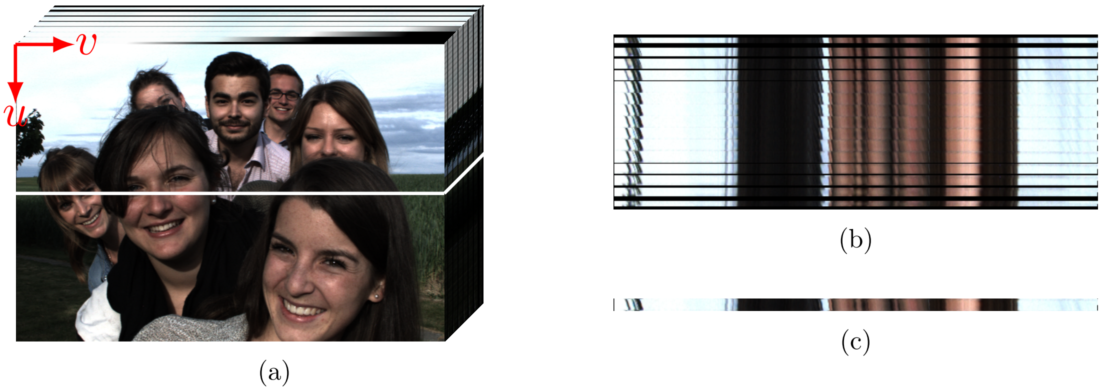


Figure 2.11: Example of the extraction of EPI using the Friends LF from [43]: (a) intersection of the SAIs stack, (b) resulting stack of EPIs, and (c) EPI resulting from the angular position $t = 8$.

2.4 Experimental Test Conditions

The Common Test Conditions of the JPEG-Pleno [49] set the basis for light field coding experiments. These conditions are essentially targeted at lossy compression algorithms evaluation. This section describes the test conditions that are followed in the experiments described in this thesis, unless stated otherwise. These are based on those defined in Common Test Conditions of the JPEG-Pleno version 3.3 with a few modifications to make them compatible with lossless coding. The experimental conditions specify: the datasets to be used and the performance metrics to be computed.

The JPEG-Pleno CTC require encoders be compared to a set of anchors obtained by encoding the LF sub-aperture images in a serpentine scan order with the High Efficiency Video Coding (HEVC) standard [50]. Quality metrics, like PSNR and Bjøntegaard delta (BD), are computed in the 4D LF format, after conversion to the YC_bC_r colour space. As this thesis deals with lossless compression, the proposed algorithms performance will be measured only in terms of bitrate.

2.4.1 Test Materials

Various LF datasets are available for researchers. The JPEG-Pleno CTC specifies five datasets with different characteristics, both from plenoptic cameras and HDCA, to be used on its core experiments. Two of these datasets are used in this thesis, selected due to their diverse content, one from the École Polytechnique Fédérale de Lausanne (EPFL) [43] and another from the Heidelberg Collaboratory for Image Processing (HCI) [51]. Further information about the datasets and their selection criteria can be found in [49].

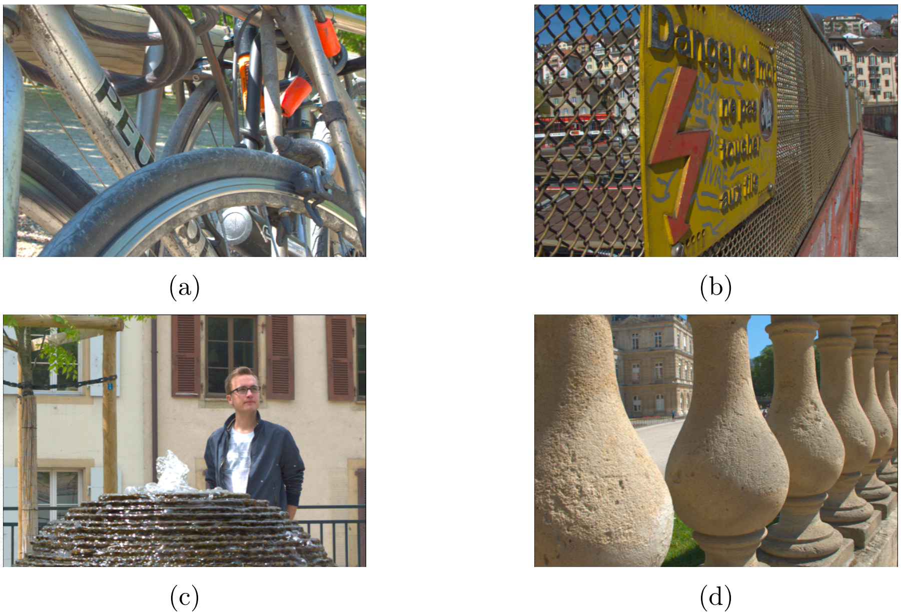
The EPFL dataset [43] is composed of natural and outdoor scenes and it was acquired with a Lytro Illum B01 unfocused plenoptic camera. As explained in Section 2.2.3, the raw lenslet LF is converted through the use of the LF toolbox. For the JPEG-Pleno CTC test set, four images were selected, of which the central SAI are shown if Figure 2.12. Due to the LRBF, out of 15×15 viewpoints only the inner 13×13 SAIs are used. This dataset characteristics are:

- Content: natural, outdoors.
- Angular resolution: 13×13 SAIs.
- Spatial resolution: 625×434 pixels.
- Bit depth: 10 bits.

As described before, the JPEG-Pleno processing chain uses the LF Toolbox [46] to the convert the raw lenslet files from the EPFL dataset and convert them to 4D LFs. The *Colour Correction* step of the pre-processing chain expands the colour range, as it can be seen in Figure 2.8. Consequently, this expansion is detrimental to the coding efficiency. Therefore, as the compression process is lossless, this step is omitted before encoding and, if necessary, can be performed at the decoder side.

The HCI dataset [51] is a computer generated HDCA dataset representing common objects. For the JPEG-Pleno CTC test set, two images were selected. The central SAI of each of those LFs are shown in Figure 2.13. This dataset characteristics are:

- Content: synthetic generated common objects.
- Angular resolution: 9×9 SAIs.

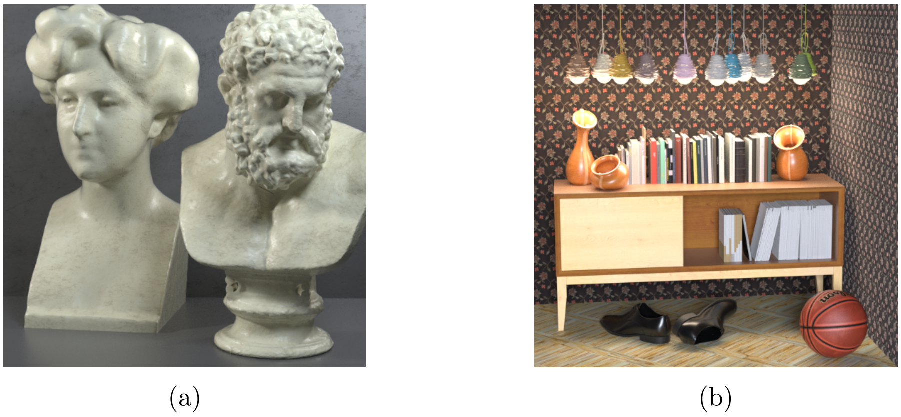


(a) (b)

(c) (d)

Figure 2.12: Central SAI from the JPEG-Pleno CTC selected LFs of the EPFL dataset [43]: (a) Bikes, (b) Danger de Mort, (c) Fountain & Vincent 2, and (d) Stone Pillars Outside.

- Spatial resolution: 512 × 512 pixels.
- Bit depth: 8 bits (provided as 10 bit by JPEG-Pleno).



(a) (b)

Figure 2.13: Central SAI from the JPEG-Pleno CTC selected LFs of the HCI dataset [51]: (a) Greek, and (b) Sideboard.

To complement the previously described datasets, a selection of medical LF plenoptic images, the light field image dataset of skin lesions (SKINL2)[3], was also included in this work [27]. It is composed of skin lesion images acquired in a medical context, and allows testing the proposed techniques with medical images, one of the main target applications of this thesis. Figure 2.14 shows the central SAI for each of the three selected LFs used in this work. This dataset characteristics are:

- Content: skin lesions images.

[3] Available at `http://on.ipleiria.pt/plenoisla`.

- Angular resolution: 9 × 9 SAIs.
- Spatial resolution: 1920 × 1080 pixels.
- Bit depth: 10 bits.

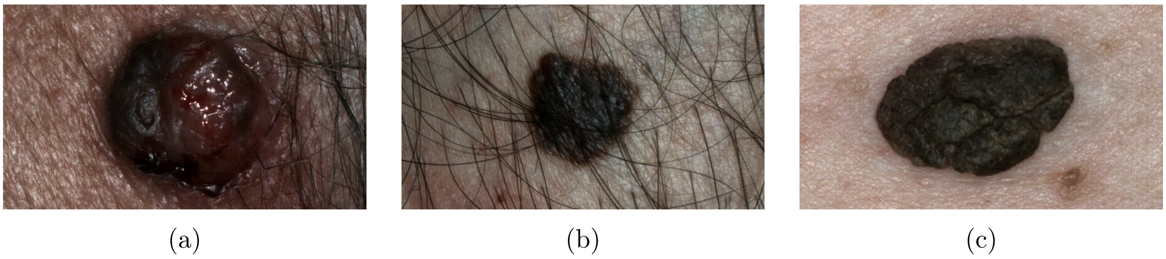


Figure 2.14: Central SAI from the selected LFs of the SKINL2 dataset [27]: (a) Img 1, (b) Img 2, and (c) Img 3.

Table 2.1 shows a summary of the test materials characteristics, including the acquisition method, resolution, bit depth, and number of LFs.

Table 2.1: Characteristics of the datasets used in this work.

Dataset	Type	Resolution	Bit depth	LFs
EPFL	Lenslet: acquired with a Lytro Illum B01 Camera	$13 \times 13 \times 625 \times 434$	10	4
HCI	HDCA: computer generated	$9 \times 9 \times 512 \times 512$	10	2
SKINL2	Lenslet: acquired with a Raytrix R42 camera	$9 \times 9 \times 1920 \times 1080$	10	3

2.4.2 Coding Performance Metrics

The quality metrics specified in the JPEG-Pleno CTC are essentially focused on lossy compression, *i.e.*, the peak signal to noise ratio (PSNR) and the BD [52] metrics. However, since this thesis is focused on lossless compression theses metrics are not needed to evaluate the performance of the proposed methods for which the only relevant criterion is the compression rate, or bitrate.

In order to measure the bitrate, the Common Test Conditions of the JPEG-Pleno use the ratio between the number of bits necessary to represent the (encoded) image and the number of pixels in the whole LF, *i.e.*, bits-per-pixel (bpp), defined as:

$$bpp = \frac{B_{C_1} + B_{C_2} + B_{C_3}}{T \times S \times V \times U}, \qquad [bits/pixel] \tag{2.6}$$

where B_{C_i} is the number of bits of the compressed colour component C_i (*e.g.*, YUV) of the image, $T \times S$ represents the number of SAIs, and $V \times U$ the number of pixels of a SAI. In parallel with the bpp the compression ratio will be used to evaluate the performance of the proposed methods, this metric is defined as the ratio between the uncompressed and the compressed file sizes:

$$\mathrm{CR} = \frac{\text{Uncompressed file size}}{\text{Compressed file size}}. \tag{2.7}$$

From the previous equation it can be easily inferred that lower bpp values results in a higher compression ratios. In the following chapters this metric is not calculated explicitly, but is often mentioned when comparing different methods.

Another relevant metric is the random access penalty (RAP), that measures the cost of the random access capabilities of a codec. The random access penalty is defined by the JPEG-Pleno CTC as the ratio between the number of encoded bits required to access a region of interest – sub-aperture images in this work – and the total amount of encoded bits. This figure of merit is defined by:

$$\mathrm{RAP} = \frac{\#\text{ bits required to decode one SAI}}{\#\text{ bits required to decode the full LF}}. \tag{2.8}$$

The RAP is reported for the region of interest that presents the largest value. In encoders without random access capabilities this metric results in RAP = 1. Otherwise, when only a part of the LF needs to be decoded it results in $0 < RAP < 1$. In general, the compression efficiency is penalised by coding options designed to obtain a low value for RAP. Therefore, RAP must be evaluated along with compression efficiency because there is usually a tradeoff between these two performance metrics.

2.4.3 Encoders Configuration

As previously stated, the JPEG-Pleno CTC use the HEVC [50, 53, 54] encoder as an anchor for the evaluation of proposed encoders. However, HEVC was not developed specifically for lossless compression, thus a set of lossless encoders are included for broader evaluation of the proposed methods. The encoders used in this thesis can be divided in two categories, intra and inter encoders. JPEG 2000 [55], Lossless and Near-lossless Compression of Continuous-tone Still Images (JPEG-LS) [56, 57], and Context based Adaptive Lossless Image Codec (CALIC) [58] are intra encoders, meaning that their compression algorithms encode each image individually. HEVC (including its range extension [59]), Versatile Video Coding (VVC) [60,61], and Minimum Rate Predictors (MRP) [62] are intra / inter encoders, meaning that their compression algorithms can also exploit the similarities between frames within a sequence. The performances of the proposed encoding methods are compared with these state-of-the-art encoders[4]. The state-of-the-art encoders implementations and choice of parameters are shown in Table 2.2. Only the main parameters are presented, everything else was left with the default values and settings.

[4] Not all of these encoders will be used for all the experiments.

Table 2.2: Configurations of the state-of-the-art encoders used in this work.

Encoder	Software	Configuration Parameters
JPEG 2000	JPEG 2000 reference software [63]	• 6 resolutions • 64 × 64 code-blocks • Reversible DWT 5-3 • Multi-component transform disabled (-mct 0)
JPEG-LS	JPEG-LS reference software [64]	• Standard configuration (-ls 1) • YCbCr colour transformation bypass active (-c)
CALIC	Martin Briano's implementation [65]	• Lossless mode (-d=0) • 2D CALIC (-m 1)
HEVC	HM reference software v16.12, including the Format Range Extension [66]	• Main RExt profile (Profile=main-RExt) • Lossless cost mode • QP 0 (zero) • Transform quantization bypass (TransquantBypassEnable=1) • 32 intra period (IntraPeriod=32) • 16 GOP size (GOPSize=16) • CRA intra random access point (open GOP) (DecodingRefreshType=1) • TZ Search motion estimation (FastSearch=1)
VVC	VTM reference software v10.0 [67]	• Using configuration file: Random Access GOP32 • Using lossless configuration files • QP 0 (zero)
MRP	Author's software from [62]	• Maximum number of classes dependent on image size • Bidirectional prediction with GOPsize = 8 (-B -G 7) • Histogram packing and variable block size prediction enabled • Reference pixels: defined for each experiment

CHAPTER 3

Light Field Lossless Coding: a Review

Contents

This chapter presents relevant background and a review on coding techniques related with the research work carried out in this thesis. After a brief introduction on the basic principles of lossless coding in Section 3.1, the JPEG-Pleno framework for light field (LF) coding is presented in Section 3.2. In particular, the two algorithms selected for the JPEG-Pleno standard are discussed. Section 3.3 presents a thorough review of lossless LF coding algorithms described in the literature. The chapter is concluded with a detailed description of the Minimum Rate

Predictors (MRP) [62, 68] lossless compression algorithm, which is the basis of most of the contributions of this thesis.

3.1 Lossless Coding Basics

The compression of a digital signal aims to obtain a compact representation of said signal using a smaller amount of bits than its original representation. The foundations of signal compression, *i.e.*, source coding, were laid by Claude Shannon in "A Mathematical Theory of Communication" [69, 70], where the author devised the fundamental definitions of information theory.

Lossless compression, as implied by its designation, is defined as a coding/decoding process where no information is lost. This type of coding is commonly used in applications that cannot tolerate differences between the original data and its reconstruction. Lossless coding is commonly used for data that are expensive to collect, that require considerable processing to analyse, or that might involve legal disputes due to inaccurate representations and should, therefore, be stored with maximum accuracy. As previously introduced, medical imaging is an example of an application with such accuracy requirements due to diagnostic certainty constraints. The information theory, set by Shannon, defines any stochastic process as an information source, producing a sequence of symbols from a finite alphabet. The amount of information carried by each symbol was mathematically defined, based on the probability of its occurrence. The information theory uses the statistical properties of the symbols produced by an information source, which might be correlated between themselves, to establish the channel capacity required to transmit the source information to a receiver [69]. In other words, how many bits are required to represent the source information. This is achieved through proper encoding of such information, which in this context is called source coding. The aim of source coding is to find out a compact binary representation of the source information that is close to the entropy of the source, *i.e.*, the average amount of information conveyed per symbol.

Although several types of source coding approaches might be used, in lossless compression generally three steps are used, as illustrated in Figure 3.1. Most encoding methods start by performing some sort of prediction, to remove the redundancy between symbols. Then, the resulting prediction error is modelled to determine the source statistics, that are used in the entropy coding step. These three steps are further detailed in the following sections.

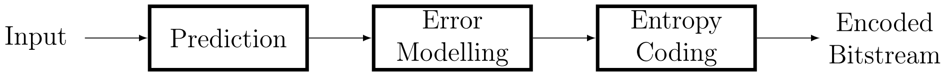


Figure 3.1: Lossless coding generic diagram.

3.1.1 Entropy Coding

In information theory a discrete data source is one whose symbols are extracted from a known discrete alphabet with a finite set of symbols [71]: $\mathcal{A} = \{a_1, a_2, \dots, a_K\}$, *e.g.*, alphanumeric characters. In a generic source like this, there might be some sort of dependency between consecutive extracted symbols. In such case, the entropy of the source $\mathcal{S}$, which measures its average information level, is defined as [72]:

$$H(\mathcal{S}) = \lim_{n \to \infty} \frac{1}{n} G_n, \tag{3.1}$$

where

$$G_n = -\sum_{i_1=1}^{K} \cdots \sum_{i_n=1}^{K} P(X_1 = i_1, \dots, X_n = i_n) \cdot \log_2 \left(P(X_1 = i_1, \dots, X_n = i_n) \right), \tag{3.2}$$

and $\{X_1, \dots, X_n\}$ is a sequence of length n from the source $\mathcal{S}$. If the symbols generated from the source $\mathcal{S}$ are independent and identically distributed, then $\mathcal{S}$ is generally called a memoryless source. The symbols generated by a memoryless source can be modelled as a random variable X that takes values in $\mathcal{A}$ with probabilities $P(a_1)$ through $P(a_K)$. Thus the entropy, in bits per symbols, of such a memoryless source is defined as:

$$H(X) = -\sum_{k=1}^{K} P(a_k) \cdot \log_2(P(a_k)). \qquad [bits/symbol] \tag{3.3}$$

The entropy is dependent only on the probability distribution of the source alphabet, not on the actual symbols. The entropy is bounded such that $0 \leq H(X) \leq \log_2(K)$, thus it is non-negative and upper-bounded by the base-2 logarithm of the alphabet size. The upper bound is reached only when the all symbols of the alphabet are equally probable. The entropy is a theoretical limit for the minimum amount of bits necessary to represent the source information without loss.

The amount of information conveyed by a symbol from such a memoryless source is inversely proportional to the probability of its occurrence. The information of a given symbol a_k, in bits, generated from a memoryless data source is given by:

$$I(a_k) = -\log_2(P(a_k)). \qquad [bits] \tag{3.4}$$

Several techniques have been developed to approach the compression entropy limit, generally referred to as entropy coding. Common entropy coding techniques include: Huffman codes [73], arithmetic encoding [74], Golomb codes [75], and dictionary coding [76]. These types of coding, essentially assign lower length codes to the more frequently occurring source symbols, unlike fixed length codes, which use the same number of bits to encode each symbol.

3.1.2 Predictive Coding

In the case of sources with memory, such as the case of natural language text where the letters forming the words or sentences are not completely random, prediction can be used to remove the symbols dependency, thus approximating a memoryless source. Signal processing operations designed to remove the memory from the source are, generically, reversible. Usually, the prediction of an arbitrary symbol x_n is obtained by a linear combination of previously generated symbols, *i.e.*,

$$\hat{x}_n = \sum_j a_j \cdot x_{n-j}, \tag{3.5}$$

where $\hat{x}_n$ is the prediction of the n-th symbol, x_{n-j} are causal symbols, and a_j are the coefficients of the linear prediction. Prediction aims to exploit redundancies present in the message to be encoded by estimating a given symbol, generated by the source, from previously observed symbols from the same source. In general, such estimate is not perfect, thus the prediction error is defined by:

$$e_n = x_n - \hat{x}_n, \tag{3.6}$$

where x_n represents the n-th symbol to encode, $\hat{x}_n$ its prediction and e_n the corresponding prediction error. Ideally, after the prediction, the resulting error should be perfectly decorrelated to allow optimal entropy coding. The coefficients of the linear predictor are commonly chosen to minimise a mean square error (MSE) criteria [71]:

$$E\left[(x_n - \hat{x}_n)^2\right] = E\left[e_n^2\right]. \tag{3.7}$$

3.1.3 Error Modelling

When prediction is used, except for a few special cases, the resulting prediction error distribution is unknown. The performance of an entropy encoder is optimal when the distribution of the prediction errors if fully known, because in this case an optimal entropy code can be designed for that specific distribution. Therefore, modelling the residuals statistics is of utmost importance to allow the design of efficient encoders. Therefore, in practical implementations, most codecs use a combination of prediction, modelling, and entropy coding.

When the source is the result of a well known physical process, a physical model might be used. However, in many cases physical modelling of the process is too complex, or even impossible to obtain. In such cases, the model might be obtained either through certain assumptions or based on empirical estimation of the prediction error statistics. If the model is obtained with recourse to an assumption approach the prediction errors are modelled by a given probability distribution. When the model is based on an empirical observation approach the statistics can be inferred as each symbol is encoded, leading to adaptive entropy coding procedures.

3.2 JPEG-Pleno Light Field Coding

The JPEG-Pleno standardization initiative has led to the development of two encoders, namely the Multidimensional Light Field Encoder (MuLE) [77], which is based on a 4D-discrete cosine transform (DCT), and the Hierarchical Warping, Merging, and Sparse Prediction (WaSP) [78] that exploits the inter sub-aperture image (SAI) redundancy. The former is better suited to encode LFs acquired with lenslet cameras, while the latter achieves higher efficiency for high density camera array (HDCA) light fields. The following sections provide a more detailed overview of the algorithms behind these encoders.

3.2.1 Multidimensional Light Field Encoder

The Multidimensional Light Field Encoder (MuLE) [77] was designed to exploit the 4D redundancy of LFs by applying a 4D-DCT to 4D blocks of pixels. The bitplanes of the resulting 4D coefficients are partitioned using hexadeca-trees, and the ensuing information is encoded through the use of an adaptive arithmetic encoder. The three main operations of MuLE, as shown in the block diagram of Figure 3.2, can be described as follows:

1. **4D Transform:** the 4D-DCT used in MuLE is a separable transform that is applied to each dimension of the LF individually. This allows to exploit the redundancy in each dimension, generating coefficients which energy is concentrated around the DC component in the 4D frequency space. This process is analogous to that used for 2D DCT. In order to efficiently represent the zero and non-zero coefficients, a bitplane clustering is used.

2. **Hexadeca-tree Bitplane Clustering:** the bitplanes of 4D-DCT coefficients, resulting from the first block in Figure 3.2, are clustered through a hexadeca-tree bitplane coder. As each block has four dimensions, the segmentation results in 16 leaf blocks, *i.e.*, sub-blocks, resulting in a hexadeca-tree, similar to what occurs in a bi-dimensional quadtree. The tree partition allows to encode blocks where all coefficient magnitudes are below a certain threshold, with a single bit, ‘0’. Otherwise, the partition is signalled with a ‘1’ and the 16 resulting blocks have half the size of the original block in all dimensions. The partition continues until all the relevant coefficients are processed.

3. **Entropy Coding:** after the conclusion of the hexadeca-tree clustering, MuLE uses a context-based binary adaptive arithmetic encoder. Different contexts are used for the resulting information: a binary context for the hexadeca-tree segmentation flags; a non-binary context for the DC coefficients of the 4D-DCT; and a non-binary context for the AC coefficients for bitplane of the 4D-DCT.

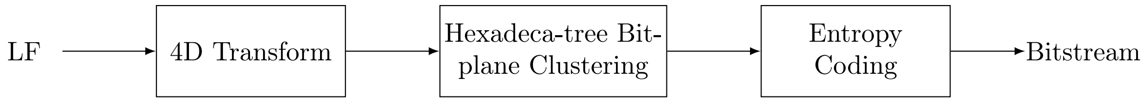


Figure 3.2: MuLE block diagram.

3.2.2 Hierarchical Warping, Merging, and Sparse Prediction

The Hierarchical Warping, Merging, and Sparse Prediction (WaSP) LF encoder [78] exploits the redundancy between the SAIs in the LF, by using depth information from the scene to warp reference SAIs to the position of the SAI to encode. WaSP sorts the SAIs into hierarchical layers, which grants the encoder some random access capabilities. WaSP was designed with the intent of being compatible with the JPEG family encoders, by using JPEG 2000 as the underlying intra-codec. The WaSP coding process can be divided in three steps:

1. **Encode SAIs at the lowest hierarchy layer:** the encoder starts by compressing the LF SAIs belonging to the lowest hierarchy layer with JPEG 2000. The WaSP encoder structure allows to replace the JPEG 2000 by any other encoder, as in [79], however within the JPEG-Pleno framework, only JPEG 2000 is used.

2. **Encode depth maps:** in the second step, the depth maps of N SAIs are encoded with JPEG 2000. This information is used to synthesise the depth maps and associated SAIs that will be used as references to encode the remaining SAIs.

3. **Encode remaining information using WaSP:** in the third step, all the remaining SAIs, that were not previously encoded in the first step, are encoded in a hierarchical fashion with the WaSP algorithms. In order to encode an SAI in a hierarchical layer H_L, the algorithm requires the depth maps of SAIs in previous hierarchical levels, $1, ..., H_L - 1$ as inputs. This module has four main operations:

   i. Warping: this step starts by warping the reference SAIs depth and colour information to the position of the SAI to be encoded. This operation generates N synthesised depth maps and SAIs for the (t, s) position of the SAI to be encoded.

   ii. Merging: the warped references depth maps and SAIs are merged using the camera centres. For the merging of the SAIs, a set of least-squares coefficients are used, which can be fixed for low bitrates or adaptive for the optimal case, where the coefficients are calculated taking into account the ground truth and need to be explicitly transmitted. In the case of the fixed coefficients, these are derived from the distance of the current SAI to the SAIs that were warped. As several viewpoints are used in the merger, the occlusions will likely be uncovered in the resulting merged SAI. The remaining occluded pixels are estimated through the use of a post-processing median filter. In the case of the depths merger, no ground truth is available. Therefore, the encoder resorts to a merging algorithm which requires no additional parameters. If the warped depth provides multiple merging candidates for a pixel, then the final value is selected by taking the median of these candidates.

iii. Sparse prediction: the sparse predictor Optimised Orthogonal Matching Pursuit (OOMP), described in [80], uses orthogonal least squares modelling to select the regressor pixels and calculate the associated coefficients that better predict the SAI using the merged information as reference. The information related to the sparse predictor, such as the position of the regressors and the values of the coefficients corresponding to these regressors, is transmitted in the bitstream.

iv. Encoding of the SAI residuals: finally, the residuals resulting from the difference between the last operation sparse prediction and the SAI to encode are compressed using JPEG 2000.

3.3 Light Field Lossless Coding

Several articles have been published reporting on methods for LF image coding, most of them focused on lossy coding techniques. Some methods exploit the inherent correlations between the neighbouring micro images which share a significant amount of information, due to the partially overlapping field of views. For instance, in [81], a 3D-DCT is applied to a stack of MIs, to exploit their correlation. In [82], lenslet images are compressed by extending the H.264 [83] spatial prediction to take advantage of the similarity between the MIs by estimating and compensating the micro images disparities. In the case of holoscopic video, disparity compensation is combined with motion-compensated temporal prediction, as followed in [84], by using the tools of High Efficiency Video Coding (HEVC) as the core compression algorithms. Other methods exploit the correlation between sub-aperture images, using a pseudo-video approach, where such images are encoded as frames of a video sequence using standard video coding methods [85,86]. In [87], the lenslet image is decomposed into various sub-aperture images, which are assembled into a cube, in order to apply a three-dimensional wavelet transform or DCT, followed by quantisation and entropy encoding. In [88], a scheme based on Multiview Video Coding is proposed, where the sub-images are re-arranged in a multiview video and encoded with the Multiview Video Coding standard. Some of these solutions do not exploit depth information, leaving room for more efficient prediction schemes that take advantage of multidimensional coding, as done by the authors in [89, 90]. A comprehensive review of lossy compression methods on the literature is available in [91].

The field of lossless compression of light fields, however, has been less explored, leaving room for further research and new developments, as addressed in this thesis. Nevertheless, potential contributions on LF lossless coding methods, shall depart from existing methods described in the literature. Therefore, in this section, a review of the literature centred in LF lossless coding is presented, providing a critical analysis of these techniques and identifying limitations that motivate further research. In this literature review, the light field lossless compression techniques are grouped into the following sections, according to the LF representation format: MI, SAI, or epipolar plane image (EPI).

3.3.1 Micro Image Compression

This first category can be divided in two, as some works choose to encode the raw sensor data and others the MIs after the conversion to the 4D LF, as described in Section 2.2.3. In the former case, encoders compress the raw 'RGGB' Bayer image obtained from the camera sensor. However, such methods require a significant amount of camera metadata to be included in the coded data, because this information is necessary to build the 4D LF and to extract 2D images of the scene. Thus the compression efficiency might be greatly affected by this camera metadata.

Lenslet Raw Data Compression

For lenslet raw data compression, the work presented in [92], proposes to split the LF into four colour planes and then computing the horizontal and vertical displacement between the micro images, after analysing the entropy of each of these planes. Afterwards, a sequential prediction process is performed and the residue is encoded using a Lempel-Ziv-Markov chain algorithm (LZMA) [93]. The whole process is depicted in the coding architecture shown in Figure 3.3.

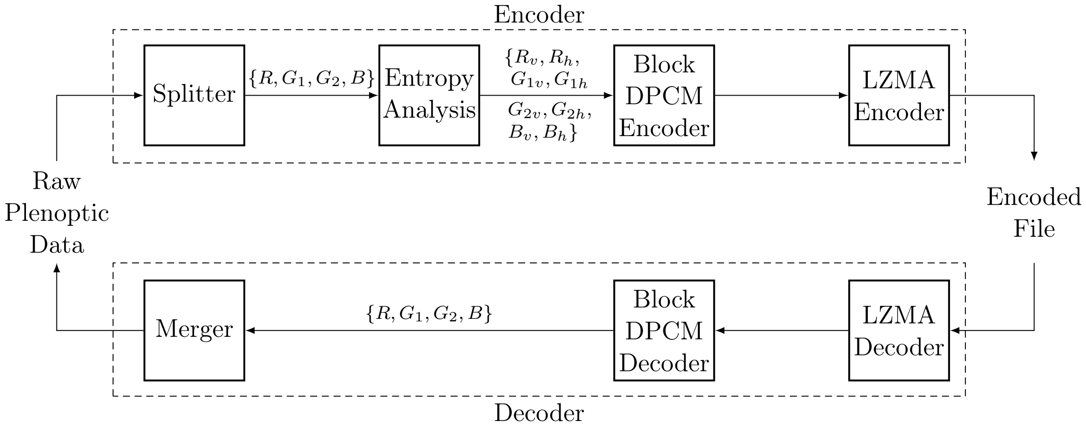


Figure 3.3: Architecture of the method proposed in [92].

Initially, the raw sensor data is separated into colour components by the *Splitter*. This module decomposes the raw data into the raw colour components, resulting in four matrices corresponding to the four colour components, as for every red and blue pixel two green pixels are captured (and considered as independent channels), due to the colour filter array. The corresponding decoder module, *Merger*, performs the inverse operation.

The resulting colour planes are independently processed by an *Entropy Analysis* module. Due to the characteristics of the plenoptic images, each colour component is made of circular micro images with very similar content, *i.e.*, pixel values, amongst themselves. This module calculates an optimal displacement, (v_d, h_d) vertical and horizontal respectively, that achieves the best matching between the pixels from a block and its replicas, *i.e.*, in the different MIs, for each colour plane.

This displacement is used to optimise a prediction process in the block differential pulse code modulation (DPCM) encoder. This DPCM encoder partitions each colour component into blocks, of (v_d, h_d) size, which are predicted using the previous horizontal block as reference. Finally, the resulting information is encoded with a LZMA encoder. Compared to JPEG 2000 and JPEG XR, this method reduces the bitrate by 56%, on average.

A different approach is proposed in [94], where the authors present a method named Sparse Modelling Compression (SMC) to predict the current MI using the nine closest causal micro images. The SMC starts by estimating the micro-lenses hexagonal grid parameters $\mathbf{\Psi}$, and encodes them into the bitstream. These parameters include the rotation angle α_m, the horizontal h_o, and vertical offset v_o of the hexagonal grid with respect to the sensors coordinate system, which are necessary to obtain the rectified LF from the raw data. Then, an image identifying the central pixels of each micro image is encoded using JPEG 2000.

In a second stage, the encoder determines which pixels do not belong to any MI after applying an octagonal mask to each lens centre. These pixels are differentially encoded by using arithmetic coding. The third stage runs through each MI in a row-wise order, and solves a regression problem for each SMC, which is formulated as the prediction of vector $\mathbf{v}_{1:192}^{(i,j)}$, *i.e.*, predicting the 192 pixels from each hexagonal MI, belonging to MI (i, j), as a linear combination of the following vectors:

1. the column vector $\mathbf{1}$ of length $K = 192$, whose elements are all ones, to compensate for a possible bias term;
2. the vector containing the elements labelled 0, ..., 191 of $\mathbf{v}^{(i,j)}$, which ensures the prediction of the k-th element of $\mathbf{v}^{(i,j)}$ by the its $(k-1)$-th element, *i.e.*, each element of the spiral is predicted by its preceding element;
3. the vectors performing the cross-prediction through the use of the nine causal neighbouring MIs.

The prediction of the microlens (i, j) elements, $\mathbf{d} = \mathbf{v}_{1:192}^{(i,j)}$, is given by:

$$\mathbf{d} = \mathbf{A}\mathbf{\Theta} + \epsilon, \tag{3.8}$$

where $\mathbf{\Theta}$ is the parameter vector, $\mathbf{A}$ is the regression matrix, and ϵ represents the prediction residuals. The regression problem is solved by finding the parameters $\mathbf{\Theta}$, with the only optimisation objective being the minimisation of the number of bits needed to encode the prediction residuals ϵ and the parameters vector. Equation 3.8 is solved as a sparse regression problem where only a fraction of the coefficients are non-zero. The problem is solved by using an approach that is similar to the minimum description length sparse prediction described in [95], by the same authors. The resulting parameters that need to be encoded are: the sparsity mask γ, its size, and the non-zero coefficient values $\boldsymbol{\theta} = \boldsymbol{\Theta}_\gamma$. The residuals are calculated using the estimated sparse prediction. Finally, the residual image is resampled into a 4D LF structure and encoded using conditional contexts, which are defined as the average of its log-absolute values. Reported results show that the proposed algorithm surpasses JPEG 2000 by up to 15.7% in terms of bitrate savings.

More recently, the same authors proposed two encoding schemes for plenoptic camera sensor images dubbed Sparse Relevant Regressors and Contexts (SRRC) [96]. The first one called SRRC-PHASE exploits the redundancies present in the micro-lens structure and in the Bayer mask pattern of the raw LF. The second scheme, SRRC-DEPTH, uses the same patch-by-patch based prediction (the left neighbour patch is used as reference), with sets of sparse predictors designed for sets of pixels situated in different depth levels of the scene, associated with classes.

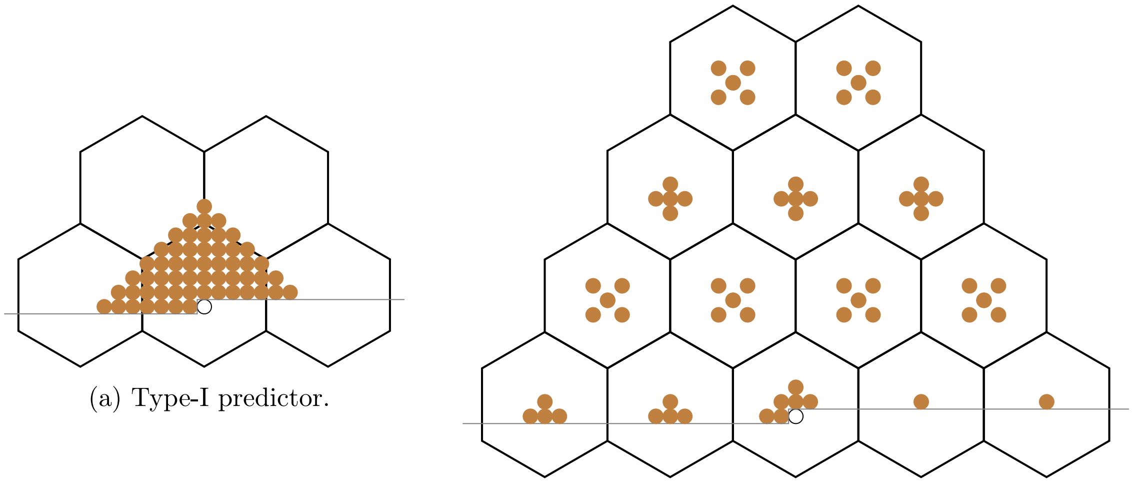
The codec splits the lenslet image in rectangular patches, later classified into the prediction classes, each roughly corresponding to a micro-lens, centred at the estimated MI centre. The same parameters $\boldsymbol{\Psi}$, described in [94], are transmitted in the bitstream, which can be used for instance to estimate the MIs centres. Due to the natural misalignment of the hexagonal lattice, some of the patches (each with 13 × 15 pixels) can overlap slightly. Both the patches and the samples within them are traversed in a raster scan order.

The encoder exploits the intra- and inter-patch correlation, induced by the Bayer pattern and the micro-lenses array (MLA), by considering that each sample in a patch can be modelled as a linear combination of causal samples within the same patch and samples in the left neighbouring patch that were already encoded. The encoder models a distinct linear sparse predictor (with a maximum of $N_S = 20$ non-zero coefficients) for each patch class and sample location, resulting in $13 \times 15 \times M$ linear predictors that need to be calculated and transmitted. The weights of the linear prediction models are computed by solving a least squares problem. The resulting information include the sparsity mask indicating the non-zero coefficients.

In the SRRC-PHASE scheme, the samples composing each ‘RGGB’ group are labelled according to their Bayer phase. The relation between the phase of a sample in the current patch $\mathbf{P^c}$ and the sample in its neighbour patch $\mathbf{P^{cn}}$, defines the selection of the linear prediction models for each sample, through the consideration of 16 pairs of phases (ϕ^c, ϕ^{cn}). For the SRRC-DEPTH scheme, the algorithm uses the depth of the central sample of each patch to attribute a class to each sample. The depth information is quantised to M intervals: $D_q(v,u) = 1 + \lceil D(v,u)(M-1) \rfloor$, and for each sample a class is selected. Classes which are lowly populated after the quantization are reassigned, to keep each class with a minimum number of elements. The classification is explicitly transmitted to the decoder. Comparison of these two schemes shows that the former produces better results in most cases.

Finally, the SRRC encoder uses the magnitudes of the sparse prediction vector elements as the indicator for the context modelling used in the arithmetic coding of the prediction residuals. SRRC manages to surpass the SMC method described earlier by approximately 0.6 bits-per-pixel (bpp), on average.

Miyazawa *et al.* use a modified version of the lossless encoder MRP to compress the 'RGGB' raw image obtained from the camera sensor [97]. The authors consider the output of the lenslet camera sensor as a two-dimensional image, and proceed to encode the samples in a raster scan order. The redundancy between the samples is removed with linear prediction, using references from the previously coded ones. Two types of linear predictors are proposed with the arrangements shown in Figure 3.4. The first linear predictor, called Type-I, exploits both spatial and spectral correlations of the Bayer pattern arrangement and might encompass references from other MIs. The second linear predictor, called Type-II, exploits inter micro images redundancies, thus the reference samples are placed in previously encoded MIs centred around the same pixel location in all reference MIs.



(a) Type-I predictor.

(b) Type-II predictor.

Figure 3.4: Arrangement of reference samples used in [97].

In order to achieve an adaptive prediction, the authors designed multiple sets of linear predictors for each LF to encode. Instead of partitioning the image in the traditional 2D rectangular blocks, this encoder uses modified Voronoi cells for the partitioning of the image in hexagonal blocks aligned to the MI structure. An extra partition is performed to minimise the impact of vignetting at the edges of the micro-lenses, by dividing the hexagonal block into seven blocks and separating the edges from a central, smaller, hexagonal block. As the blocks in the MI edges tend to have similar gradients due to vignetting, a merging process is performed within a wider hexagonal region that encompasses an area nine times larger than a single MI, so that a single linear prediction can be applied.

The linear predictors are designed individually for each of the seven types of sub-blocks, in order to provide a finer classification of the LF samples and limiting the side information needed for the adaptive prediction. The prediction of a given sample $\mathbf{p} = (x, y)$ is given by:

$$\hat{s}(\mathbf{p}) = \sum_{k=1}^{K} a_{m,c}(k) \cdot s(\mathbf{p} + \mathbf{r}_k), \tag{3.9}$$

where $a_{m,c}$ represents the set of coefficients of the optimal predictor $m \in 1, 2, ..., M$ selected for a sample $\mathbf{p}$, K is the number of samples that support the linear prediction, $s(\mathbf{p})$ represents the LF samples, and $\mathbf{r}_k$ is the relative offset of the k-th reference sample. The encoder uses context-adaptive arithmetic coding to compress the resulting prediction residuals and the associated side information.

Codecs that encode the raw lenslet LF, like the ones described in this section, have the drawback of requiring the rectification of the LF to be performed at the decoder side. This operation is essential, for instance, for extracting the viewpoints from the LF and, this way, the computational complexity of calculating the rectified LF is passed to the decoder side. Additionally, some works do not take into account the cost of compressing and transmitting the side information needed to perform the operations described in Section 2.2.3.

Rectified Micro Image Compression

In [98], Schiopu proposed to address the compression of MIs, called macro-pixels in his paper, using deep learning based prediction, namely a Macro-pixel Predictive Convolutional Neural Network (MP-CNN). The proposed MP-CNN uses six causal MIs to compute a MI prediction.

The Context based Adaptive Lossless Image Codec (CALIC) encoder [58] was used as a base to implement the MP-CNN prediction. The common CALIC pixel-wise raster scan order was changed to a MI-wise raster scan. The MIs are encoded through the following procedure:

1. the first MI is predicted using the CALIC predictor;
2. the encoder creates a MI volume V_p using up to 6 MIs on the causal neighbourhood of the current MI (M_p);
3. V_p is used in MP-CNN to compute the MI prediction $\hat{M_p}$;
4. the prediction residuals $\hat{M_p} - M_p$ are encoded in a raster scan order.

The encoding procedure proposed in this work allows to replace the MP-CNN predictor with different state-of-the-art predictors. The authors use the predictors of CALIC and Lossless and Near-lossless Compression of Continuous-tone Still Images (JPEG-LS), called LOw COmplexity LOssless COmpression for Images (LOCO-I), as a comparison to the proposed MP-CNN method. The context modelling for the residuals is also adapted to the MI structure from the CALIC context modelling. The arithmetic coder provided by CALIC is used without changes. The experiments show that the proposed method is able to surpass CALIC by approximately 20%, in terms of bitrate. However, no comparison is made with other encoders that might exploit the inherent redundancies present in the LF structure.

The same author further explores the use of deep-learning techniques to compress LF images in [99]. This work employs deep learning to both encode and synthesise the LF images. A deep-learning approach is developed to synthesise in a single step the entire array of $N \times N$ SAIs of the LF. This method employs only a small selection of reference SAIs to synthesise the entire LF. A different deep-learning based method is developed to losslessly encode the LF, expanding on the findings of the authors previous paper [98]. The architecture of this method is shown in Figure 3.5.

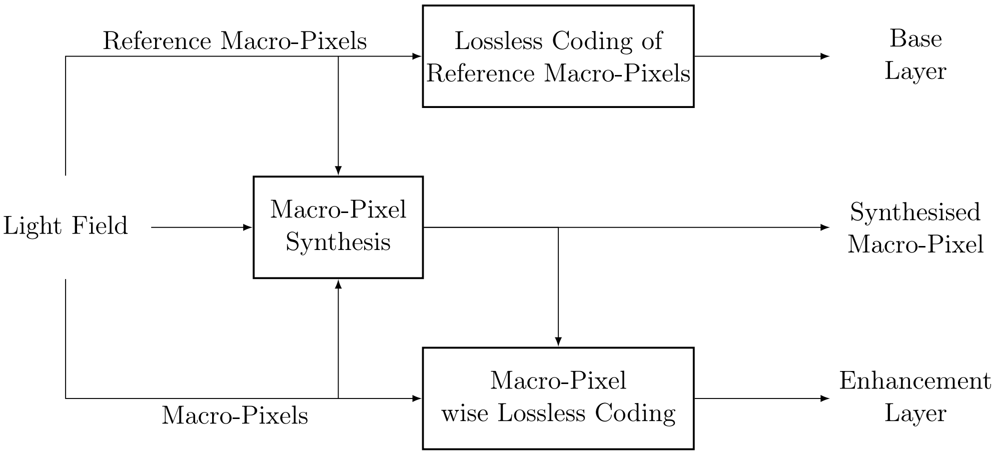


Figure 3.5: Architecture of the method proposed in [99].

The method starts by selecting the reference SAIs. The encoder uses four configurations for the reference selection, ranging from 2×2 to 5×5 SAIs, which are arranged in a symmetrical square shape. The configurations were selected to represent cases where the MI synthesis is performed based on a small number of references or by a large number of references. Then, the selected reference SAIs are losslessly encoded, resulting in the base layer, shown in Figure 3.5. This encoding procedure uses the pixel-wise Residual-Error Predictive Convolutional Neural Network (REP-CNN) [100].

The proposed deep-learning method is used to synthesise the entire LF in a single step with the selected reference SAIs. The algorithm generates the patch for synthesis based on surrounding reference MIs, and then the MPS-CNN model is applied. The employed neural-network design follows a multi-resolution feature extraction paradigm. The synthesised LF is used as a foundation for a deep-learning based method that encodes the details of the LF image. The synthesised LF provides extra information that leads to an improved MI prediction, that can use non-causal synthesised MIs. For each MI, a patch for coding is collected using six already encoded MIs and two synthesised MIs. Then the Prediction using Synthesized MPs based on Convolutional Neural Network (PSMP-CNN) model is applied to compute the MI prediction.

In a similar fashion to the method proposed in [98], in this work the authors use the CALIC codec as the basis in which the proposed predictions are used. Thus, the arithmetic coding of the prediction residuals follow a similar method to that used in CALIC, including some of the modifications used in [98]. Reported results, show that the method surpasses the one from [98] by approximately 11%, in terms of compression ratio.

3.3.2 Sub-Aperture Image Compression

This section, addresses the literature review on lossless compression of light fields represented in the SAI format. The major focus of LF coding, both lossy and lossless, has been on compressing light field data in SAIs format.

In [95], Helin *et al.* propose to use the redundancy between one SAI and its neighbours through the use of a predictive scheme, where a optimal sparse predictor is designed for each segmented region of the SAI to encode. The segmentation is based on the quantised depth map.

The SAIs are encoded sequentially starting with the central one, which is encoded with lossless JPEG 2000. The quantised depth map of the central SAI is also encoded, using the algorithm from [101], and transmitted in the bitstream. Each of the remaining SAIs is conditionally encoded based on causal SAIs. The central SAI is partitioned into K regions of constant depth based on the quantised depth map. Then the disparities are used to warp the constant depth regions of the central SAI to the next SAI. Pixels that do not belong to any region after the warping, due to occlusions, are allocated to the nearby regions.

The SAIs are processed in a spiral scanning order, as in [85]. The prediction uses the five closest causal SAIs as references. A sparse predictor is designed for each region, depending on the causal neighbourhood of the pixel to predict and also on the pixels on the same neighbourhood in causal SAIs, provided that they belong to the same region of the pixel to predict. The sparse prediction problem is formulated as:

$$\text{minimise}_\theta \quad \|\boldsymbol{y} - \boldsymbol{D\theta}\|_2 \qquad \text{s.t.} \quad \|\boldsymbol{\theta}\|_0 \leq \kappa, \tag{3.10}$$

where $\boldsymbol{y}$ is the vector of pixels to predict, $\boldsymbol{D}$ is the matrix containing the regressors for the prediction, such as the reference pixels in the reference SAIs, $\boldsymbol{\theta}$ are the coefficients that define the prediction relations, and κ is the prediction order, *i.e.*, the number of coefficients. This problem is solved by using Orthogonal Matching Pursuit (OMP), where the optimal prediction order $\hat{\kappa}$ is chosen using length optimisation. Non-relevant coefficients are set to zero and the optimal prediction order, $\hat{\kappa}$, represents the number of remaining non-zero coefficients,

The prediction coefficients are encoded with Golomb-Rice coding, and the sparsity mask is transmitted as a side information header. For the prediction residuals, context modelling and arithmetic coding are used in a similar fashion as in [102]. The authors use quantised gradients classified into 16 contexts. The information that needs to be transmitted is: the central SAI encoded with JPEG 2000, the quantised depth map of the central SAI, the disparities for each SAI, the prediction mask, the coefficients, and the residuals.

In [80], the authors propose a more refined method built upon the one in [95]. The main differences are: the segmentation is based not only on the depth information, but also on the colour information; a new sparse design method is used (OOMP)[1]; the sparsity mask and the

[1] Other algorithms are studied but this is found to produce the best results.

prediction coefficients are encoded with variable length coding, and their cost is accounted for in the minimum description length (MDL) criterion; and cross-colour prediction is added. The new sparse design method OOMP solves the problem of forward selection in least squares modelling. The main difference to OMP is the trade-off between the computation time and the efficiency of the coefficients calculation of the problem formulated in Equation 3.10.

Another variant of [95] was proposed by Tabus in [103]. This method models the scene and the lenslet array geometry, providing the necessary information to warp the SAIs into different positions. This method provides lossy-to-lossless scalable compression with random access (RA) capabilities. It starts by converting the raw lenslet image to a stack of sub-aperture images. This process is performed in the RGB image, but a colour-transform is applied before the encoding process. The pixels within a MI are marked using a template function, in a spiral order. The non-rectified SAIs are formed from pixels at the same position within each MI. This method results in different SAIs than those described in Section 2.3, as the images are formed without applying the processing chain described in Section 2.2.3. A set of selected reference SAIs are jointly encoded, *i.e.*, as a single image, with JPEG 2000, which ensures the possibility of random access to the SAIs. For low bitrates, the method can encode with JPEG 2000 only the references, while the remaining SAIs can be reconstructed predictively.

The authors use the depth estimation method from [104] to obtain a high resolution depth map estimation, which is then re-quantised to 16 levels. The quantised depth map is used to segment the SAIs into regions, following the same approach as described in [95]. This depth map is encoded with the algorithm from [101]. The algorithm proceeds to estimate the displacements of the regions from the central SAI to a side SAI in the same manner as in [95]. A difference between this and the original method is that the possible displacements are defined by the hexagonal lattice of the LF. For each region, the best displacement vector is found as the one providing the lowest MSE between the central SAI and the displaced side SAI. Pixels that are not in any region after the displacement due to occlusions, for instance, are treated in the same way as in [95]. The displacements of each SAI and region are encoded by rearranging the symbols to an image that is similar to a disparity map, which is encoded with the same algorithm as in [101].

For each region in an SAI segmentation, a sparse predictor $\mathbf{\Theta}_l$ is computed. The regressors for the sparse prediction are taken from the pixels that were already encoded on the same SAI and from eight causal SAIs of the neighbourhood. The sparse predictor is the same as the one in [95], and the side information of the predictors is encoded in the same way. For lossless compression the authors chose to transform the RGB image to a single plane 'RGGB' mosaic image. After the lossless compression of the 'RGGB' mosaic, a demosaicing process is applied, resulting in a near-lossless RGB image. The difference between the reconstruction and the original RGB LF is encoded with lossless JPEG 2000. The method, for the lossless case, is compared with JPEG 2000, showing bitrate savings of around 44%.

Schiopu *et al.* propose to exploit the similarities between SAIs using an adaptive predictive coding algorithm, called Context Modeling of Subaperture images (CMS) [105]. The algorithm predicts the current SAI, $\mathbf{I}$, at the position (t, s), defined as in Equation 2.4, using as reference a neighbouring SAI, $\mathbf{I}_{ref}$, placed at the same column or line, $(t, s \pm 1)$ or $(t \pm 1, s)$, respectively. The CMS method is represented by the diagram in Figure 3.6.

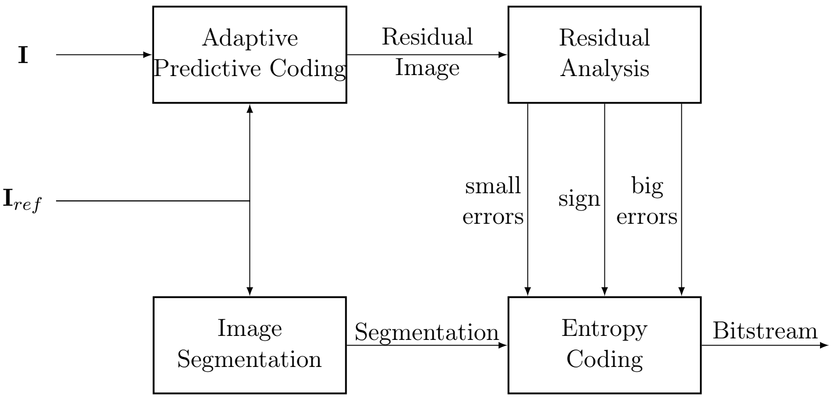


Figure 3.6: Architecture of the method proposed in [105].

The prediction is computed by applying a median filter to the prediction lists $\mathcal{P}$, such that: $\mathcal{P}_1 = \{\mathbf{I}_{ref}(v, u)\}$, $\mathcal{P}_2 = \{\mathbf{I}_{ref}(v, u), \mathbf{I}_{ref}(v, u + 1)\}$, $\mathcal{P}_3 = \{\mathbf{I}_{ref}(v, u), \mathbf{I}_{ref}(v, u - 1)\}$, and, finally, $\mathcal{P}_4 = \{\mathbf{I}_{ref}(v, u), \mathbf{I}_{ref}(v, u + 1), \mathbf{I}_{ref}(v, u - 1)\}$. The optimal predictor is selected by computing the cost of encoding the residuals using Golomb-Rice codes resulting from using each list as the predictor. The encoder uses prediction contexts, computed by using a binary mask resulting from an edge detector, for which the optimal prediction from the previously described list is found. The prediction residuals are encoded with entropy coding. The most common residual values, which have the lowest absolute values, are encoded using a context modelling algorithm. On the other hand, the largest residuals are encoded using a Golomb-Rice algorithm. This division is performed in the *Residual Analysis* block. For each residual symbol the signal is separately encoded as a '0' for positive values and as a '1' for negative values. The *Image Segmentation* block is used to generate a segmentation of the image which divides the images into regions containing pixels with similar values. The segmentation is obtained with the Simple Linear Iterative Clustering algorithm [106], where the resulting regions are used as contexts. The CMS encoder outperforms HEVC by approximately 9%, on average.

In [107], Schiopu *et al.* propose an improved method based on the previously described CMS [105]. Modifications are essentially performed to the *Adaptive Predictive Coding* and to the *Image Segmentation* blocks. The new method is denoted CMS with gradient-based detection. The authors introduce a gradient-based edge detector that is added to determine the segmentation, both in *Adaptive Predictive Coding* and in *Image Segmentation.*

The new computation of the binary edge matrix in *Adaptive Predictive Coding* uses a threshold based approach as in [105] to determine the contexts. The new gradient-based edge detector results in better prediction contexts, when compared with CMS. As in CMS, this new encoder uses segmentation to determine the contexts for the arithmetic coding. However, two types of regions are considered: *Object-based regions*, used for selecting the most important objects of the scene, and *Edge-base regions*, used for refining the contexts around the image edges. Two algorithms were tested for the segmentation, the Quantum Cut based Segmentation for Coding algorithm and the Scaled Difference Segmentation algorithm, with the latter achieving the highest coding ratio. The new algorithm achieves a compression efficiency that is 2% higher than that of CMS, on average.

Some of the works addressed in this literature review, concerning the lossless compression of LF data in the SAI format, require the use of depth and/or disparity maps for the coding process. While having the depth map on the decoder side could be an added value for the extraction of extra viewpoints of the scene, for instance, it also raises additional concerns. Two problems might arise from using the depth in the encoder. The first pertains to the methods of obtaining the depth map, given that most acquisition setups do not address this issue. Depth maps are usually computed from the actual LF, however these methods are not always reliable and could negatively affect the codec performance. The second issue is related to the need to compress the depth map along with the colour information of the LF, which can decrease the overall coding efficiency of the methods when compared to others (especially if noisy depth maps are generated by the depth estimation algorithms), as the depth map can also be estimated on the decoder side.

3.3.3 Epipolar Plane Image Compression

Finally, to the best of the author's knowledge, only two works are dedicated to exploit the information of epipolar plane images for losslessly encoding the LFs. The first one, by Mukati and Forchhammer, proposes to adapt the CALIC codec [58] to compress EPI images [108] in a new method called Context Adaptive Compression of Epipolar Plane Images (EPIC). Bearing in mind the structure of the EPIs in light fields, the authors propose to improve the prediction, error energy estimation, and context modelling of CALIC.

The usual CALIC prediction module, Gradient-Adjusted Predictor (GAP), is replaced by another that takes into consideration the EPI structure. The authors propose an EPI Slope based Predictor (ESP), in order to appropriately predict the pixel intensities along each epipolar line. The ESP estimates the slope of the edges in EPI regions, followed by a prediction of the pixels intensity through a quadratic interpolation. The slopes of each region are estimated by employing 2 × 2 horizontal and vertical edge filters. In order to improve the method robustness, the variance of each region is used to weight the average of the slopes. ESP takes advantage of the intensity of pixels remaining the same in an epipolar line to estimate the perpendicular distance of neighbouring indices. The estimated distances and the intensity of nearby pixels are used in a 1D quadratic interpolation to estimate the intensity of the current pixel along the epipolar line. A least squares problem is solved to estimate the coefficients of the quadratic interpolation that minimise:

$$\underset{\alpha,\beta,\gamma}{\arg\min} \sum_{j\in J} \left(I_j^H - \left[\alpha d_j^2 + \beta d_j + \gamma\right]\right)^2 \tag{3.11}$$

where α, β, and γ are the interpolation coefficients, I_j^H is the intensity of the pixel to be predicted at the distance $d = 0$, and γ represents the in interception of the y-axis of the quadratic curve.

EPIC builds upon the estimation of the error energy function of CALIC by adding the prediction residuals as an additional component in the error energy function. The estimator of this function is quantised into eight bins, for which the edges were optimised to minimise the entropy of the errors in a set of EPIs extracted from various LFs. Context modelling of the prediction residuals is defined by the intensities of the local neighbourhood. Further refinement of the contexts is performed so that the sensitivity to noise can be reduced. Reported results showed that EPIC is able to outperform encoders like HEVC by approximately 8%, for the École Polytechnique Fédérale de Lausanne (EPFL) dataset (resampled to a bit-depth of 8).

The previous work is further expanded in [109], increasing the compression performance by 5.3% and reducing the computational complexity of the EPIC encoder. Also, the possibility of near-lossless compression was introduced. The CALIC predictor is replaced by a new one called EPI-based Predictor (EPIP). This new predictor provides improved intensity prediction and takes into consideration the structure of the EPIs through the estimation of the epipolar lines slope.

EPIP assumes local smoothness of the disparity maps to estimate the slope of the epipolar lines. With this assumption, the gradient vector is estimated by blending the gradient vectors of four neighbouring regions. The gradient vector for each region r is calculated by:

$$\nabla_{r_i} = \sum_n F_i(n) B_r(n), \tag{3.12}$$

where $i \in \{x, y\}$, $F_i(n)$ are 2×2 filters, $B_r(n)$ represents the pixel values in the neighbouring regions, n indexes all elements in the blocks and filters, and ∇_{r_i} presents the component i of the gradient vector r. Robust estimation of the gradient vectors is achieved by removing the outliers and through a weighted sum of the inlying gradient vectors.

As in [108], this method assumes that the pixel intensity along an epipolar line remains constant. This property allows to use the previously calculated slope to predict the pixels intensity. The estimated gradient vector ∇_r is used to calculate the projected relative distance of the neighbouring pixels position. With the intensity of the neighbouring pixels and the projected distances, the prediction of the current pixel is interpolated using a quadratic curve fitting. The relationship between the projected distance and the intensity of neighbouring pixels is given by:

$$I_j = \alpha d_j^2 + \beta d_j + \gamma, \tag{3.13}$$

where I_j represents the intensity of neighbouring pixels, α, β, and γ are the coefficients for the prediction that are calculated by solving a least squares problem, and d_j is the distance from the projection origin to the neighbouring pixels positions. For robust prediction, a weighted average is taken between the prediction and an average of the intensities. Additionally, to reduce the influence of noise and of potential occlusions, the encoder uses a weighted average prediction taking into account the prediction from the vertical (taken in (t, v)) and the horizontal (taken in (s, u)) EPIs:

$$\hat{I} = \frac{\nu_V \hat{I}_H + \nu_H \hat{I}_V}{\nu_H + \nu_V}, \tag{3.14}$$

where ν_H and ν_V represent the local variations in the neighbourhood for the horizontal and the vertical EPI, respectively. The prediction residuals are encoded by using a context modelling arithmetic coding as in [108].

3.4 Minimum Rate Predictors

Most of the coding solutions developed and proposed in the scope of this thesis are based on the Minimum Rate Predictors (MRP) lossless encoder. MRP is a highly efficient lossless image codec surpassing other state-of-the-art codecs for most types of images. Therefore, this section provides an overview of the inner works of the MRP algorithm, which will be helpful to understand the techniques proposed in the following chapters.

Minimum Rate Predictors coders, like other lossless coding algorithms, try to minimise the coding rate of the prediction error [68], rather than minimising the mean square prediction error for a given rate constraint, as occurs in lossy encoders. The approach used by the MRP algorithm to achieve high coding ratios is to estimate the amount of information conveyed by prediction errors, and then design the predictors with the goal of minimising the rate. To cope with the content diversity of generic images, MRP uses an adaptive prediction scheme for each image. As for the prediction residue, the algorithm uses entropy coding based on context modelling. Such prediction residues are sorted into one of several predetermined groups, *i.e.*, contexts, depending on their characteristics. The set of residues pertaining to a given group are then represented by a generalised Gaussian probability density function [68] which parameters are estimated group by group.

Mathematically, MRP is formulated as the search for linear predictors, associated with classes, that minimise the number of bits used to encode the prediction errors and associated information. Classes are characterised by linear prediction models, each with a set of coefficients. As described in [68], the information associated with the prediction errors e in an image region R, is given by

$$I(R) = \sum_{n=0}^{N-1} \left\{ - \sum_{p_0 \in g_n} \log_2 \alpha_n + \frac{\log_2 \epsilon}{2} \cdot \sum_{p_0 \in g_n} \frac{e^2}{\sigma_n^2} \right\}, \tag{3.15}$$

where $I(R)$ indicates the estimate of the total amount of information measured in bits, N is the number of groups associated with the context modelling, g_n is the set of pixels in the n-th group, σ_n is the variance of the prediction error e of the n-th group, ϵ is Euler's number, and $\alpha_n = \frac{\Delta e}{\sqrt{2\pi\sigma_n^2}}$, with Δe representing a sufficiently small quantisation step-size of e. The objective of the encoding procedure is to minimise $I(R)$, by properly determining the sets of linear prediction coefficients for the M classes. The classes associated with linear prediction models in MRP are adapted to the characteristics of each image, through the use of a variable block size partitioning scheme in the iterative minimisation of the cost function. In general, the MRP algorithm processing flow can be represented by the block diagram shown in Figure 3.7, which is further described by Algorithm 3.1, where each colour represents the same operation in both the figure and the algorithm. The modules include three main stages: *Fixed Block-Size*, *Variable Block-Size*, and *Arithmetic Coding*. MRP uses a range coder [110] for the arithmetic compression of all generated information.

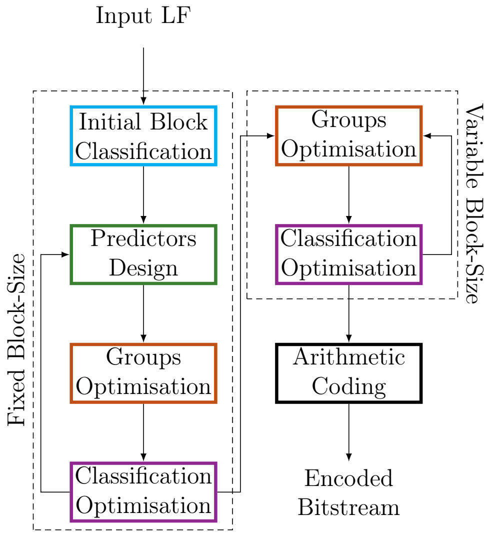


Figure 3.7: Functional diagram of the MRP algorithms.

3.4.1 Fixed Block-Size

In the *Fixed Block-Size* stage, blocks of 8×8 pixels are used to calculate the coefficients of the M linear prediction models that define the corresponding classes. First, the blocks are sorted according to their pixel variance (σ^2) and distributed by the classes in an increasing variance order to initialise the algorithm, in the *Initial Block Classification* module. Each class is associated with a linear model used for the prediction of pixel p_0 shown in Equation 3.16.

$$\hat{s}(0) = \sum_{k=1}^{K} a_m(k) \cdot s(k), \tag{3.16}$$

where $a_m(k)$, represents the prediction model weights associated to reference pixels for the m-th class associated with the pixel p_0. In Equation 3.16, $s(k)$ are the reference pixel values at position k of the prediction support region (see Figure 3.8, where p_0 represents the pixel to predict $s(0)$, and the remaining p_k represent the prediction reference pixels $s(k)$), and K is the size of the prediction support region. For each class m, the coefficients (a_m) of the corresponding linear prediction model are determined in *Predictors Design* module, by solving a set of Yule-Walker equations, using the inherent pixel correlations present in all the blocks associated with the class m.

Algorithm 3.1 MRP high level algorithm.

```
Input: image
#Initial block classification
Sort and classify $8 \times 8$ blocks by pixel variance
#1st optimisation loop using fixed block size
for 1 to MAX_ITERATIONS do
#Predictors design
    for each class do
        Compute the prediction coefficients ($a_m$)
        Calculate residuals
    Calculate prediction residue encoding cost ($B_r$)
#Groups optimisation
    for each class do
        Compute $C$ quantisation thresholds to minimise $B_r$
    Calculate $B_r$
#Classification optimisation
    for each $8 \times 8$ block do
        Move neighbour blocks classes to front of list
        Calculate residuals for all classes
        Select class that minimises $B_r$
    Calculate $B_r$
    if 10 iterations without improvement then
        end for loop
#2nd optimisation loop using variable block size
for 1 to MAX_ITERATIONS do
    Calculate prediction coefficients encoding cost ($B_a$)
#Groups optimisation
    for each class do
        Compute $C$ quantisation thresholds to minimise $B_r$
        Update shape parameter in probability models
    Calculate quantisation thresholds cost ($B_t$)
#Classification optimisation using VBS
    for Each $32 \times 32$ block do
        ClassOptimisation
    Calculate cost $J$
    if 10 iterations without improvement then
        end for loop
Remove non-utilised classes
Run arithmetic coding
Output: Encoded bitstream
```

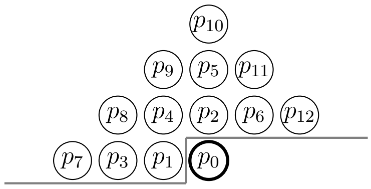


Figure 3.8: Support template used for prediction in MRP.

Algorithm 3.2 Classification optimisation procedure.

```
procedure ClassOptimisation
    Move neighbouring blocks classes to front of the list
    Calculate residuals
    Select class that minimises $B_r$
    if level > 0 then
        Calculate cost of not partitioning block ($J_1$)
        Partition block in quadtree fashion
        for each resulting block do
            ClassOptimisation
        Calculate sum of cost of partitioned blocks ($J_2$)
        if $J_2 < J_1$ then
            Partition block
    Return: Cost, partition structure and class selection
```

Then, in *Optimise Groups*, the algorithm calculates a set of thresholds that are used in the quantisation of the prediction residuals context (C). The thresholds divide the context into groups that use the same probability model – defined by the parameters of a Gaussian probability density function, for the arithmetic coding. In the optimisation of the groups context modelling, the prediction error is used to find a probability model, represented by a generalised Gaussian probability density function. The context for a pixel is determined by finding an estimate of the variance of the prediction residuals at that pixel, given by:

$$C = \sum_{k=1}^{K} \frac{1}{\delta_k} \left| s(k) - \hat{s}(k) \right| . \tag{3.17}$$

The weighting factors δ_k are proportional to the Euclidean distance between the current pixel, p_0, and the reference ones, p_k, as given by Equation 3.18, $\hat{s}(k)$, is the predicted value of the k-th reference pixel, and $s(k)$ are the reference pixel values. The parameter C is then quantised using threshold values and each pixel is classified into one of the groups corresponding to a constant variance.

$$\delta_k = \frac{\sqrt{d_x(k)^2 + d_y(k)^2}}{64}, \tag{3.18}$$

where d_y and d_x are the spatial distances, measured in pixels, between a reference pixel k and the pixel to encode.

The first optimisation loop is concluded by the *Classification Optimisation* module, which refines the initial block classification into the newly calculated linear models of the classes. In this operation, the algorithm starts by sorting a lookup table of the classes in order to move the classes of neighbouring blocks to the start of the table. By sorting the classes table and transmitting the associated index, the algorithm favours classes of neighbouring blocks and allows for a more efficient coding of the selected class index, as the resulting indices will generally have small values. Then, the prediction residuals of each block are calculated for all the classes prediction coefficients, a_m, to find the class that minimises the prediction residuals encoding cost B_r, calculated by:

$$B_r = \sum_{p_0} L\left(e|\hat{s}(0), n\right), \tag{3.19}$$

where $L(e|\hat{s}(0), n)$ represents the encoding cost of the prediction error, which is calculated by using the probability density estimate of the prediction errors belonging to the n-th group, assumed to follow a Gaussian distribution. The first optimisation loop ends when the algorithm reaches the maximum number of iterations, or 10 consecutive iterations without improvement.

3.4.2 Variable Block-Size

The second optimisation loop refines the operations of the previous loop by using variable block size (VBS), and fitting the probability models. The *Variable Block-Size* takes into account the cost in bits of encoding both the side information and the prediction errors:

$$J = B_a + B_m + B_t + B_r, \tag{3.20}$$

where B_a, B_m, and B_t are the encoding costs of the prediction coefficients, class selection, and context modelling threshold values, respectively. For every iteration where J is calculated, all its sub costs B are recalculated.

In the previous loop, at the *Groups Optimisation* step, only the threshold for the quantisation of C was optimised. Contrarily, in this phase, the algorithm also updates the shape parameter of the generalised Gaussian probability density functions, which are used to model the prediction residuals of each group. Then, in the *Classification Optimisation* step, the MRP divides the image in blocks that are partitioned in a quadtree fashion, selecting for each block the class which linear predictions model was designed in the first loop that minimises the encoding cost. The quadtree partition stage is indicated by the level value associated with a given block. The initial block size in the VBS optimization is 32 × 32 pixels (indicated by level equal to four), which can be partitioned down to blocks with 2 × 2 pixels (indicated by level equal to 0). This optimization aims at minimising both side information, represented by B_m in Equation 3.20, and the prediction error cost. B_m can be further represented by:

$$B_m = B_{m_{\text{flags}}} + B_{m_{\text{class}}}, \tag{3.21}$$

where $B_{m_{\text{flags}}}$, given by Equation 3.22, represents the cost of selecting the partition flags and $B_{m_{\text{class}}}$ the cost of selecting a class for each resulting block.

$$B_{m_{\text{flags}}} = \begin{cases} -\log_2(P_{\text{ctx}}), & \text{not partitioned} \\ -\log_2(1 - P_{\text{ctx}}), & \text{partitioned} \end{cases}. \tag{3.22}$$

In Equation 3.22, P_{ctx} represents an approximation of the probability of the blocks in a given context not being partitioned, given by:

$$
\begin{aligned}
P_{\text{ctx}} =& \underset{p}{\operatorname{argmin}} \left\{ -\log_2(p) \cdot \#ctx_0 \ -\log_2(1-p) \cdot \#ctx_1 \right\}, \\
\text{s.t.} & \qquad p \in \{0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95\},
\end{aligned}
\tag{3.23}
$$

where $\#ctx_0$ and $\#ctx_1$ represent the number of occurrences of the 0 and 1 partition flags, respectively, in each context, *i.e.*, the number of times each block is either not-partitioned or partitioned in each given context. Variable p represents the probability that minimises the cost of selecting a flag in each context. This probability is selected from a discrete set to minimise the cost of transmitting the real probability value, therefore only a single index needs to be sent to the decoder. The quantified probability calculation of Equation 3.23 follows that used in MRP, which was first introduced with the variable block size in [68].

Finally, the context calculation is determined at each level by the number of neighbouring blocks which were partitioned at said level:

$$
\mathrm{C}_{\text{qtflags}}(level, b) = \sum_{n_b \in N(level,b)} \text{qtflags}(n_b), \tag{3.24}
$$

where b represents a block, qtflags the partition flag for block n_b with the size given by *level* (0 and 1 for non-partitioned and partitioned blocks, respectively), and n_b are the neighbour blocks belonging to the set $N(level, b)$ of neighbouring blocks partitioned at the same *level* as b. As MRP uses 4 levels for the block sizes and in the quadtree three neighbours are being considered, there are a total of 16 different contexts.

The block sizes define the quadtree levels ranging from 4 to 0. For each quadtree level, the decision to partition a block depends on whether or not the sum of the sub-blocks costs (J_2) is smaller than the cost of the parent non-partitioned block (J_1). The encoding of the index of the selected class is achieved by the same process described in the previous optimisation loop.

3.5 Minimum Rate Predictors Video

Over the years some solutions based on MRP appeared in the literature. More recently, the author proposed a MRP video codec [62], which employs a bidirectional prediction scheme where the reference frames are selected in a similar fashion to that of HEVC. The proposed bi-directional prediction uses two frames, chronologically before and after the frame to encode, as the support for the prediction, as shown in Figure 3.9. The new bi-directional prediction is given by:

$$
\hat{s}(0) = \sum_{k=1}^{K_c} a_m(c,k) \cdot s(c,k) + \sum_{k=1}^{K_b} a_m(b,k) \cdot s(b,k) + \sum_{k=1}^{K_f} a_m(f,k) \cdot s(f,k), \tag{3.25}
$$

where $a_m(i,k)$, with $i \in \{c, b, f\}$, represent the prediction model weights associated to pixels in the current and reference frames (forward and backward), respectively, for the m-th class. In Equation 3.25, $s(i,k)$, with $i \in \{c,b,f\}$, are the pixel values in the current and reference frames at position k of the prediction support region (see Figure 3.9), and K_c, K_b, and K_f are the sizes of the prediction support regions. In MRP video, the prediction support templates were defined to be centred at the position of the pixel to be predicted, indicated by the circles in Figure 3.9. In the same way, the context for a pixel was updated to reflect the changes in the prediction, as seen in Equation 3.26.

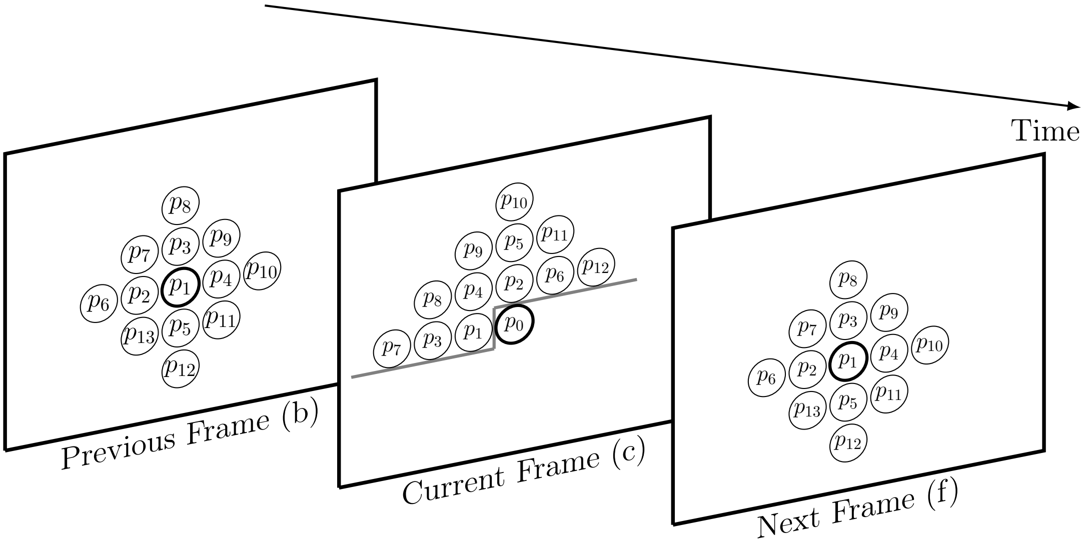


Figure 3.9: Support templates used for intra and bidirectional prediction.

$$C = \sum_{k=1}^{K_c} \frac{1}{\delta_k} \left| s(c,k) - \hat{s}(c,k) \right| + \sum_{k=1}^{K_b} \frac{1}{\delta_k} \left| s(b,k) - \hat{s}(b,k) \right| + \sum_{k=1}^{K_f} \frac{1}{\delta_k} \left| s(f,k) - \hat{s}(f,k) \right|. \tag{3.26}$$

The weighting factors δ_k are proportional to the Euclidean distance between the current pixel, p_0, and the reference ones, p_k, as given by Equation 3.27 The parameter C is then quantised using threshold values and each pixel is classified into one of the groups corresponding to a constant variance.

$$\delta_{j,k} = \frac{\sqrt{d_x(k)^2 + d_y(k)^2 + \lambda^2}}{64}, \tag{3.27}$$

where d_y and d_x are the spatial distances, measured in pixels, between a reference pixel k and the pixel to encode, λ represents the distance between frames and is set to 1 for the reference frames, and set to 0 for the current frame.

3.5.1 Histogram Packing

Prior to the actual encoding, a pre-processing step is used to compact the input image/video histogram. This technique, which was introduced to MRP video in [62], improves the performance of lossless prediction based encoders, provided that it produces an image with lower total variation (TV) than the original. The literature shows that prediction and transform based encoders have a poor performance when compressing images with sparse histograms [111]. Essentially, it is proven that when an image histogram is sparse, meaning that not all the possible amplitude values are used, and the used values are not contiguous, encoders perform poorly. This problem was alleviated in MRP video by using histogram packing prior to encoding.

The histogram packing method used in MRP video operates as follows: first, the number of different amplitude values in the images is determined to compute the dynamic range of the histogram packed sequence; then, the original values are mapped onto the new amplitude range, through the use of a lookup table, preserving the ordering of the amplitudes. The image maximum value parameter is changed for the new active maximum value present after the histogram packing is performed. This parameter affects several structures in MRP such as the arithmetic coder, and has an impact in the compression efficiency.

The information needed to revert the histogram packing is transmitted by encoding the conversion table through the run length encoding (RLE) method described in [112]. As the MRP encoder compresses video sequences the histogram is calculated for the input sequence at once, for simplicity. This ensures that the reference images have pixels in the same range as the image to encode. The encoder uses the TV to decide whether the histogram packing should be used. If this operation produces an image with lower TV than the original, this means that the approximation error is reduced, thus increasing the compression efficiency.

3.5.2 Coded Stream Syntax

The syntax of MRP video bitstreams includes a sequence header, followed by the information at frame level, which consists in the number of classes, quadtree division and classification, predictors coefficients, and prediction error. This is organised as shown in Figure 3.10, representing the sequence of the syntax elements: sequence header (SH), number of classes (NC), class partition (CP), class coefficients (CC) and threshold values (ThV), followed by the prediction residuals (PR). The shaded blocks, in light grey and grey, represent the side information at sequence and frame level, respectively.

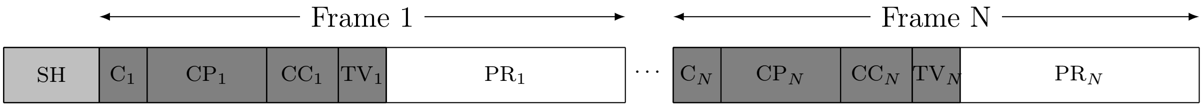


Figure 3.10: MRP stream frame-based structure.

3.6 Summary

This chapter discussed basic aspects of image coding, presenting the underlying principles of current lossless LF compression approaches based upon the information theory, followed by a review of the most relevant works on LFs lossless compression. The literature review, presented in Section 3.3, provided a comprehensive overview on LF lossless coding, allowing to better assess the motivation and research advances of this thesis beyond the current state of the art.

The aforementioned review discussed diverse works addressing LF coding methods capable of providing an overall high compression efficiency. However, none of these present a comprehensive encoding solution that is able to achieve both a high compression efficiency, and also provide other capabilities in the bitstream, such as RA and scalability. This thesis addresses this gap by proposing, amongst other contributions, a new encoder that provides support for these enhanced capabilities to the bitstream, allowing to access specific SAIs or areas of the LF. Another gap that was found in this review is related to the colour space in which the images are encoded. In fact, most, if not all, the previously quoted works use the common RGB colour space, which is not the most efficient for compression [113] due to the redundancies between colour channels. Another example that is worthwhile to mention is the lack of enhanced capabilities for flexible navigation in the encoded LF.

The chapter is concluded with a thorough description of the MRP algorithm used for lossless image coding. In previous works [62] it has been shown to achieve state-of-the-art performance on the lossless compression of biomedical images. Therefore, the MRP was selected as the base for most of the contributions described in the following chapters.

CHAPTER 4

Compatible Light Field Image Coding

Contents

In order to face the new image compression challenges unveiled by the light field (LF) technology, this research started by investigating methods compatible with existing codecs. In this context, compatible compression means to use a generic state-of-the-art codec to encode the 4D LF data, re-arranged in a format that is compatible with the encoder. It allows the use of existing encoders to compress new types of data, therefore minimising the effort needed for early adoption of LF imaging. Often, pre-processing methods are used to increase the compression efficiency of LFs, a topic also addressed in this chapter. Such approaches may be based on algorithms that either reduce the redundancy of the LFs or conform the LF data format to the requirements of each specific encoder.

The present chapter is organised as follows, Section 4.1 presents various data arrangement formats for LFs that are compatible with state-of-the-art encoders [J3], Section 4.2 presents a study on the impact of reversible colour transforms on the coding efficiency [J4, C5], and Section 4.3 proposes a pre-processing disparity compensation algorithm for LFs [C1]. Section 4.4 presents an experimental evaluation of the proposed methods, and Section 4.5 concludes the chapter by summarising its main contributions.

4.1 Light Field Data Arrangements

In general, there are several factors that impact the performance of a video or image encoder, like input content, data quality and representation, coding architecture and algorithm, as well as system parameters.

The performance of coding algorithms is greatly dependent on the data format that is used for the representation of LF visual content, therefore the structure arrangement of the input images can be a key factor to boost the compression performance. In this section, three LF data arrangements are used that are compatible with generic image and video codecs, providing different formats for LF conforming compression. The data arrangements are: lenslet array, pseudo-video sequence (PVS) (stack of sub-aperture images), and epipolar plane image (EPI) stack arrangements.

A **lenslet array** is an image composed of a two-dimensional array of micro images (MIs). In this arrangement, a pattern corresponding to the micro-lens array of the optical system is clearly discernible, as shown in Figure 4.1. In the case of Lytro Illum camera, each of these micro images has 15 × 15 pixels, where the black pixels at the corners of each MI correspond to values computed with low reliability, as explained in Chapter 2. Since they may be used in LF reconstruction, they must also be losslessly encoded. In general, compression of lenslet array images use intra-frame coding algorithms, exploiting the spatial redundancy within the 2D set of data as a whole.

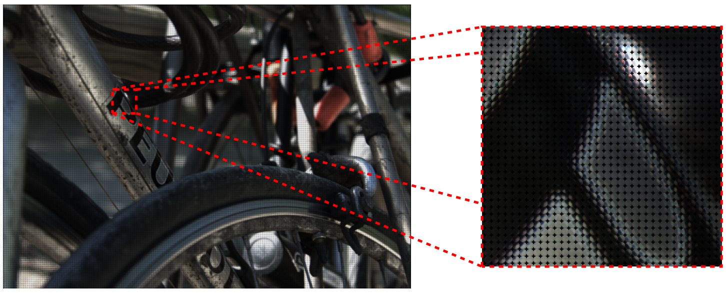


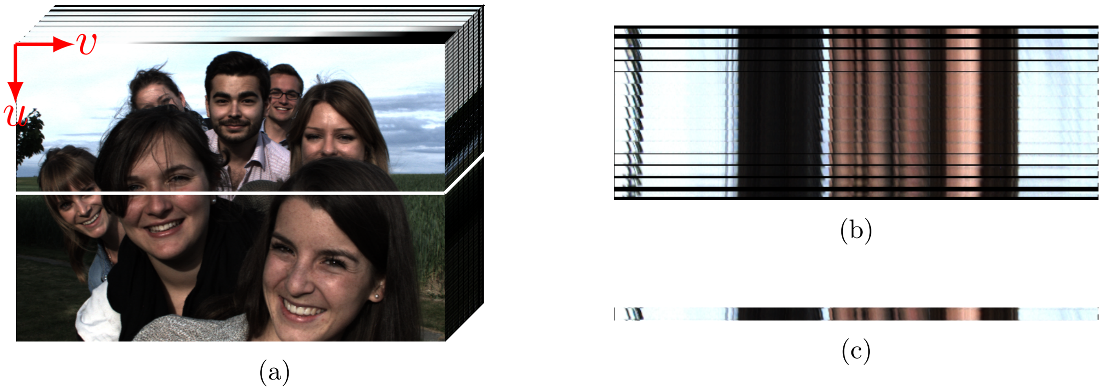
Figure 4.1: Example of the lenslet LF representation of the Bikes image from [43].

A common format that is used to represent LF data is the so-called **pseudo-video sequence**, which consists on a stack of SAIs, where the individual SAIs are re-interpreted as video frames. As mentioned in Chapter 2, the spatial position of each pixel in an MI represents the direction of a light ray. Thus, an alternative data format reinterprets the multiple sub-aperture images, derived from the LF micro images, as a short pseudo-video sequence (PVS), where the SAIs assume the role of video frames. In spite of the existing redundancy between adjacent LF micro images, higher compression performance is expected when pseudo-video is used instead of lenslet images. This consideration is due to the video encoders being able to make use of some of the angular redundancies described in Section 2.3. The black pixels in the corner of each micro-image generate black sub-aperture images, which are referred to as low reliability black frames (LRBFs) and have different properties than the remaining SAIs.

The compression performance of LF PVS using standard lossy video encoders was addressed in [85], using the state-of-the-art High Efficiency Video Coding (HEVC) encoder with two different arrangement schemes for the stack of SAIs, namely Raster and Spiral scan. However, to the author's best knowledge, the impact of such different data formats for lossless encoding has never been investigated prior to this thesis. Additionally, due to its inherent characteristics, the LRBF images may be properly exploited to improve the compression efficiency. Thus, this work investigates the performance of lossless coding using various schemes to generate PVS extracted from the stack of SAIs, namely a new scheme named Spiral-blackend (SBE).

The representations of the three scan orders used in this work are depicted in Figure 4.2. As can be inferred from Figure 4.2a and 4.2b, in the frame sequence resulting from Raster and Spiral scans, the LRBF images are inserted between other sub-aperture images with significant information, which may be detrimental for inter-frame prediction. In the proposed SBE scan, the PVS is built upon the spiral order, but packing together all LRBF images at the end of the sequence, rather than including them in the middle of the sequence as before. In Figure 4.2c, the scan starts at the central SAI, in an SAI array sense, proceeding along the green spiral, in order to minimise the disparity between consecutive scanned views, thus displacing the corner frames to the end of pseudo-video sequence. The green path is followed by the yellow path (outer path) and then by the red path (external path), in a clock-wise fashion. The order of the paths is indicated by the number in blue inside the white squares.

The **EPI stack** representation uses epipolar plane images that are formed through the intersection of a specific pair (t, v) or (s, u) in the LF array, as explained in Chapter 2. The depths of the objects in the scene are closely related to the slope of the lines present in these images [48]. For instance, a vertical line indicates that an object is located at an infinite distance from the camera, and therefore such object has no disparity between the SAIs. An example of the representation of an EPI slice from the pseudo-video sequence Friends 1 is shown in Figure 4.3. This type of image presents very specific characteristics, as it comprises smooth areas and sharp edges. Therefore, it is expected that methods based on linear predictors, like Minimum Rate Predictors (MRP), perform more efficiently on predicting the sharp edges, thus achieving higher compression efficiency than other encoders.

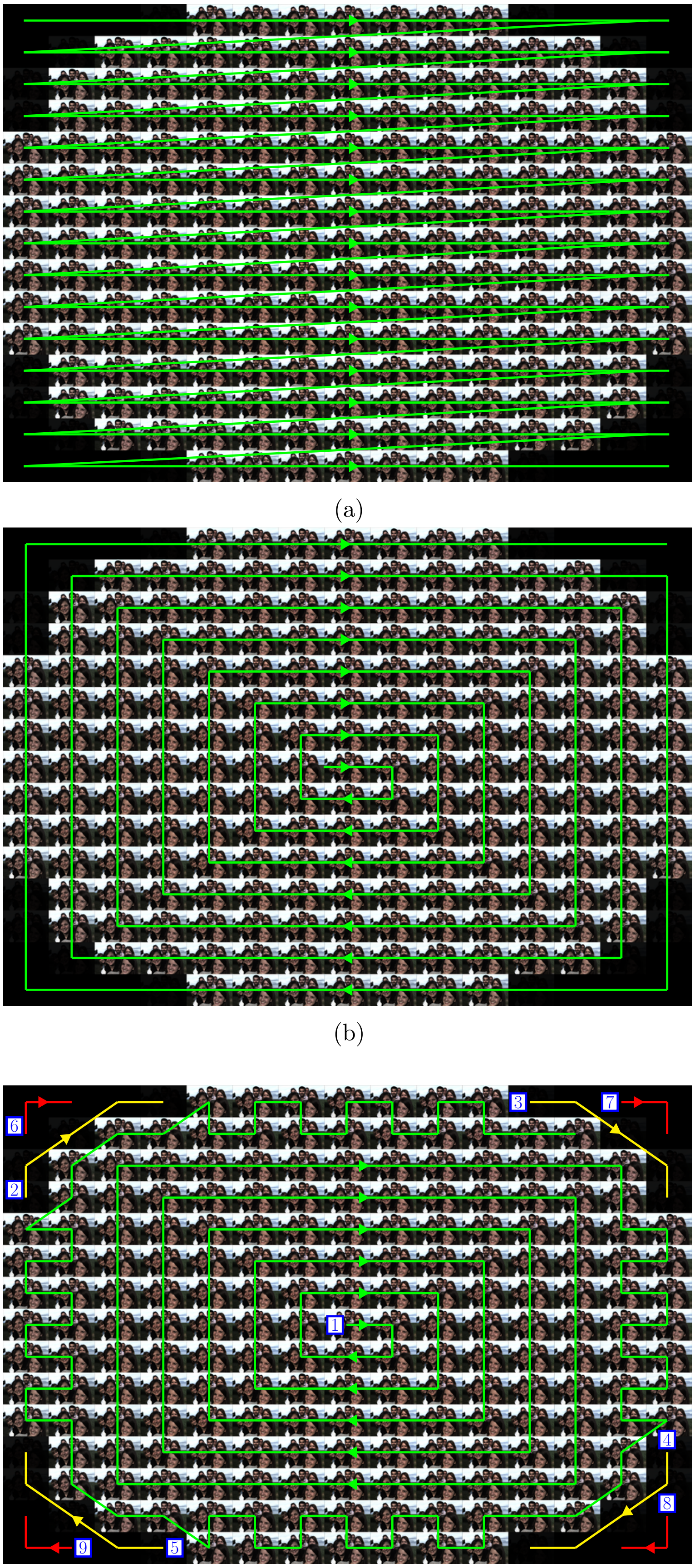


(a)

(b)

(c)

Figure 4.3: Extraction of an EPI using the Friends 1 LF from [43]: (a) intersection of the SAIs stack, (b) resulting stack of EPIs, and (c) EPI resulting from the angular position $t = 8$.



(a)

(b)

(c)

Figure 4.2: Representation of the used scans: (a): Raster, (b): Spiral and (c) Spiral-blackend.

4.2 Reversible Colour Transforms

Due to its inherent inter-component correlation, the RGB colour space is not the best choice when compressing natural images [113]. This fact has prompted the development of other colour spaces, to which the RGB images are converted prior to the encoding process. Since this work is focused on LF lossless compression, only reversible colour transforms should be considered to ensure perfect reconstruction of the original image after decoding. To the best of the author's knowledge, the use of these alternative colour spaces had not been considered in the context of light field data compression prior to this thesis.

In the scope of this work 6 different colour spaces, and associate reversible transforms, were explored, namely, A2 [114], LDgDb [115], LDgEb [115], RCT [55], RDgDb [115], and YCoCg [116]. This choice is grounded on previously published research that indicate these spaces have the best performance in the context of image compression. For example, RCT and YCoCg spaces are used in the JPEG 2000 and JPEG XR standards, respectively, because of their superior performance [115]. However, since higher component decorrelation does not always translate into higher compression efficiency, it is important to extend the study to other colour spaces and associated transformations. With the exception of A2 and RDgDb, which use the original G and R components as the Y component, all the other transforms have one component that represents the luminance.

For the purpose of lossless compression these transforms are constrained to be reversible, with a potential range expansion. Other colour spaces, like YCbCr, are the result of irreversible transforms, unless a range expansion of 2 bits-per-sample is accepted [115]. Any linear transformation can be made reversible (integer-reversible) given that the determinant of the transformation matrix is equal to ±1, as this allows the use of a lifting scheme for factorisation of the transformation matrix [115]. As mentioned in [115], these lossless transforms are designed to enable higher compression ratios at the expense of small range expansions. The mathematical expressions of these reversible transforms, which only involve sums, subtractions and bit-shifts, lead to a bit-depth of the transformed image that is, at most, one bit higher than that of the original image. Despite the range expansion, due to the decorrelation performed by these transforms, the overall compression performance is improved.

Table 4.1 presents the mathematical definitions of each transform, where $C_j,\ j = 1, 2, 3$ and $\lfloor . \rfloor$ represent the resulting components and the floor operation, respectively. Table 4.2 presents the mathematical expressions that define the corresponding inverse transforms.

Table 4.1: Reversible colour transforms from RGB to target colour space (forward transform).

Target Colour Space	C_1	C_2	C_3
A2	G	$B - G$	$R - G$
LDgDb	$R - \left\lfloor \frac{C_2}{2} \right\rfloor$	$R - G$	$G - B$
LDgEb	$R - \left\lfloor \frac{C_2}{2} \right\rfloor$	$R - G$	$B - C_1$
RCT	$\left\lfloor \frac{R+2G+B}{4} \right\rfloor$	$B - G$	$R - G$
RDgDb	R	$R - G$	$G - B$
YCoCg	$B + \left\lfloor \frac{C_2}{2} \right\rfloor + \left\lfloor \frac{C_3}{2} \right\rfloor$	$R - B$	$G - B - \left\lfloor \frac{C_2}{2} \right\rfloor$

Table 4.2: Reversible colour transforms from the colour space to RGB (inverse transform).

Origin Colour space	R	G	B
A2	$C_3 - G$	C_1	$C_2 + C_1$
LDgDb	$C_1 - \left\lfloor \frac{C_2}{2} \right\rfloor$	$R - C_2$	$G - C_3$
LDgEb	$C_1 + \left\lfloor \frac{C_2}{2} \right\rfloor$	$R - C_2$	$C_3 + C_1$
RCT	$C_3 + G$	$C_1 - \left\lfloor \frac{C_2+C_3}{4} \right\rfloor$	$C_2 - G$
RDgDb	C_1	$C_1 - C_2$	$G - C_3$
YCoCg	$B + C_2$	$C_3 + C_1 - \left\lfloor \frac{C_3}{2} \right\rfloor$	$C_1 - \left\lfloor \frac{C_3}{2} \right\rfloor - \left\lfloor \frac{C_2}{2} \right\rfloor$

4.3 Disparity Compensation for Multidimensional LF Encoding

This section describes a novel disparity compensation method to enhance the matching between the various SAIs of a 4D LF. The proposed method was initially designed to be used with LF lossless encoders. However, due to the algorithm characteristics, the resulting disparity compensated LF has a larger resolution than the original one, resulting in worse compression efficiency in lossless codecs. Nevertheless, as shall be demonstrated, this method improves the performance of 4D transform-based LF lossy encoders, namely to the Multidimensional Light Field Encoder (MuLE) [77]. Figure 4.4 presents the workflow of the proposed approach, where disparity compensation operates as a pre-processing of the 4D LF in the SAI domain ($\mathcal{L}^{t,s}(v,u)$ as represented by Equation 2.4). The resulting image $\widetilde{\mathcal{L}}^{t,s}(v,u)$ is then encoded without any further changes nor modifications of the encoding algorithm. The objective of the pre-processing is to obtain a disparity-compensated LF, $\widetilde{\mathcal{L}}^{t,s}(v,u)$ given by Equation 4.1, where, ideally, there is no disparity between the resulting SAIs and consequently the prediction error is minimised. This algorithm was designed to be perfectly reversible, which means that the inverse operation restores the original LF without loss. To accommodate such requirement the number of pixels of the resulting disparity compensated LF is larger than initially, because when pixels are shifted to compensate for the disparity new regions are created at their previous positions that need to be in-painted. The new disparity compensated LF is then encoded using the 4D LF encoder MuLE. The experimental results show that, despite the increase on the input data size, the proposed approach is able to improve the compression efficiency of MuLE, in terms of Bjøntegaard delta (BD) rate reduction, for high density camera array (HDCA) LFs.

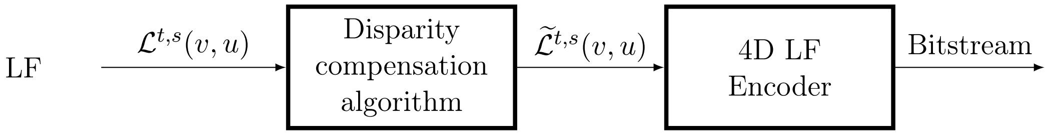

LF
$\mathcal{L}^{t,s}(v,u)$
Disparity compensation algorithm
$\widetilde{\mathcal{L}}^{t,s}(v,u)$
4D LF Encoder
Bitstream

Figure 4.4: Workflow of the proposed method.

As previously stated, the disparity compensation algorithm increases the amount of data, *i.e.*, the number of pixels, that need to be compressed, which has a negative impact on the compression efficiency of lossless encoders. However, in the case of lossy encoders, namely MuLE (described in Section 3.2.1), this algorithm is able to improve their compression efficiency. Therefore, despite not being the main focus of this thesis, the disparity compensation algorithm was evaluated for lossy compression, using the MuLE encoder, which at the time of writing this thesis does not have lossless compression capabilities. Future developments of MuLE may benefit from the pre-processing stage proposed in the remaining of this section.

Due to the LFs acquisition setup, each SAI is slightly different from its neighbours, owing to the baseline distance between the cameras or micro lenses. Therefore, each SAI in a row or column after the disparity compensation can be represented by:

$$\widetilde{\mathcal{L}}^{t,s}(v,u) = \mathcal{L}^{t,s}(v - \mathcal{D}_{\mathcal{V}}(v,u), u - \mathcal{D}_{\mathcal{H}}(v,u)), \tag{4.1}$$

where $\mathcal{D}_{\mathcal{H}}$ and $\mathcal{D}_{\mathcal{V}}$ represent the horizontal and vertical disparity between matching pixels in two neighbouring sub-aperture images, respectively. For most of the JPEG-Pleno datasets [117], it is reasonable to assume that the cameras (or micro lenses) are equally spaced, both horizontally and vertically. Under these conditions, the disparity of any two consecutive SAIs should be the same for object regions at equal distances from the camera plane, such that $\mathcal{D}_{\mathcal{H}} = \mathcal{D}_{\mathcal{V}} = \mathcal{D}$.

Prior compensation of this disparity before encoding is expected to increase the compression efficiency for 4D encoders, such as MuLE, due to the higher data correlation resulting from better pixel alignment. Considering that a scene is composed of several objects at various depths, each SAI pixel can have a disparity value different from those of its neighbours, which should be independently computed in order to obtain a better compression ratio. Since the determination of a disparity value for each pixel would entail a large increase in the amount of coded data, as well as a huge computational complexity, in this work the image is divided into 4D non-overlapping blocks and a block-level disparity $\mathcal{D}_b$ is estimated. This is achieved by finding the disparity $\mathcal{D}_b$ that minimises the entropy of the coefficients of the 4D-discrete cosine transform

(DCT) of the compensated block $\widetilde{\mathcal{L}}_b^{t,s}(v,u)$, given by,

$$\mathcal{D}_b = \underset{\mathcal{D}_b}{\arg\min}\, \mathrm{H}\left(\mathcal{Q}_p * \left\lfloor \frac{\mathrm{DCT}\left(\widetilde{\mathcal{L}}_b^{t,s}(v,u)\right)}{\mathcal{Q}_p} \right\rceil\right), \tag{4.2}$$

where $\mathrm{H}(\Omega) = -\sum_i P_i \log P_i$ represents the entropy, with $P_i = P(\Omega_i)$ the probability of the i^{th} symbol in the set Ω, $\lfloor\cdot\rceil$ is the rounding to the nearest integer operation, $\mathrm{DCT}(\cdot)$ denotes the 4D DCT of size equal to the block $\widetilde{\mathcal{L}}_b^{t,s}(v,u)$, and $\mathcal{Q}_p$ is the uniform quantisation step, which was chosen to be 16, as to provide 4 bit precision for the coefficients which is sufficient for the algorithm purposes.

The 4D disparity compensated blocks ($\widetilde{\mathcal{L}}_b^{t,s}(v,u)$) are computed by shifting each 2D SAI block (defined in (v,u)) by an amount proportional to the disparity. The shifting must also be proportionate to the angular position t or s because the computed disparity is measured between blocks on neighbouring SAIs. Therefore with increasing angular positions the shifting amount must compensate for the shifts of the blocks in lower angular positions. After the optimal disparity compensation is found, the relative positions of all SAI blocks, originally co-located in each SAI (*i.e.*, with the same (v,u) coordinates and different (t,s)), are shifted accordingly, thus minimising Equation 4.2.

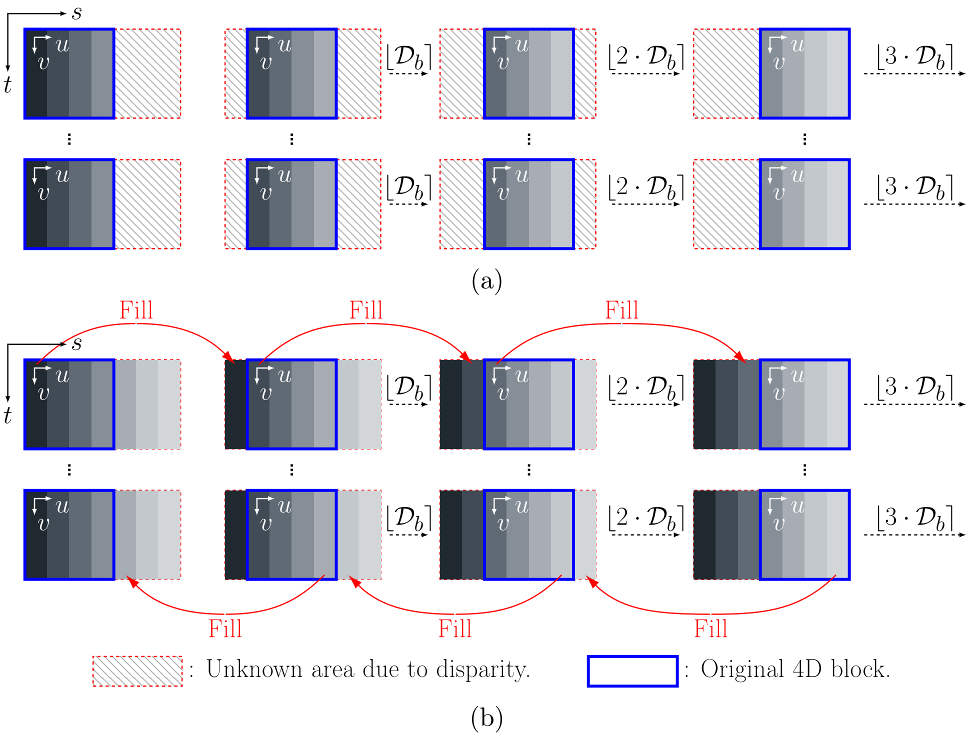
To perform the minimisation, two parameters are needed: the size of the 4D blocks B_S and the maximum allowed disparity ($\mathcal{D}_M$), both expressed in pixels. Given the B_S and $\mathcal{D}_M$ values as input, all disparity values belonging to $|\mathcal{D}| = [0, ..., \mathcal{D}_M]$ are tested for each block, in steps of 0.1 pixels. Assuming the horizontal (t) and vertical (s) disparities are the same ($\mathcal{D}_b$), due to the camera geometry, this search can be performed only along the s dimension. For each s, the respective SAI block is shifted by an amount that depends on $\mathcal{D}_b$ and the position s itself, as given by

$$\mathcal{D}_b^s = \lfloor \mathcal{D}_b \cdot (s-1) \rceil, \tag{4.3}$$

where, again, $\lfloor\cdot\rceil$ represents the rounding to the nearest integer operation. Equation 4.3 shows that the further away a view is from the reference, *i.e.*, from the SAIs with $s = 1$, the further it needs to be moved. As sub pixel interpolations are not performed to keep the method fully reversible, the previously described displacement treats cases of $\mathcal{D} < 1$, by only moving a block $\mathcal{L}_b^{t,S_c}(v,u)$ when the accumulated disparity $D_b^{S_c}$ reaches the next integer value. S_c represents the current SAI position. This process is shown in Figure 4.5a for the cases where D_b equals one.

The proposed disparity compensation adds new undetermined areas (hashed regions in Figure 4.5a) when each SAI block $\mathcal{L}_b^{t,s}(v,u)$ is shifted. Figure 4.5b shows the filling of the undetermined areas, using the neighbouring SAI information, *i.e.*, the respective number of lines or columns is copied from the co-located position in the previous SAI, so that these areas are the same in all SAIs, producing a disparity compensated image with increased 4D redundancy. For each new 4D block, Equation 4.2 provides the optimal disparity value $\mathcal{D}_b$.

The optimal disparity obtained for each 9 × 9 × 32 × 32 pixels 4D block in the Greek LF,



s
t
u
v
$\lfloor \mathcal{D}_b \rceil$
$\lfloor 2 \cdot \mathcal{D}_b \rceil$
$\lfloor 3 \cdot \mathcal{D}_b \rceil$
(a)
Fill
Fill
Fill
Fill
Fill
Fill
▨ : Unknown area due to disparity.
□ : Original 4D block.
(b)

Figure 4.5: Disparity compensation algorithm diagram: (a) new undetermined areas, (b) filling of the undetermined areas.

from [117], is shown superimposing the central SAI in Figure 4.6. It can easily be seen that regions at different depths are represented by different ranges of disparities.

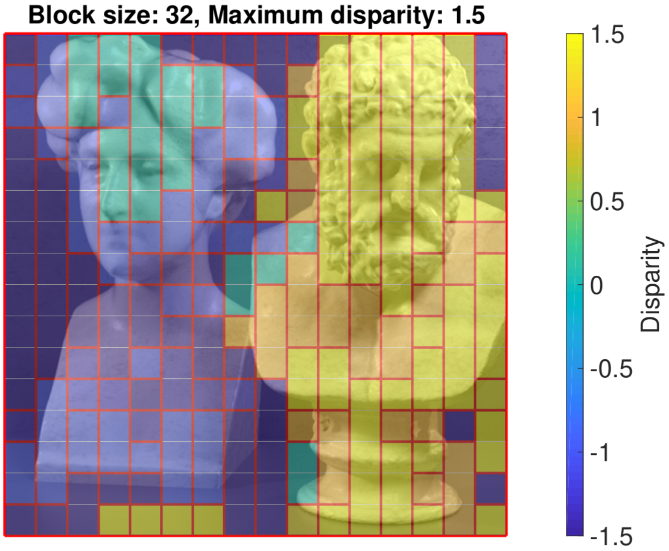


Figure 4.6: Optimal disparity for each 32 × 32 pixels block of image Greek.

As pointed out before, the requirement of a fully reversible algorithm, leads to a disparity compensated LF with higher number of pixels than the original one, as illustrated in Figure 4.5. In fact, the original 4D block in blue, with dimensions $t_b \times s_b \times v_b \times u_b$, has each of its spatial blocks, originally at the same (v, u) position, shifted amongst themselves. This operation means that the previous dimensions v_b and u_b will be multiplied by a factor of $\mathcal{D}_b \cdot (T - 1)$ (or

$\mathcal{D}_b \cdot (S-1)$, but generally T equals S), as illustrated in Figure 4.5a, where $T = 4$. The previously exposed means that the same block, after disparity compensation, has the following dimensions: $t_b \times s_b \times v_b \cdot \mathcal{D}_b \cdot (T-1) \times u_b \cdot \mathcal{D}_b \cdot (S-1)$. This procedure is responsible for the increase of pixels to encode after the disparity compensation, which are filled as shown in Figure 4.5b.

From the algorithm description, it can be inferred that, the higher the value of $\mathcal{D}_M$, the higher the potential increase in the number of pixels in the final disparity compensated image. In order to minimise the number of extra pixels in the final image, blocks of equal disparity are grouped in super blocks, either in v or u direction, which is chosen to minimise the final number of pixels. The algorithm is applied with the calculated disparity to each super block as a whole. The super block partition is represented in red in Figure 4.6 (for the v direction). Such operation reduces the amount of extra information needed to fill the undetermined areas presented in Figure 4.5. It is important to balance the trade-off between the increased inter-SAI redundancy and the number of extra information to encode. In the processed image, although the number of pixels to encode is larger, the 4D redundancy is increased. The result of this is that the encoder is able to achieve higher compression ratios.

Finally, after applying the disparity compensation to each super block, they are concatenated in order to assemble the final image. As each of these blocks has a different number of pixels, a padding operation, performed by repeating the last pixels of the block, is performed to fill the gaps in the final disparity compensated image, as can be seen in Figure 4.7. For instance, in this example the algorithm increased the size of the SAI from 512 × 512 to 704 × 607 pixels.

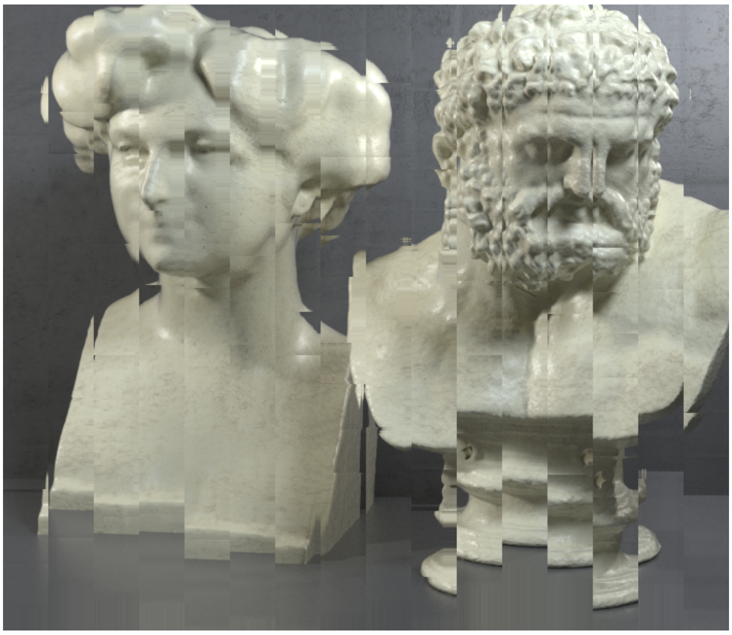


Figure 4.7: Example of a compensated SAI computed with proposed algorithm.

4.4 Experimental Evaluation

This section starts by performing an evaluation of the compression efficiency of the LF data arrangements described in Section 4.1 for various state-of-the-art encoders. These experiments use the RGB format as input for encoding all LF images. Then, the same combinations of encoders and data arrangements are used to assess the impact of the use of the different colour spaces described in Section 4.2. The results of these experiments will be used as a benchmark for the methods proposed in the following chapters. The work presented in Sections 4.1 and 4.2 was developed before the release of the Common Test Conditions of the JPEG-Pleno (JPEG-Pleno CTC) [49]. At the time, the only available information was provided by the Call for Proposals of the JPEG-Pleno (JPEG-Pleno CfP) [26], which initially set a different processing methodology for the École Polytechnique Fédérale de Lausanne (EPFL) dataset [43], that were followed in the experiments pertaining to these sections. In the JPEG-Pleno CfP, the processing chain for raw lenset images followed the pipeline explained in Section 2.2.3. However, the images were scaled to fit in 8 bit representation and the rectified LF uses the whole 15 × 15 SAIs, including all the LRBFs. Finally, only the dataset made available by EPFL was used with a total of 12 LFs (instead of 4), as defined in the 2016 IEEE International Conference on Multimedia and Expo Grand Challenge: Light-Field Image Compression [43]. This section is concluded with the experimental evaluation of the disparity compensation algorithm described in Section 4.3.

4.4.1 Light Field Data Arrangement Evaluation

To simplify the compression efficiency analysis for the different lossless encoders, the results are presented according to the used LF image representations, namely lenslet array, PVS and, EPI stack. While the lenslet array representation can only be encoded as a 2D image, SAI and EPIs are stacked to form a pseudo-video signal, enabling video encoders to exploit the correlation along the third dimension inherent to a sequence of sub-images. As demonstrated by the experimental results, some encoders have better compression performance when using specific LF image formats. For example, encoders like HEVC and a modified version of MRP [62], which are able to exploit correlation in a third dimension akin to the temporal dimension in video are more adequate to compress the pseudo-video image stacks. In these experiments, the number of reference pixels in MRP were configured as follows:

- Lenslet array: 72 pixels.
- PVS: 56 on the intra frame, 20 pixels on the current frame and 5 pixels on the reference frame for unidirectional prediction frames, and 12 pixels on the current frame and 13 pixels on the reference frames for bidirectional prediction frames.
- EPI stack: 30 pixels in the intra frame, 20 pixels on the current frame and 5 pixels on the reference frame for unidirectional prediction frames, and 12 pixels on the current frame

and 13 pixels on the reference frames for bidirectional prediction frames.

As mentioned in Section 3.4, MRP is a single-component encoder, without support for multi-component images. Therefore, two schemes were used to encode the three components of the RGB LF images: (i) encoding each RGB component individually and taking the sum of the three to calculate the bitrate (ii) concatenating the RGB components, side-by-side horizontally, prior to encoding into a single bitstream.

Coding of Lenslet Arrays

Table 4.3 lists the compression ratios achieved on each LF in the lenslet array arrangement, as well as the average rate per encoder, all expressed in bits-per-pixel (bpp). In the following tables the best result for each row is highlighted in bold.

Table 4.3: Coding performance for lenslet array format (bpp).

Sequences	JPEG 2000	JPEG-LS	CALIC	HEVC	MRP
Ankylosaurus	14.24	**10.76**	12.45	11.18	11.26
Bikes	13.12	10.46	11.71	**9.91**	10.71
Color	13.51	**10.75**	12.24	10.89	11.09
Danger	11.99	9.52	10.69	**9.30**	9.75
Desktop	13.32	8.35	10.68	**8.28**	9.13
Flowers	12.34	9.80	11.03	**9.41**	10.08
Fountain	13.91	11.26	12.59	**10.79**	11.58
Friends	12.62	**9.10**	11.09	9.44	9.57
ISO	14.02	10.88	12.34	**10.32**	11.22
Magnets	14.20	**10.65**	12.22	11.09	11.05
Stone	12.67	10.07	11.35	**9.91**	10.22
Vespa	11.74	9.39	10.55	**9.15**	9.60
Average	13.14	10.08	11.58	**9.96**	10.44

Results presented in Table 4.3 show that for lenslet arrays HEVC and Lossless and Near-lossless Compression of Continuous-tone Still Images (JPEG-LS) achieve the highest average coding performance among the tested algorithms, far better than the remaining ones. HEVC performs slightly better, 0.12 bpp lower, than JPEG-LS. The lowest compression efficiency is obtained with JPEG 2000.

Pseudo-Video Sequence Coding

In the second set of experiments, the LF data was compressed using a PVS arrangement and the same encoders, as shown in Table 4.4. Prior work [85] showed that the scan order that was followed to select the sub-aperture image for stacking or serializing the data into pseudo-video is an important factor that affects the resulting compression. To characterise this impact, the three scanning orders described in Section 4.1 were used to encode with HEVC, and MRP, and the results segmented accordingly. Given that JPEG 2000, JPEG-LS and Context based Adaptive Lossless Image Codec (CALIC) are intra encoders, the scanning order does not affect the compression efficiency, as each SAI is encoded on its own. These results are compared with [95], where Helin *et al.* encode SAIs in a multiview format, by taking each sub-aperture image as a different view, and exploiting the inter-view redundancy (*i.e.*, the angular redundancy).

Table 4.4: Coding performance for PVS arrangement (bpp).

Sequences	JPEG 2000	JPEG-LS	CALIC	HEVC			MRP			[95]
				Raster	Spiral	SBE	Raster	Spiral	SBE	
Ankylosaurus	9.79	9.60	9.33	7.94	7.86	7.85	**7.51**	**7.51**	7.52	8.10
Bikes	11.59	11.02	9.45	7.83	7.75	7.72	7.45	**7.41**	7.42	9.57
Color	9.62	9.29	10.44	8.06	7.94	7.94	7.30	**7.28**	7.30	9.79
Danger	11.49	10.94	9.64	7.36	7.25	7.23	7.07	**7.01**	7.02	10.59
Desktop	8.65	7.86	10.83	6.01	5.97	5.96	**5.68**	5.69	5.69	6.98
Flowers	12.12	11.49	10.92	7.46	7.43	7.41	**7.17**	7.20	7.21	10.74
Fountain	11.38	10.95	8.18	8.39	8.32	8.30	**7.81**	**7.81**	7.83	9.64
Friends	10.41	10.00	11.47	6.88	6.80	6.78	6.45	6.42	**6.41**	8.27
ISO	11.05	10.45	9.59	8.23	8.11	8.09	7.73	**7.71**	**7.71**	9.81
Magnets	9.76	9.52	10.06	7.93	7.84	7.84	**7.53**	**7.53**	7.55	8.18
Stone	12.01	11.56	11.36	7.61	7.58	7.54	**7.40**	7.44	7.41	9.24
Vespa	9.80	9.25	10.88	7.21	7.13	7.12	6.71	**6.68**	6.71	9.88
Average	10.64	10.16	10.18	7.58	7.50	7.48	7.15	**7.14**	7.15	9.23

As expected, encoders using inter-frame prediction tools, which can explore redundancy between consecutive SAIs, exhibit the best average performance: MRP achieves the best efficiency, followed by HEVC (0.34 bpp higher than MRP), both outperforming the intra encoders. From the results, it is also clear that MRP and HEVC achieve better performance when encoding the LF images as pseudo-video than in lenslet format, by 3.30 and 2.48 bpp, respectively. This gain is notably smaller for the intra-frame encoders. In this latter case, using PVS rather than lenslet reduces the coding rate by approximately the same amount magnitude, with the exception of JPEG-LS, where the reduction is only a few hundredths of bpp.

The gains obtained by HEVC and MRP, when compared with the lenslet array encoding, can possibly be associated to the efficiency of the encoding tools implemented by these two encoders, which help to exploit inter-SAI angular correlation. The first of these tools uses bi-directional prediction which, in the case at hand, most likely succeeds in overcoming the problem of predicting pixels of one SAI that are occluded in neighbouring SAIs taken from slightly different perspectives. The second feature of both encoders, which also helps improving the compression ratio, is sub-pixel accurate prediction. It is possible that, for a given baseline distance, the disparity observed in adjacent sub-aperture images has fractional values, depending on the depth of the scene objects. Therefore, being able to perform inter-SAI predictions with fractional-pixel accuracy is likely to improve the prediction quality, driving the coding rate down. Thus, it is expectable that the sub-pixel accurate motion estimation/compensation of HEVC contributes to improve the encoding performance of LF pseudo-video sequences.

The MRP algorithm, on the other hand, does not use motion compensation, or explicitly sub-pixel accurate prediction, as predictors involve pixels located at integer-grid positions. It can however achieve sub-pixel accurate prediction, without motion compensation, because the linear predictors can use fractional coefficients, thus defining prediction values which are interpolations between pixels of the prediction support region. As an example, consider a case where only two reference neighbouring pixels of prediction support region are used; if the linear prediction coefficients associated with these pixels are 0.5, for instance, then the predictor is essentially computing an interpolation of these two pixels, *i.e.*, an half-pixel position reference value. Linear prediction in MRP can thus produce a result similar to sub-pixel accuracy, with the ensuing advantages for the coding of the sequence of SAIs.

Another variable with impact on the compression ratio, and which analysis is important, is the type of scan order used to create the pseudo-video sequence from the stack of sub-aperture images. As can be seen in Table 4.4, the scanning order used has only a small effect on the coding performance. In the HEVC case, the 'SBE' method provides the most efficient results, with a difference of up to 0.1 bpp with respect to the other scan orders. For MRP, the best results are obtained with the 'Spiral' scan, but the encoding performance differences observed when using different scans are minimal 0.01 bpp at most. The 'Raster' scan order produces the worst results for both encoders, with a more pronounced negative effect in HEVC. Most likely, this poor performance is due to the fact that, unlike the other scan orders, this arrangement does not keep neighbouring SAIs close to each other in the serialised sequence, breaking the high inter-SAI correlation structure that exists in the original 2D tiling. Additionally, unlike what happens in the 'SBE' arrangement, where the similar LRBF images are packed together at the end of the PVS, slightly improving the performance of HEVC, in the case of the 'Raster' scan order the LRBF images are distributed along the pseudo-video sequence, further deteriorating the inter-frame prediction performance. It is also noticeable that for MRP the 'Spiral' and 'SBE' orders do not improve the coding results obtained with the 'Raster' order very much. Perhaps this is due to the very good adaptation mechanisms used in the MRP prediction step, which make the method very robust.

One alternative way of handling the LRBF images is to simply not encode them. This is possible because usually the structure of the micro-lenses images is known, and so the low reliability pixels can be identified and removed. This solution was also tested, where the pseudo-video sequences obtained using the Spiral scan order and truncated to ignore the LRBF images were encoded using HEVC and MRP, achieving on average coding ratios of 7.30 bpp and 6.99 bpp, respectively, which is an improvement of 0.18 bpp for HEVC and 0.16 bpp for MRP. However, not coding these images results in a compression method which is no longer lossless, inducing slight deteriorations to the performance of post-processing operations like refocusing that use these low reliability corner pixels, resulting however in imperceptible degradations on the refocused images.

The rightmost column of Table 4.4 presents results of a recent LF encoding method [95]. Despite the fact that such algorithm was specifically designed for LFs, the results show that the compression efficiency of both HEVC and MRP is consistently higher for all data arrangements (*i.e.*, lower bpp). On average the method presented in [95] requires about 29% more bpp than MRP and 23% more than HEVC to encode the same sequences.

Coding of Epipolar Plane Image Stack

The coding of LF images represented in the third format, EPI stack, and using the same encoders as before was also evaluated. These experimental results are presented in Table 4.5, showing again that MRP is the most efficient encoder for this type of arrangement, followed by the HEVC (0.54 bpp higher). However, the MRP and HEVC results obtained with the EPI representation are worse than those achieved by encoding with the same encoders in the PVS format, although better than the results achieved for the lenslet array representation format.

Table 4.5: Coding performance for the EPI format (bpp).

Sequences	JPEG 2000	JPEG-LS	CALIC	HEVC			MRP		
				Raster	Spiral	SBE	Raster	Spiral	SBE
Ankylosaurus	9.56	8.83	9.35	9.07	8.79	8.43	8.17	8.01	**7.73**
Bikes	10.05	9.50	10.24	8.91	8.68	8.33	8.19	8.11	**7.82**
Color	9.97	8.83	9.62	8.99	8.76	8.41	7.98	7.82	**7.57**
Danger	11.04	9.30	9.96	8.43	8.19	7.84	7.84	7.76	**7.54**
Desktop	9.64	6.91	7.47	6.69	6.55	6.28	6.15	6.08	**5.85**
Flowers	10.69	8.86	9.74	8.60	8.44	8.05	7.93	7.97	**7.74**
Fountain	11.33	9.82	10.42	9.44	9.32	8.97	8.52	8.55	**8.31**
Friends	10.34	8.03	8.92	7.86	7.57	7.21	7.19	7.02	**6.70**
ISO	10.94	9.40	10.05	9.21	8.96	8.60	8.40	8.29	**8.02**
Magnets	10.59	8.75	9.23	9.10	8.82	8.47	8.13	7.99	**7.70**
Stone	8.29	9.26	10.03	8.70	8.49	8.06	8.23	8.19	**7.90**
Vespa	10.65	8.32	9.03	8.23	8.00	7.67	7.28	7.21	**7.00**
Average	10.26	8.82	9.51	8.60	8.38	8.03	7.83	7.75	**7.49**

In comparison with the case where the PVS format is used, MRP presents a lower performance decrease (0.35 bpp) than HEVC (0.55 bpp). The best results are obtained for both encoders using the 'SBE' scan. This result may occur because the LRBF images are grouped together at the end of the pseudo-video sequence, which means that due to the EPI extracting process they are grouped in the EPIs as well, thus improving the prediction.

Impact of Visual Content on Coding Efficiency

A visual inspection evaluation was performed to understand whether characteristics of the LFs content have any consistent relationship with the coding performance. Direct observations indicate that the sequence with worst compression results for HEVC and MRP, Fountain, presents many details in the bottom region of the image, making it harder to compress. This dependence on spatial frequency content seems to be confirmed by a similar analysis of the easier-to-compress

Desktop image. This image has large smooth areas, which lead to higher compression ratios. Other than visual content with high spatial coding complexity, a more detailed visual analysis also reveals that the presence of structured texture can influence the coding efficiency when the lenslet array and EPI stack arrangements are used. That is the case of the Fountain image, where the visible horizontally oriented textures apparently degrades the quality of the prediction and, consequently, decreases the compression efficiency. Further aspects are similar to those already known from traditional image coding methods.

Colour Image Coding with Minimum Rate Predictors

As indicated before, the previous results were obtained using RGB LF images for all encoders. While in standard encoders, such as HEVC, the correlations between the colour components are already exploited, in MRP there is more flexibility to exploit such similarities with pre-processing methods because it is a single-component encoder.

To exploit these correlations, the three colour components from each frame of the pseudo-video sequence were concatenated in order to form a composite frame to be encoded as a single plane with three times the number rows. Using a single plane, it is expected that the class block-based linear predictor design and the optimisation procedure of MRP are able to exploit the redundancies between different colour channels. This is mainly achieved by reducing the overhead necessary to repeat encoding of some classes in different colour channels and also by assigning the same class to blocks from different channels but with similar characteristics. To test the effectiveness of this approach, the same set of LFs represented as PVS, using the 'Spiral' image scan order, were encoded using MRP. The colour components of each frame were concatenated in two directions, side-by-side (horizontally) and top-bottom (vertically). As expected, due to MRP characteristics, the direction of this concatenation operation had little or no effect on the compression results, thus we present the results only for the vertical concatenation in Table 4.6.

Table 4.6: Coding performance for component concatenation in MRP (average bpp).

PVS			EPI		
Raster	Spiral	SBE	Raster	Spiral	SBE
7.11	**7.10**	7.11	7.75	7.68	7.43

As can be seen in Table 4.6, the results revealed just a slight increase of the MRP performance by 0.04 bpp, for sub-aperture images, and about 0.07 bpp (on average), for EPIs, when compared to the independent encoding of the colour components, shown in Table 4.4, thus (weakly) confirming the previous intuition behind the idea of concatenating the colour planes. For lenslet array, the results are of the same order of magnitude as those in Table 4.6.

Encoding and Decoding Time Complexity Analysis

The computational complexity of the coding methods described earlier was measured as the running time of the encoding and decoding operations, for both the MRP and HEVC encoders. A dedicated server running Ubuntu Server 16.04 with an Intel Xeon E3-1240 V2 @ 3.40GHz CPU and 23.5 GB of RAM was used for these experiments. The relative complexity of MRP to HEVC is shown in Table 4.7 and discussed below. In these results, the running time of HEVC is used as reference, which means that HEVC complexity equals 1.0.

Table 4.7: MRP encoding / decoding running time relative to HEVC.

Process	Lenslet	PVS			EPI		
		Raster	Spiral	SBE	Raster	Spiral	SBE
Encoding	99.1	3.7	3.8	4.1	2.9	3.0	3.4
Decoding	1.5	1.4	1.4	1.4	1.1	1.2	1.3

Due to the encoding algorithm optimisation mode, which runs for all image pixels at once, MRP is much more adversely affected by the increasing size of images. This effect is observed in the lenslet array arrangement, which is 99 times slower than HEVC. For the other LF formats, the running time complexity is much lower, ranging from 2.9 times for EPI stack spiral to 4.1 times for PVS ‘SBE’. For decoding, the MRP algorithm exhibits very close complexity relative to that of HEVC. The worst case is again for the lenslet format, where MRP is 1.5 times slower than HEVC.

It should be noted that the MRP encoder is not optimised for speed, which largely justifies the relatively high complexity of MRP encoding in comparison with HEVC. As shown by these experiments, the disparity of processing speed between MRP and HEVC decoding is significantly smaller than for encoding. Nevertheless, an optimisation of the MRP decoder implementation could easily reduce the complexity difference to HEVC.

Analysis of the Bitstream Overhead

The side information for the different data arrangements, as described in Section 4.1, is shown in Table 4.8. The last three columns show the number of bits spent in each of the syntax elements explained in Section 3.5.2, namely sequence header (SH), number of classes (NC), class partition (CP), class coefficients (CC), threshold values (ThV), and the prediction residuals (PR) for the Vespa LF (see Figure 3.10).

As it can be observed in Table 4.8, the side information represents a small percentage of the total file size, ranging from 3.05% for PVS to 9.61% for the lenslet array. The lenslet array format incurs in the highest overhead, which mostly results from its data structure with a more detailed quadtree and smaller blocks. The amount of side information required for the PVS and EPI stack formats is higher for NC, CC and ThV, due to the larger number of frames.

Table 4.8: Side information in bits for different data arrangements using Vespa image from [43].

Type		Description	Size	Freq.	Lenslet Array	PVS	EPI
Side Information	SH	Magic number, software version, width, height and number of frames, pixel bit depth, number of B type frames, bi-directional prediction type flag, number of components, number of groups, prediction order values and histogram packing information.	200	Once	600	600	600
	NC	Number of classes used for a given frame.	8	Per frame	24	5,400	10,416
	CP	Quadtree partition and block classification information.	Variable	Per frame	56,259,936	7,617,400	13,953,448
	CC	Coefficients values for each class.	Variable	Per frame	68,784	3,612,248	5,998,792
	ThV	Encoding of the probability distribution thresholds values.	Variable	Per frame	14,328	1,253,616	1,977,072
	Total	Total amount of bits spent on the side information.			56,343,672 (9.61%)	12,489,264 (3.05%)	21,940,328 (4.94%)
PR		Encoding of the prediction residuals.	Variable	Per frame	529,828,920 (90.39%)	397,219,056 (96.95%)	422,222,840 (95.06%)

4.4.2 Colour Space Effect on Coding

This section presents the results and discussion of the study addressing the impact of using other colour spaces instead of the RGB, resulting from reversible transforms, and LF data arrangements on the compression efficiency of lossless encoders. Up to seven LF arrangements (described in Section 4.1) were combined with four lossless encoders, namely JPEG 2000, JPEG-LS, HEVC, and MRP. MRP uses the same number of reference pixels as set in Section 4.4.1. The experiments in this section include a dermatological LF acquired with the same camera model as the EPFL dataset. This LF is named SkinSpots, Figure 4.8 shows the central SAI of the LF.

The simulation results are shown in Table 4.9, presented in bpp, where the average results for all combinations of encoders, data arrangement (plus scanning pattern for HEVC and MRP), and reversible transform are shown. The best results in terms of compression efficiency, for each pair *encoder - data arrangement* are underlined, and the best result for each data arrangement is highlighted in bold. Table 4.10 shows the pixel-average total variation (TV) for each colour space. The TV has been found to be a good indicator of image compression performance in lossy coding [118], which relates to the amount of residue that is left to encode in predictive lossless coding, *e.g.*, a higher value of TV means more residue to encode, thus lower compression efficiency.

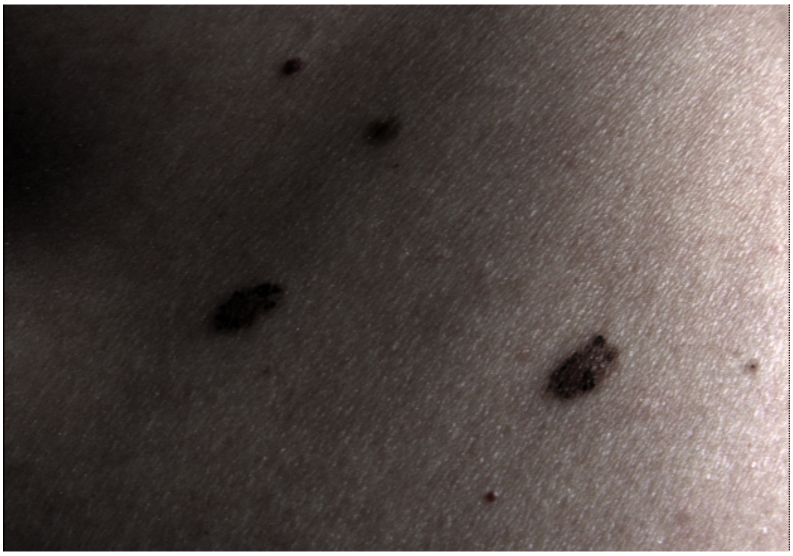


Figure 4.8: Sub-aperture image of the SkinSpots LF.

Table 4.9: Average bitrates (bpp) for all colour transforms and data arrangements.

Encoder	Arrangement	RGB	A2	LDgDb	LDgEb	RCT	RDgDb	YCoCg
JPEG 2000	Lenslet	13.21	10.31	9.95	10.08	9.92	9.99	9.97
	PVS	10.67	9.73	9.46	9.49	9.41	9.60	9.41
	EPI	10.28	9.12	8.87	8.91	8.84	8.98	8.86
JPEG-LS	Lenslet	10.14	8.83	8.54	8.65	8.52	8.60	8.55
	PVS	10.21	9.85	9.59	9.62	9.55	9.74	9.57
	EPI	8.87	8.39	8.16	8.20	8.13	8.27	8.17
HEVC	Lenslet	10.04	9.64	9.39	9.45	9.37	9.45	9.41
	PVS Raster	7.63	7.62	7.43	7.43	7.41	7.47	7.44
	PVS Spiral	7.55	7.54	7.36	7.35	7.34	7.39	7.37
	PVS SBE	7.53	7.52	7.34	7.33	7.32	7.37	7.34
	EPI Raster	8.67	8.59	8.39	8.38	8.36	8.46	8.39
	EPI Spiral	8.44	8.37	8.17	8.16	8.14	8.23	8.17
	EPI SBE	8.08	8.02	7.81	7.81	7.79	7.88	7.83
MRP	Lenslet	10.50	8.53	8.33	8.46	**8.18**	8.28	8.35
	PVS Raster	7.21	6.66	6.50	6.53	6.46	6.62	6.49
	PVS Spiral	7.19	6.61	6.46	6.48	**6.41**	6.57	6.45
	PVS SBE	7.20	6.62	6.46	6.47	6.42	6.58	6.44
	EPI Raster	7.90	7.26	7.08	7.11	7.02	7.19	7.07
	EPI Spiral	7.81	7.14	6.95	6.98	6.90	7.06	6.94
	EPI SBE	7.54	6.88	6.69	6.72	**6.66**	6.81	6.68

Analysing the results by encoder it can be seen from Table 4.9 that the Intra only encoders (JPEG 2000 and JPEG-LS) exhibit the highest compression efficiency when using the stack of EPIs, regardless of the colour transform. As for the video/inter encoders, the best results are obtained with the PVS format, using the 'SBE' scan for HEVC, and the 'Spiral' scan for MRP.

Considering the use of reversible colour transforms, it can be clearly observed that the alternative colour spaces lead to higher compression efficiency in comparison to RGB. Moreover, although the differences can be quite small (*i.e.*, in the order of tenths of bpp), the highest compression ratios are always obtained with RCT, for all encoders and data arrangements.

Table 4.10: Average total variation for all colour transforms and data arrangements.

	RGB	A2	LDgDb	LDgEb	RCT	RDgDb	YCoCg
Lenslet	38.93	19.36	18.22	18.58	**18.06**	18.38	18.45
PVS	23.88	15.99	14.83	15.03	**14.62**	15.23	14.86
EPI	32.58	17.71	16.58	16.87	**16.41**	16.82	16.73

As shown in Table 4.10, the average TV of an image consistently decreases after the conversion from RGB to any other colour space and the lowest TV is obtained for the RCT. These results, in particular the relative performance of encoding methods / colour transforms, confirm that the total variation provides a relevant indication about compression efficiency. In fact, as discussed in [118], reducing the TV of an image is expected to cut down approximation errors, leading to higher compression efficiencies. The results of Tables 4.9 and 4.10, extend previous findings to the lossless case.

In absolute terms, the MRP encoder using the PVS arrangement, with the 'Spiral' Scan, produces the best results with 6.41 bpp, compared to the 7.32 bpp of HEVC, 8.13 bpp of JPEG-LS and 8.84 bpp of JPEG 2000. The best performance obtained with HEVC and MRP when encoding the PVS and EPI stack is coherent with the findings of last section.

The trends observed in the average values listed in Table 4.9 are also consistently observed for all images used in the experiments. Table 4.11 presents some evidence of this consistency for the case of HEVC and MRP combined with the RCT transform, applied to the PVS obtained with the Spiral scan pattern.

In particular it should be pointed out that the encoding results listed for the SkinSpots image do not deviate from the general behaviour observed with the other test images. This also provides some hints about the expected performance to be observed when encoding LF dermatological images, where lossless compression is particularly relevant.

4.4.3 Disparity Compensation

For these experiments, the adopted procedure followed the lossy JPEG-Pleno CTC version 3.3. In regard to the disparity compensation, for each of the LFs, multiple block sizes and maximum disparities were used, and the resulting images from the preprocessing step were encoded with MuLE without changes in the encoder. In order to determine the configuration that results in the best coding efficiency, the Bjøntegaard metric was employed, taking as reference the HEVC anchor provided in [49].

The compression efficiency obtained by the proposed algorithm (MuLE with Disparity Compen-

Table 4.11: Bitrate (bpp) for the set of LFs, using the RCT transform and the PVS spiral scan.

Light Fields	HEVC	MRP
Ankylosaurus	7.70	6.77
Bikes	7.49	6.50
Color	7.78	6.74
Danger	7.00	6.07
Desktop	5.89	5.22
Flowers	7.19	6.26
Fountain	8.05	6.90
Friends	6.63	5.81
ISO	7.87	6.87
Magnets	7.68	6.80
Stone	7.33	6.43
Vespa	6.91	6.01
SkinSpots	7.95	7.00
Average	7.34	**6.41**

sation) in comparison with Hierarchical Warping, Merging, and Sparse Prediction (WaSP) and MuLE, is presented in Table 4.12.

Table 4.12: Disparity compensation results (% BD rate) using HEVC as reference.

Type	LF	WaSP	MuLE	**Proposed**
Lenslet	Bikes	3.35	-40.42	**-40.42 (128, 2.5)**
	Danger	-10.85	-38.79	**-38.79 (128, 2.5)**
	Fountain	13.09	-15.79	**-15.79 (64, 2.0)**
	Stone	-2.34	-32.93	**-32.93 (64, 2.0)**
HDCA	Greek	-34.67	67.51	**-9.88 (128, 2.5)**
	Sideboard	-28.01	34.76	**-13.15 (32, 1.0)**
	Tarot	24.54	155.62	**30.70 (128, 2.5)**

The last column of the table also shows, within parentheses, the best configuration for each image, in terms of block size and maximum disparity. For a better perception of the relative coding performance, the rate-distortion curves of Greek LF are shown in Figure 4.9, for the various encoders.

All the results include the cost of transmitting the block size and the disparity for each block. This cost was calculated as: $\lceil \log_2(B) \rceil + \#D \times \left\lceil \log_2\left(\frac{2\times\max(|D|)}{0.1}\right)\right\rceil$, where B is the block size, D is the array of disparities to transmit, $\#D$ is the number of elements in the array, and 0.1 is the step used for the range of disparities.

As discussed in [119], MuLE achieves the highest efficiency for the lenslet type LFs, outperforming HEVC and WaSP. However, due to the higher disparity of the HDCA LF images, the 4D-DCT employed by MuLE is not very efficient, presenting a coding BD rate up to 155.6% higher than HEVC. For lenslet LF images, the proposed disparity compensation algorithm achieves the same

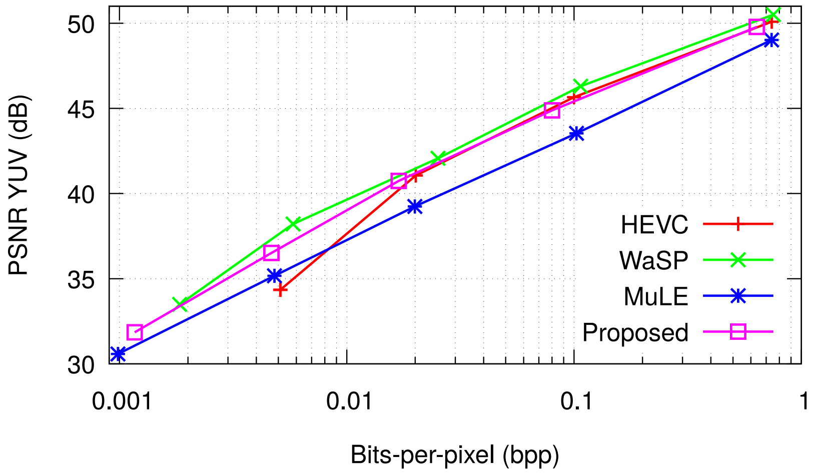


Figure 4.9: Rate distortion curves for Greek.

results as those obtained by the MuLE algorithm. This was expected due to the small disparities of lenslet LF images and the high compression efficiency of MuLE in this case (the maximum disparity of lenslet LF images is in general less than one pixel).

In regard to the HDCA LFs, the results show that MuLE benefits from the use of the disparity compensation algorithm, for instance for Greek and Sideboard images the method outperforms HEVC. Although, Tarot BD rate is still higher than HEVC, the surplus was reduced from 155.5% to 30.7%.

While for lenslet LFs the proposed method produces no BD rate changes, for HDCA images the proposed method consistently improves the compression efficiency of MuLE, achieving BD rate gains of 44.44%, on average, as can be seen in Table 4.13.

Table 4.13: Disparity compensation results (% BD rate) using MuLE as reference.

Type	LF	Proposed
HDCA	Greek	-43.12
	Sideboard	-35.85
	Tarot	-54.36
	Average	**-44.44**

4.5 Summary

This chapter introduced a research study about LFs lossless coding using MRP and other state-of-the-art image and video encoders. Several data arrangements for encoding LF data were investigated, namely lenslet array, PVS and EPI stacks, in order to assess and compare the coding performance of different encoding algorithms for each of these data formats.

Results show that the lenslet array arrangement lead to the worst performance for nearly all encoders, which allows the conclusion that, if there is no strong reason for using such format, then it is better to use PVS or EPI stacks, but preferably the former. This chapter also demonstrates that the MRP algorithm with bi-directional predictors consistently achieves higher compression ratios in comparison with the most prominent state-of-the-art lossless image and video encoders, like HEVC.

A similar study was performed while using reversible colour transforms as a pre-processing step, before the encoding process of the studied encoders. The results strengthened the previous conclusions in terms of the best data arrangement and encoder. Furthermore, as expected, it was shown that the compression efficiency of all encoders greatly benefits from the use of a reversible colour transform prior to the encoding process. Of all the studied transforms, RCT allows the encoders to achieve the highest compression efficiency.

This chapter also proposes a pre-processing method that minimises the disparity between the LF SAIs. This algorithm improves the compression efficiency of 4D-DCT based LF encoders, by aligning the SAIs, thus resulting in a more efficient representation of the DCT components. This algorithm enabled the performance of MuLE to reach similar values to that of WaSP. However, due to the characteristics of the algorithm, which increases the volume of information to encode, this method does not provide the same improvements for lossless encoders.

Due to the superior performance demonstrated in this chapter by MRP, this encoder will be the basis of the methods proposed in the next chapters.

CHAPTER 5

4D Light Fields Compression

Contents

Common representations of light fields (LFs) use four-dimensional data structures, where a given pixel is closely related not only to its spatial neighbours within the same view, *i.e.*, sub-aperture image (SAI), but also to its angular neighbours, co-located in adjacent views. In this context, two pixels are spatially co-located if their coordinates (v, u) are the same in different SAIs, *i.e.*, different angular positions. As explained in Section 2.3 (and shown by the results of the previous chapter), the four-dimensional structure of LFs presents extra layers of redundancy between pixels, when compared with regular 2D images. LFs present both spatial redundancy (as in 2D images) and angular redundancy between neighbouring pixels in the $t \times s$ dimensions. In this chapter, these redundancies are exploited to obtain compressed representations of the LF, using prediction algorithms specifically tailored to estimate pixel values based on references in the both the spatial and angular vicinity. Then, new encoding schemes are proposed, taking advantage of both types of redundancies of four-dimensional light fields to improve the coding performance of Minimum Rate Predictors (MRP) for this type of data.

The three proposed prediction and partition modes exploit the specific characteristics of the four-dimensional structure of the LFs in order to benefit from the data correlations across the different domains as a whole. Section 5.1 presents the underlying processing pipeline that is commonly used by the three proposed methods. In Section 5.2, a method based on a hexadecatree with 4D blocks is proposed [J2]; the method presented in Section 5.3 separates the block partition for the angular dimensions ($t \times s$) from that of the spatial dimensions ($v \times u$) [J2]; and the method introduced in Section 5.4 uses a conventional 2D quadtree partition [C3, C4, J2]. Section 5.5 presents the experimental evaluation of the three proposed methods. Finally, Section 5.6 summarises the chapter.

5.1 Minimum Rate Predictors Based 4D LF Encoders

The algorithms presented in this chapter were investigated and developed upon the structure of MRP. Figure 5.1 shows a simplified diagram for the family for 4D LF MRP encoders, where the Fixed Block-Size Module and Variable Block-Size Module comprise the same set of operations as previously shown in Figure 3.7. Further details can be found in Algorithm 5.1, of which one can highlight the inclusion of a pre-processing step that includes the RCT described in Chapter 4 and the histogram packing method of MRP video [62], and the 4D predictors.

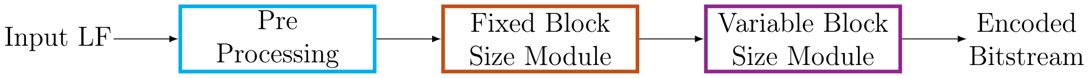


Figure 5.1: Diagram of the family of 4D LF MRP encoders.

As mentioned above, the algorithms follow the same general steps of MRP, already described in Algorithm 3.1, with the main difference being the novel four-dimensional prediction scheme that is common to the three proposed methods. In this new prediction mode the angular redundancies are exploited by using causal neighbouring SAIs, *i.e.*, already encoded SAIs, as references to extend the MRP prediction range. Due to MRP pixel-wise raster scan, not all SAI pixels might be available for prediction of the current pixel, thus the need to use a causal vicinity. The novel 4D prediction (and the mentioned causal vicinity) is represented in Figure 5.2, where the 4 causal SAI neighbours namely: left (l), left diagonal (tl), top (t), and right diagonal (tr) are used. The grey line defines the boundary between causal and non-causal pixels. Equation 5.1 shows how the prediction is computed for pixel p_0, as a linear combination of the reference pixels:

$$\begin{aligned}\hat{s}(0) = &\sum_{k=1}^{K_c} a_m(c,k)\cdot s(c,k) + \sum_{k=1}^{K_l} a_m(l,k)\cdot s(l,k)\\ &+\sum_{k=1}^{K_{tl}} a_m(tl,k)\cdot s(tl,k) + \sum_{k=1}^{K_t} a_m(t,k)\cdot s(t,k) + \sum_{k=1}^{K_{tr}} a_m(tr,k)\cdot s(tr,k),\end{aligned} \tag{5.1}$$

where $a_m(i,k)$, $i \in \{c, l, tl, t, tr\}$ represent the prediction model weights associated with pixels in the current SAI (c) and the four causal neighbouring references, respectively, for the m-th class.

Algorithm 5.1 4D MRP family high level algorithm.

```
Input: image
#Pre-processing
Apply RCT
Apply histogram packing
#Fixed Block Size Module
#Initial block classification
Sort and classify $F_b$ blocks by pixel variance
#1$^{st}$ optimisation loop using fixed block size
for 1 to MAX_ITERATIONS do
#4D Predictors design
    for each class do
        Compute the prediction coefficients $(a_m)$
        Calculate residuals
    Calculate prediction residue encoding cost $(B_r)$
#Groups optimisation
    for each class do
        Compute the context defining variable $(C)$ quantisation thresholds to minimise $B_r$
    Calculate $B_r$
#Classification optimisation
    for each $F_b$ block do
        Move neighbour blocks classes to front of list
        Calculate residuals for all classes
        Select class that minimises $B_r$
    Calculate $B_r$
    if 10 iterations without improvement then
        end for loop
#Variable Block Size Module
#2$^{nd}$ optimisation loop using variable block size
for 1 to MAX_ITERATIONS do
    Calculate prediction coefficients encoding cost $(B_a)$
#Groups optimisation
    for each class do
        Compute $C$ quantisation thresholds to minimise $B_r$
        Update shape parameter in probability models
    Calculate quantisation thresholds cost $(B_t)$
#Classification optimisation using VBS
    for Each $V_b$ block do
        ClassOptimisation
    Calculate cost $J$
    if 10 iterations without improvement then
        end for loop
Remove non-utilised classes
Run arithmetic coding
Output: Encoded bitstream
```

$s(i,k)$, $i \in \{c, l, tl, t, tr\}$ are the pixel values in the current and reference SAIs at position k of the prediction support region, and K_i, $i \in \{c, l, tl, t, tr\}$ are the sizes of each prediction support region, which may have different sizes and shapes.

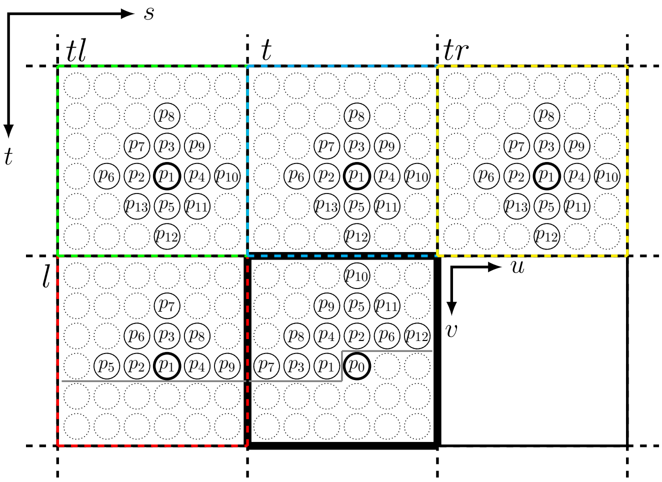


Figure 5.2: Representation of 4D prediction in MRP.

The context of the prediction errors is associated with the inverse of the distances between the current pixel and its reference pixels, and also with the prediction errors of said pixels. A weighted sum of these terms, computed as defined by Equation 5.2, is used as a context-defining variable, represented symbolically by C.

$$C = \sum_{k=1}^{K_c} \frac{1}{\delta_k^c} e_c + \sum_{k=1}^{K_l} \frac{1}{\delta_k^l} e_l + \sum_{k=1}^{K_{tl}} \frac{1}{\delta_k^{tl}} e_{tl} + \sum_{k=1}^{K_t} \frac{1}{\delta_k^t} e_t + \sum_{k=1}^{K_{tr}} \frac{1}{\delta_k^{tr}} e_{tr}, \tag{5.2}$$

where $e_i = |s(i,k) - \hat{s}(i,k)|$ and δ_k^i are weighting factors proportional to the Euclidean distance between the current pixel (p_0) and its references (p_k). The δ_k weight is defined as $\delta_k = \frac{\sqrt{d_v(k)^2 + d_u(k)^2 + \lambda^2}}{64}$, where λ represents the distance between SAIs and it is set to 0 for δ_k^c and 1 for all other δ_k^i. The resulting C undergoes a quantisation step, using the thresholds described in Section 3.4, and is then used to select the adaptive conditional probability models.

The three prediction modes proposed in this chapter all use the previously described 4D prediction, with the differences between them being related to the partitioning of the LF into blocks, and the type of partition used to obtain the sub-blocks. This is represented in Algorithm 5.1 by the variables F_b and V_b that represent the size of the top level blocks in the the fixed and variable block size modules, respectively. These values are different in the three proposed algorithms as they are closely related to the type of partitioning used by each of them. The following sections describe in detail each of the proposed prediction modes.

5.2 4D-Minimum Rate Predictors

The Four-Dimensional Minimum Rate Predictors (4D-MRP) is based on two principles: four dimensional prediction and partition of the LF into 4D blocks for class selection. The motivation for using 4D blocks is supported by the fact that, in general, the same class is selected for co-located blocks in neighbouring SAIs, as shown in Figure 5.3, where each class is represented by a different colour. It can be seen that in each SAI, separated by white dashed lines, the same structures (*e.g.*, the wheel of the bicycle) presents the same colours in roughly the same regions. This means that the same class was selected for these regions in the four SAIs. Such information indicates that bitrate savings may be achieved by reducing the quadtree signalling cost. Since using a 4D block instead of 2D, allows the encoder to avoid signalling each 2D block partitioning in each SAI.

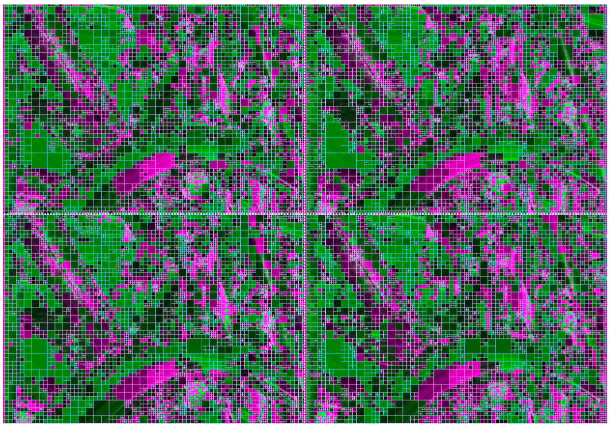
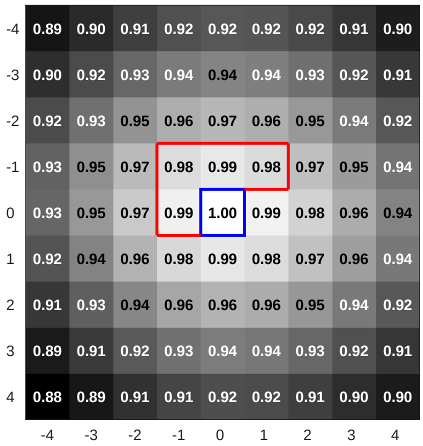


Figure 5.3: Example of MRP class selection in neighbouring SAIs, using Bikes image from [43], where each class is represented by a different colour.

The classification represented in Figure 5.3 is easily explained when considering that the SAIs in the LF present a high angular correlation amongst themselves. This means that the structures in each SAI are quite similar and, therefore, can be efficiently predicted using the same prediction model. Figure 5.4 shows the average correlation of all the SAIs in the LF and their four wide neighbourhood. In this figure the central SAI, in blue, represents the reference for the correlation calculation (and therefore the correlation with itself is one), in red are highlighted the positions of the SAIs that support the prediction, and lighter colours represent higher correlation values. As can be seen from Figure 5.4 neighbour SAIs to the reference in blue are highly correlated, with values ranging from 0.98 to 0.99.



	-4	-3	-2	-1	0	1	2	3	4
-4	0.89	0.90	0.91	0.92	0.92	0.92	0.92	0.91	0.90
-3	0.90	0.92	0.93	0.94	0.94	0.94	0.93	0.92	0.91
-2	0.92	0.93	0.95	0.96	0.97	0.96	0.95	0.94	0.92
-1	0.93	0.95	0.97	0.98	0.99	0.98	0.97	0.95	0.94
0	0.93	0.95	0.97	0.99	1.00	0.99	0.98	0.96	0.94
1	0.92	0.94	0.96	0.98	0.99	0.98	0.97	0.96	0.94
2	0.91	0.93	0.94	0.96	0.96	0.96	0.95	0.94	0.92
3	0.89	0.91	0.92	0.93	0.94	0.94	0.93	0.92	0.91
4	0.88	0.89	0.91	0.91	0.92	0.92	0.91	0.90	0.90

Figure 5.4: Average correlation between a central SAI, in blue, and a four wide neighbourhood, using Bikes image from [43], in red are highlighted the SAIs that support the 4D prediction.

As stated in Section 3.4, in MRP, a set of prediction coefficients, $a_m(i,k)$, are determined for each class independently. Then each pixel of the current SAI is assigned a class corresponding to the predictor that minimises the cost J in Equation 3.20. By using 4D blocks, the MRP quadtree partition is expanded into a hexadecatree, where each block is divided in all the four dimensions simultaneously, resulting in 16 sub-blocks in the 4D space. Let a block be represented by: $\mathcal{L}_B(t,s,v,u)$, with $t \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < T\}$, $s \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < S\}$, $v \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < V\}$, and $u \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < U\}$, in the block coordinates. Then the hexadecatree partition can be defined by Equation 5.3.

$$\begin{aligned} &P_{\text{Hex}}(\mathcal{L}_B(t,s,v,u)) = \\ &\quad \{\mathcal{L}_{B_i}(t_i,s_i,v_i,u_i), \forall i \in \{0,\ldots,15\}\}\,, \text{ with} \end{aligned} \begin{cases} t_i = a + T \times (1 + (-1)^{\lfloor i/8 \rfloor + 1})/4 \\ s_i = b + S \times (1 + (-1)^{\lfloor i/4 \rfloor + 1})/4 \\ v_i = c + V \times (1 + (-1)^{\lfloor i/2 \rfloor + 1})/4 \\ u_i = d + U \times (1 + (-1)^{i+1})/4 \\ \\ \forall a \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < T/2\} \\ \forall b \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < S/2\} \\ \forall c \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < V/2\} \\ \forall d \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < U/2\} \end{cases}, \tag{5.3}$$

where T, S, V, and U represent the dimensions of the 4D block, $\lfloor . \rfloor$ represents the floor operation, and a, b, c, and d are auxiliary variables.

Figure 5.5a represents the tree structure partition and Figure 5.5b shows a LF 4D block with four by four SAIs and the visual representation of its 4D partition. In Figure 5.5b the angular dimensions (t,s) are partitioned by the blue lines and the spatial dimensions (v,u) by the green lines, the index of each 4D block is highlighted in white circles in each SAI. The result of this operation are 16 sub-blocks as the original block is split along all of its dimensions at the same time. As pointed out before, the hexadecatree partitioning is expected to result in a larger number of pixels being classified with lower signalling cost, thus increasing compression efficiency.

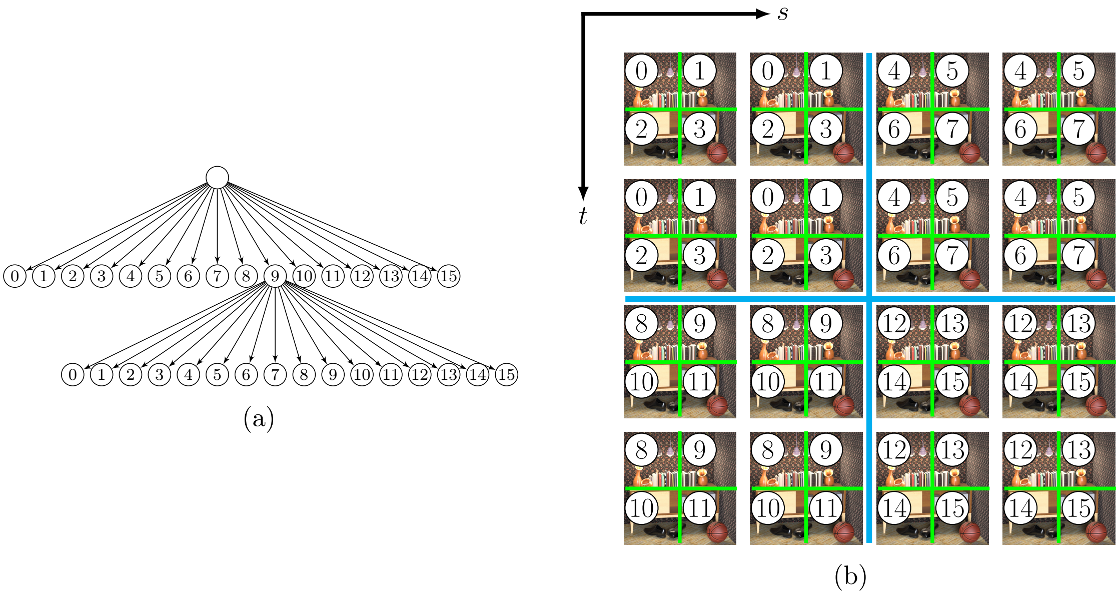
In the *Optimise Classification* stage, shown in Figure 3.7, the hexadecatree partition and block classification optimisation are performed. This optimisation aims at minimising the bitrate for both side information, represented by B_m in Equation 3.20, and the prediction error cost. Thus, B_m can be represented by:

$$B_m = B_{m_{\text{flags}}} + B_{m_{\text{class}}}, \tag{5.4}$$

where $B_{m_{\text{flags}}}$, given by Equation 5.5, represents the cost of encoding the partition flags and $B_{m_{\text{class}}}$ the cost of signalling a class for each resulting block.

$$B_{m_{\text{flags}}} = \begin{cases} -\log_2(P_{\text{ctx}}), & \text{not partitioned} \\ -\log_2(1 - P_{\text{ctx}}), & \text{partitioned} \end{cases} \tag{5.5}$$

The calculation of the cost of encoding the partition flags, shown in Equation 5.5, is given by the inverse base two logarithm of the quantised probability of blocks in a given context not being



(a)

(b)

Figure 5.5: Hexadecatree partition in the 4D-MRP: (a) tree structure partition example, and (b) visual representation of the same 4D block partition for the first level only, where the partition index of each block is shown in white circles.

selected for partition, P_{ctx}, given by:

$$\begin{aligned} P_{\text{ctx}} = & \underset{p}{\operatorname{argmin}} \left\{ -\log_2(p) \cdot \#ctx_0 \; -\log_2(1-p) \cdot \#ctx_1 \right\}, \\ \text{with} \qquad & p \in [0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95], \end{aligned} \tag{5.6}$$

where $\#ctx_0$ and $\#ctx_1$ represent the number of blocks, in a given context, that were not partitioned and partitioned, respectively, *i.e.*, the occurrences of 0 and 1 partition flags. p represents the probability that minimises the cost of encoding these flags. This probability is selected in a discrete range to minimise the cost of transmitting the real probability value, this way only a single index needs to be sent to the decoder. The quantified probability calculation of Equation 5.6 follows that used in MRP, which was first introduced with the variable block size in [68].

Finally, the context calculation for the quadtree flags is determined at each level by the number of neighbouring blocks partitioned at said level:

$$\mathrm{C}_{\text{qtflags}}(level, b) = \sum_{n_b \in N(level,b)} \text{qtflags}(n_b), \tag{5.7}$$

where b represents the block for which the context is being calculated, qtflags the partition flag for block n_b with the size given by *level* (0 and 1 for non-partitioned and partitioned blocks, respectively), and n_b represents a neighbouring block of b belonging to the set $N(level, b)$ of blocks both neighbour and at the same *level* of b. As 4D-MRP uses 4 levels for the block size and 5 neighbours are being considered in the hexadecatree there are a total of 24 different contexts.

In the *Design Predictors* stage of the encoding process, fixed size blocks are used to determine the predictor coefficients of each class m. These operations are affected by the block size, as is the coding performance, through the calculation of the coefficients $a_m(i,k)$, which decreases when a larger number of pixels exists. Experiments showed that the $8 \times 8 \times 8 \times 8$ pixels fixed block-size, analogous to what is used in MRP, was too large for the 4D space, which resulted in blocks of $8^4 = 4096$ pixels total. The same experiments showed that blocks of $4^4 = 256$ pixels were a good compromise in terms of compression ratio. Thus in this work, when 4D blocks are used in the *Fixed Block-Size* module, the size is set to $F_b = 4 \times 4 \times 4 \times 4$. For the *Variable Block-Size* module initial the block size was set to $V_b = 32 \times 32 \times 32 \times 32$.

5.3 Dual-Tree 4D-Minimum Rate Predictors

The Dual-Tree 4D Minimum Rate Predictors (DT-4D-MRP) algorithm follows the same prediction and classifications principles of 4D-MRP. However, a new dual-tree partition is introduced, that separates the partition of the angular dimensions from that of the spatial dimensions. This way, each block is either not partitioned, or partitioned on the angular or spatial dimensions. Unlike the common quadtree of MRP, the new dual-tree algorithm requires a ternary signalling for its definition. The following sections describe the new dual-tree partition and the redesigned signalling of the dual-tree, respectively. The block sizes F_b $(4\times4\times4\times4)$ and V_b $(32\times32\times32\times32)$ are the same as in the 4D-MRP case.

5.3.1 Dual-Tree Partitioning

This partitioning mode, named DT-4D-MRP, is intended to improve the problem created by hexadecatrees, which tend to use very fine partitioning with a high number of $2 \times 2 \times 2 \times 2$ blocks (the smallest possible size), albeit still achieving better performance than MRP, as shall be seen in Section 5.5. As expected, using smaller block sizes requires more signalling bits to encode the partition tree. In order to improve the compression performance a new mode which independently partitions the spatial and angular dimensions is introduced in DT-4D-MRP. This proposal is supported by different sampling present on the spatial and angular dimensions at the acquisition, which means that partitioning both dimensions at the same time may lead to over-partitioning in one of the dimensions pairs.

In this mode, the block partitioning algorithm was changed to decouple the angular from the spatial partitioning. Therefore, rather than partitioning in 16 blocks as in 4D-MRP, each 4D block was partitioned either into four blocks in the angular dimensions (t,s), into four blocks in the spatial dimensions (v,u), or not partitioned at all. Let a block be represented by $\mathcal{L}_B(t,s,v,u)$, with $t \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < T\}$, $s \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < S\}$, $v \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < V\}$, and $u \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < U\}$, in the block coordinates. Then the dual-tree partition can be defined by Equations 5.8 and 5.9, for the spatial and angular partitions respectively.

$$P_{\mathrm{DT_S}}(\mathcal{L}_B(t,s,v,u)) = \\ \{\mathcal{L}_{B_i}(t,s,v_i,u_i), \forall i \in \{0,\ldots,3\}\}\,, \text{ with } \begin{cases} \forall t \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < T\} \\ \forall s \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < S\} \\ v_i = a + V \times (1 + (-1)^{\lfloor i/2 \rfloor + 1})/4 \\ u_i = b + U \times (1 + (-1)^{i+1})/4 \\ \\ \forall a \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < V/2\} \\ \forall b \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < U/2\} \end{cases} \tag{5.8}$$

$$P_{\mathrm{DT_A}}(\mathcal{L}_B(t,s,v,u)) = \\ \{\mathcal{L}_{B_i}(t_i,s_i,v,u), \forall i \in \{0,\ldots,3\}\}\,, \text{ with } \begin{cases} t_i = a + T \times (1 + (-1)^{\lfloor i/2 \rfloor + 1})/4 \\ s_i = b + S \times (1 + (-1)^{i+1})/4 \\ \forall v \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < V\} \\ \forall u \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < U\} \\ \\ \forall a \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < T/2\} \\ \forall b \in \{\gamma \in \mathbb{N} \,|\, 0 \leq \gamma < S/2\} \end{cases} \tag{5.9}$$

The partition tree structure of DT-4D-MRP is represented in Figure 5.6, where 'S' represents the spatial partition, 'A' the angular partition and 'N' non-partitioned blocks. Figures 5.7a and 5.7b show visual representations of the separate quadtrees for spatial and angular dimensions, respectively.

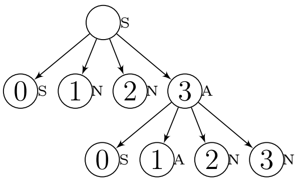
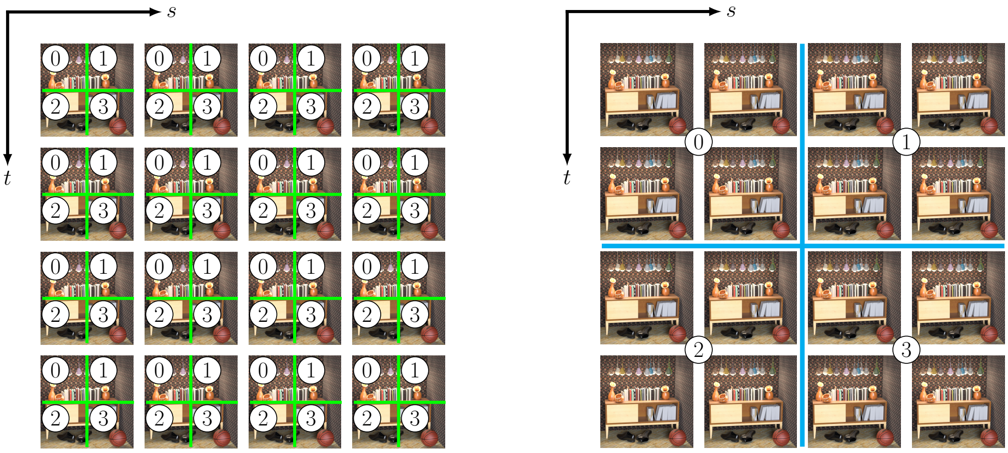


Figure 5.6: Dual quadtree partition in 4D-MRP: tree structure partition example.



(a) Spatial quadtree.

(b) Angular quadtree.

Figure 5.7: First level of the dual quadtree partition in 4D-MRP, using Sideboard LF from [51]. The partition index of each block is shown in white circles.

5.3.2 Arithmetic Coding of the Partition Tree Signalling

In this new tree structure the decisions are no longer binary but ternary, since there are three options for partitioning each block (*i.e.*, spatial, angular, none). In DT-4D-MRP, the cost of selecting a given partition mode was redefined to reflect the new signalling structure:

$$B_{m_{\text{flags}}} = \begin{cases} -\log_2(P_N), & \text{if flag = 'N'} \\ -\log_2(P_{S|\bar{N}} \cdot P_{\bar{N}}), & \text{if flag = 'S'} \\ -\log_2((1-P_{S|\bar{N}}) \cdot P_{\bar{N}}), & \text{if flag = 'A'} \end{cases}, \tag{5.10}$$

where P_N represents the probability of the 'N' flag, $P_{\bar{N}} = 1 - P_N$ the complementary probability of P_N, and $P_{S|\bar{N}}$ the probability of the 'S' flag given $\bar{N}$, resulting from:

$$\begin{aligned} P_N &= \underset{p}{\operatorname{argmin}} \left\{-\log_2(p) \cdot ctx_N \; -\log_2(1-p) \cdot (ctx_S + ctx_A)\right\}, \\ P_{S|\bar{N}} &= \underset{p}{\operatorname{argmin}} \left\{-\log_2(p) \cdot ctx_S \; -\log_2(1-p) \cdot ctx_A\right\}, \end{aligned} \tag{5.11}$$

where $p \in [0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95]$, and ctx_i represents $\text{context}_{DT}(i)$, given by Equation 5.12.

For the DT-4D-MRP, the number of possible contexts was reduced, and the computation of the context was replaced by the number of times each flag ('S', 'A', or 'N') is chosen in the tree:

$$\text{C}_{\text{qtflags}_{DT}}(F) = \sum_{b \in \text{tree}} \delta_{\text{flag}(b),F}, \tag{5.12}$$

where F represents the flag for which the context is being determined, b represents each block in the tree, flag(b) is the partition flag of each block b and δ is the Kronecker delta.

These operations are represented by Algorithm 5.2. Following the same approach as in the 4D-MRP case, a table of probabilities (*treeprob*) is used to facilitate the encoding of the partition flags context, as the index is more efficiently encoded than the actual probability value. The optimal index *o_index* corresponds to the probability p that minimises the expression in Line 6 and Line 18 of the algorithm. The cost of each flag, represented as cost $= -log(P)$, is calculated in two steps. First the probability of the 'N' flag is determined, by considering the probabilities of the remaining flags, 'S' and 'A', to represent $1 - P_N$. Then, the relative probabilities of 'S' and 'A' are computed. Which are then normalised by multiplying them by a factor of $1 - P_N$, such that $\sum_{j=N,S,A} P_j = 1$. These probabilities are used to set the probability models for the arithmetic coding of the tree partition. The flag costs are used in the optimisation of the partition tree.

The remaining steps in the arithmetic coding of the partition flags and class selection are kept unchanged. Essentially, starting with the first block, the partition flags are encoded until a block is no longer partitioned, then the selected class is encoded in the bitstream.

Algorithm 5.2 Calculation of the partition flags cost.

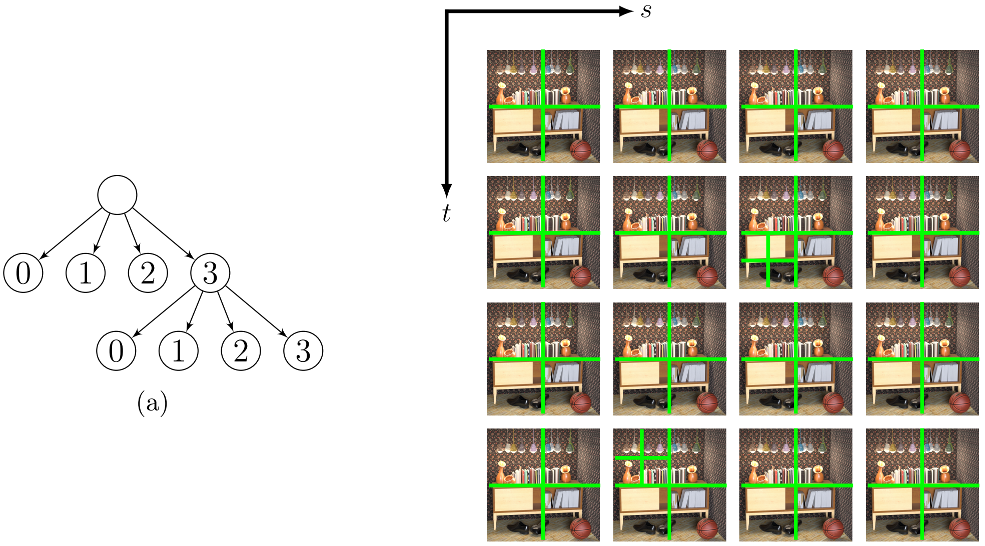
```
cost = inf
treeprob = $[0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95]$
#Determine cost of symbol 'N'
for $i = 0$ to 6 do
    $P$ = treeprob[$i$]
    $c = -\log(P)\cdot$ context['N'] $-\log(1-P)\cdot$ (context['S'] + context['A'])
    if $c <$ cost then
        cost = $c$
        o_index = $i$
$P_N$ = treeprob[o_index]
flagcost['N'] = $-\log_2(P_N)$
cost = inf
#Determine cost of symbols 'S' and 'A'
for $i = 0$ to 6 do
    $P$ = treeprob[$i$]
    $c = -\log(P)\cdot$ context['S'] $-\log(1-P)\cdot$ context['A']
    if $c <$ cost then
        cost = $c$
        o_index = $i$
$P$ = treeprob[o_index]
flagcost['S'] = $-\log_2(P\cdot(1-P_N))$
flagcost['A'] = $-\log_2((1-P)\cdot(1-P_N))$
```

5.4 Multi-Reference Minimum Rate Predictors

In Section 5.3, the DT-4D-MRP was proposed as an alternative to 4D-MRP to reduce the large overhead of hexadecatree partitioning signalling, due to the large amount of smallest size blocks used in the encoder. This section proposes a different approach to deal with this overhead. In Multi-reference Minimum Rate Predictors (M-MRP), the 4D partitioning is abandoned and only 2D partitions are used. This is done by treating the LF as a single 2D image for partition, which results in a more sensible spatial partitioning with fewer smallest blocks, despite the fact that this approach forsakes the potential benefits of using 4D blocks, as presented in 4D-MRP and DT-4D-MRP. This way the block sizes were set to be the same as in MRP, $F_b = 8\times 8$ and $V_b = 32\times 32$, for the fixed and variable block size cases, respectively. The M-MRP combines the 4D prediction described in the Section 5.2 (represented in Figure 5.2) with the conventional 2D quadtree partition, represented in Figure 5.8a and Figure 5.8b. Let a 2D block be represented by $\mathcal{L}_B(v,u)$, with $v \in \{\gamma \in \mathbb{N}\,|\,0 \le \gamma < V\}$, and $u \in \{\gamma \in \mathbb{N}\,|\,0 \le \gamma < U\}$, in the block coordinates, then the M-MRP partition can be represented by (a and b are auxiliary variables):

$$
\begin{aligned}
&P_{\mathrm{DT_S}}(\mathcal{L}_B(v,u)) = \\
&\qquad \{\mathcal{L}_{B_i}(v_i,u_i), \forall i \in \{0,\dots,3\}\}\,, \quad \text{with}
\end{aligned}
\begin{cases}
v_i = a + V \times (1 + (-1)^{\lfloor i/2 \rfloor + 1})/4 \\
u_i = b + U \times (1 + (-1)^{i+1})/4 \\
\\
\forall a \in \{\gamma \in \mathbb{N}\,|\,0 \le \gamma < V/2\} \\
\forall b \in \{\gamma \in \mathbb{N}\,|\,0 \le \gamma < U/2\}
\end{cases}
. \qquad (5.13)
$$

The partition structure of M-MRP is similar to that of 4D-MRP, in particular, and to MRP, in general. Thus, no further modifications to the signalling structure are needed. Additionally, due to the similarity of the partition method between 4D-MRP and M-MRP, the overall computational complexity remains roughly the same, as will be evidenced in the experimental results in Section 5.5.



(a)

(b)

Figure 5.8: Two dimension quadtree partition in M-MRP: (a) tree structure partition example; and (b) visual representation of the quadtree using different levels, for the Sideboard image from [51].

5.5 Experimental Evaluation

Extensive experiments were carried out to assess the performance of the proposed algorithms. In the first set of experiments the three algorithms are compared with other state-of-the-art encoders, like High Efficiency Video Coding (HEVC) or Versatile Video Coding (VVC). Then an analysis of the computational complexity is undertaken. The last set of experiments deals with the trade-off between coding efficiency and computational complexity. The experiments described in the next sections follow the conditions set in Section 2.4.

5.5.1 Comparison with State-of-the-Art Encoders

The three modes described in Section 5.2, 4D-MRP, DT-4D-MRP, and M-MRP, have been designed to exploit the unique characteristics of light field images. Their coding efficiencies were evaluated by encoding all the LFs listed in Section 2.4. The results are analysed and discussed in the following.

Experiments Using the EPFL and HCI Datasets

The results in Table 5.1 show that MRP surpasses standard codecs like Lossless and Near-lossless Compression of Continuous-tone Still Images (JPEG-LS), HEVC, or VVC. However, neither of these standard encoders, including MRP, is adapted to exploit the particular characteristics of the 4D LF data representation. The proposed 4D-MRP, M-MRP and DT-4D-MRP schemes achieve higher compression ratio than MRP, 9.68, 9.66, and 9.59 bits-per-pixel (bpp) on average, respectively. This is translated to a difference of 1.00 to 1.09 bpp on average, when compared with MRP, which encodes these LFs with an average bitrate of 10.69 bpp Such differences represent roughly 10% of bitrate savings. When compared with the most recent state-of-the-art image and video coding standard, VVC, which achieves an average bitrate of 11.34 bpp, the proposed methods, with bitrates ranging from 9.68 to 9.59 bpp, on average, present savings of 15%, on average, surpassing VVC by 1.66 to 1.75 bpp.

Table 5.1: Compression results (bpp) for EPFL and HCI datasets.

Type	Light Fields	JPEG-LS	HM	VVC	MRP	4D-MRP	M-MRP	DT-4D-MRP
LL	Bikes	16.42	13.69	13.72	12.73	11.43	11.47	**11.34**
	Danger	16.18	13.03	12.97	11.89	10.80	10.80	**10.71**
	Fountain	16.66	14.43	14.52	13.28	12.09	12.06	**11.99**
	Stone	16.77	13.44	13.38	12.36	11.29	11.20	**11.17**
HDCA	Greek	7.25	5.55	5.36	5.40	4.82	**4.78**	**4.78**
	Sideboard	12.19	8.17	8.09	8.43	7.66	7.62	**7.58**
Average		14.24	11.39	11.34	10.68	9.68	9.66	**9.59**

In regard to the individual evaluation of the three proposed methods, it can be seen that the DT-4D-MRP achieves the lowest bitrate (9.59 bpp), both on average and for each image. When compared with 4D-MRP (9.68 bpp) and M-MRP (9.66 bpp), the DT-4D-MRP is able to outperform these by 0.7 to 0.9%, respectively. The dual quadtree prediction seems to provide better adaptation of partitions to the image data, as can be inferred by the higher compression ratio. The 4D-MRP encoder presents the highest bitrate of the three, which is explained by the highly granular partition used by the encoder. This is due to the simultaneous partitioning in all dimensions, which results in smaller blocks and, consequently, more side information. M-MRP, which uses 2D instead of 4D partitions, achieves an intermediate coding efficiency when compared with 4D-MRP and DT-4D-MRP. This is owed to the fact that 2D partitions can use blocks that include pixels from multiples SAIs and blocks with larger sizes, as splitting in the four-dimensions at once is not mandatory (or possible), as was the case of 4D-MRP.

Experiments Using the SKINL2 Dataset

Further coding experiments using the light field image dataset of skin lesions (SKINL2) revealed that performance results are consistent with those observed for non-dermoscopic images. The results shown in Table 5.2 further assert the better performance of the proposed methods when compared with the state-of-the-art encoders. In fact, while the MRP encodes these LFs with an average of 7.96 bpp, the proposed methods present an average bitrate of 7.11, 6.94, and 6.97 bpp, for 4D-MRP, M-MRP, and DT-4D-MRP, respectively. These results represent bitrate differences ranging from 0.85 to 1.02 bpp, and bitrate savings larger than 11%. In comparison with the VVC, which has an average bitrate of 8.57 bpp, the proposed methods present gains of 1.46 to 1.63 bpp, on average, which represent up to 19% of bitrate savings. These results show that the proposed algorithms are able to achieve the highest compression efficiency for the skin lesions dataset, which is a noteworthy conclusion as in many application scenarios medical images are required to be losslessly compressed.

Table 5.2: Compression results (bpp) for the SKINL2.

Light Fields	JPEG-LS	HEVC	VVC	MRP	4D-MRP	M-MRP	DT-4D-MRP
Img1	13.39	8.65	8.17	8.06	7.01	**6.79**	6.84
Img2	13.59	9.10	8.74	7.81	7.10	6.98	**6.97**
Img3	13.61	9.18	8.79	8.00	7.23	**7.07**	7.09
Average	13.53	8.98	8.57	7.96	7.11	**6.94**	6.97

In regard to the comparison between the three proposed methods, it can be observed that DT-4D-MRP and M-MRP achieve similar compression efficiency, with a slight advantage for M-MRP. For this dataset (SKINL2) the bitrate savings of M-MRP are 2% and 0.43% when compared with 4D-MRP and DT-4D-MRP, respectively. This shows that there is a relative loss of efficiency of 4D-MRP and DT-4D-MRP for the current dataset, when compared with the results of Table 5.1. This might be explained by the larger resolution of the images in the SKINL2 dataset, that seems to favour the algorithms of M-MRP.

Overall the experimental evaluation demonstrates, in a consistent manner, that using both 4D prediction and 4D partition to encode LFs with MRP, leads to the higher coding efficiency of DT-4D-MRP.

5.5.2 Analysis of the Computational Complexity

Besides coding efficiency, the computational complexity is an important, although not always primary, benchmarking variable. The complexity results pertaining to the different encoders, evaluated by measuring the LF encoding times evaluated on the École Polytechnique Fédérale de Lausanne (EPFL) and Heidelberg Collaboratory for Image Processing (HCI) datasets, are presented in Table 5.3. The encoding time ratio shown in this table is obtained by dividing the encoding time of an encoder by that of MRP. These results were obtained on a computer equipped with an Intel Xeon(R) Silver 4114@2.20GHz CPU, and 192GB of DDR4@2666MHz RAM, running Ubuntu Server 20.04.1 LTS. The results show that the DT-4D-MRP has the worst performance in terms of computational complexity, when compared with the other encoders, taking on average 78 times longer than MRP to encode the same LFs. As stated in Section 5.3, this high computational complexity is due to the large number of partitioning options that need to be tested in the dual quadtree. The 4D-MRP and M-MRP proposals have computational complexities of the same magnitude, about 4 and 7 times that of MRP, respectively. However, they have a smaller compression performance than that of DT-4D-MRP. Although VVC presents about the same complexity as the 4D-MRP and M-MRP encoders, its compression performance is much lower than that of these encoders. The performance of VVC is on pair with that of HEVC, despite being an older standard. This might be explained by the fact that VVC performance optimisation has been so far mostly focused on lossy coding tools rather than improving the lossless modes.

Table 5.3: Encoding time ratio compared with MRP, for EPFL and HCI datasets.

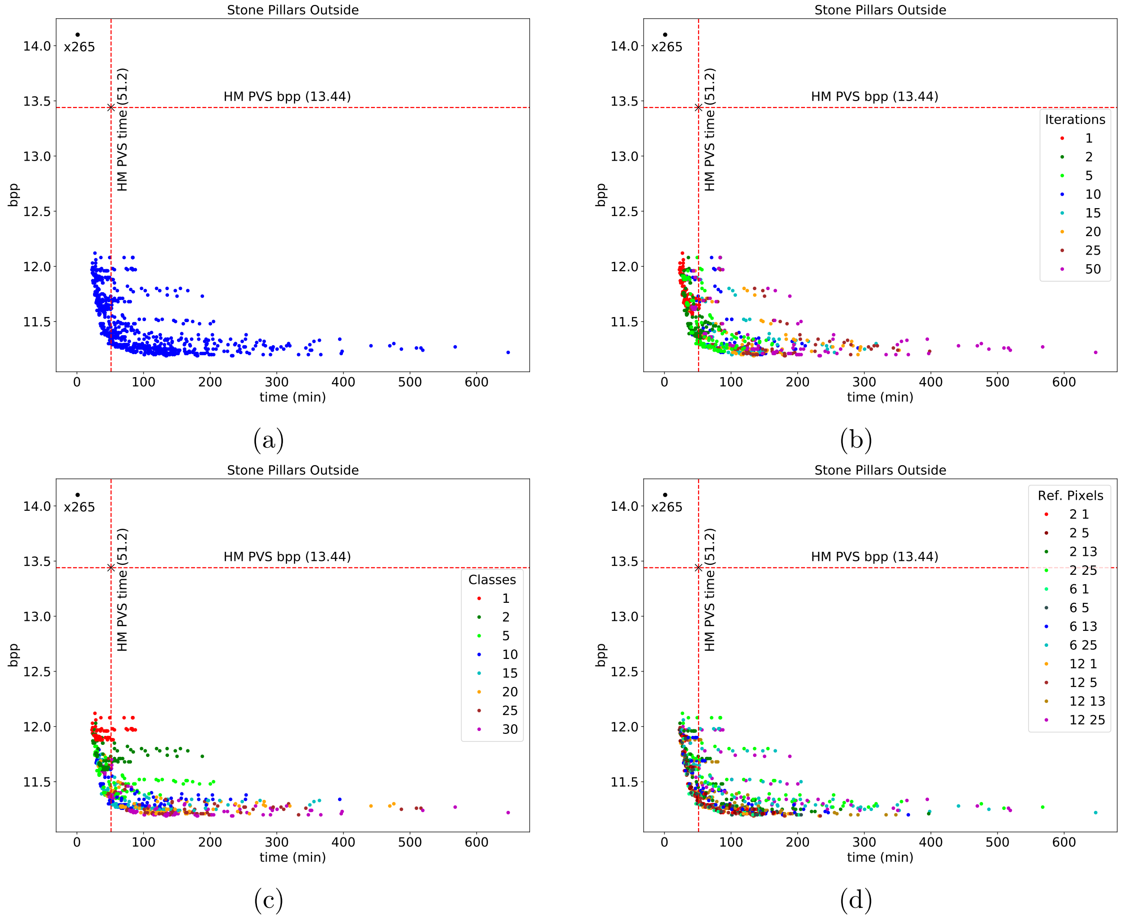
Type	Light Fields	JPEG-LS	HM	VVC	MRP	4D-MRP	M-MRP	DT-4D-MRP
LL	Bikes	1.5E-3	0.2	3.6	1.0	3.5	6.4	91.9
	Dange	2.0E-3	0.3	4.4	1.0	4.7	9.3	90.2
	Fountain	1.9E-3	0.3	4.6	1.0	3.7	8.8	122.5
	Stone	1.7E-3	0.3	4.5	1.0	3.7	8.0	89.0
HDCA	Greek	8.9E-4	0.1	3.0	1.0	4.9	6.6	35.1
	Sideboard	1.0E-3	0.2	2.6	1.0	4.3	4.1	44.4
Average		1.5E-3	0.2	3.8	1.0	4.1	7.1	78.9

5.5.3 Rate-Complexity Trade-Off in Multi-Reference Minimum Rate Predictors

This section investigates the relationship between the input parameters of M-MRP and their impact on the rate-complexity performance. An experimental study was carried out by varying the three main encoder parameters, namely, the number of optimisation iterations, the number of prediction modes, *i.e.*, classes, and the number of reference pixels. The HEVC reference software (HM) encoding time is used as a reference for the comparative study of the trade-off between M-MRP coding efficiency and computational complexity. The x265 coder, which is an optimised implementation of the HEVC standard, is also used for comparison. The analysis is based on both compression efficiency, measured in bpp, and complexity, measured by the processing time, in minutes.

The three input parameters were tested with multiple values resulting in a set of operation points defined by coding efficiency (bpp) versus computational complexity (in minutes), shown in Figure 5.9 for the Stone Pillars Outside LF, and in Figure 5.10 for the Greek LF. In these figures, the red dashed line represents the limits of HM pseudo-video sequence (PVS), whose intersection is represented by a black cross, and the x265 reference is represented by a black dot. For the x265 experiments the configurations described for the anchor in the Common Test

Conditions of the JPEG-Pleno (JPEG-Pleno CTC) were used, with appropriate changes for the lossless case, and the parallel processing tools disabled to provide a fair comparison.



(a)

(b)

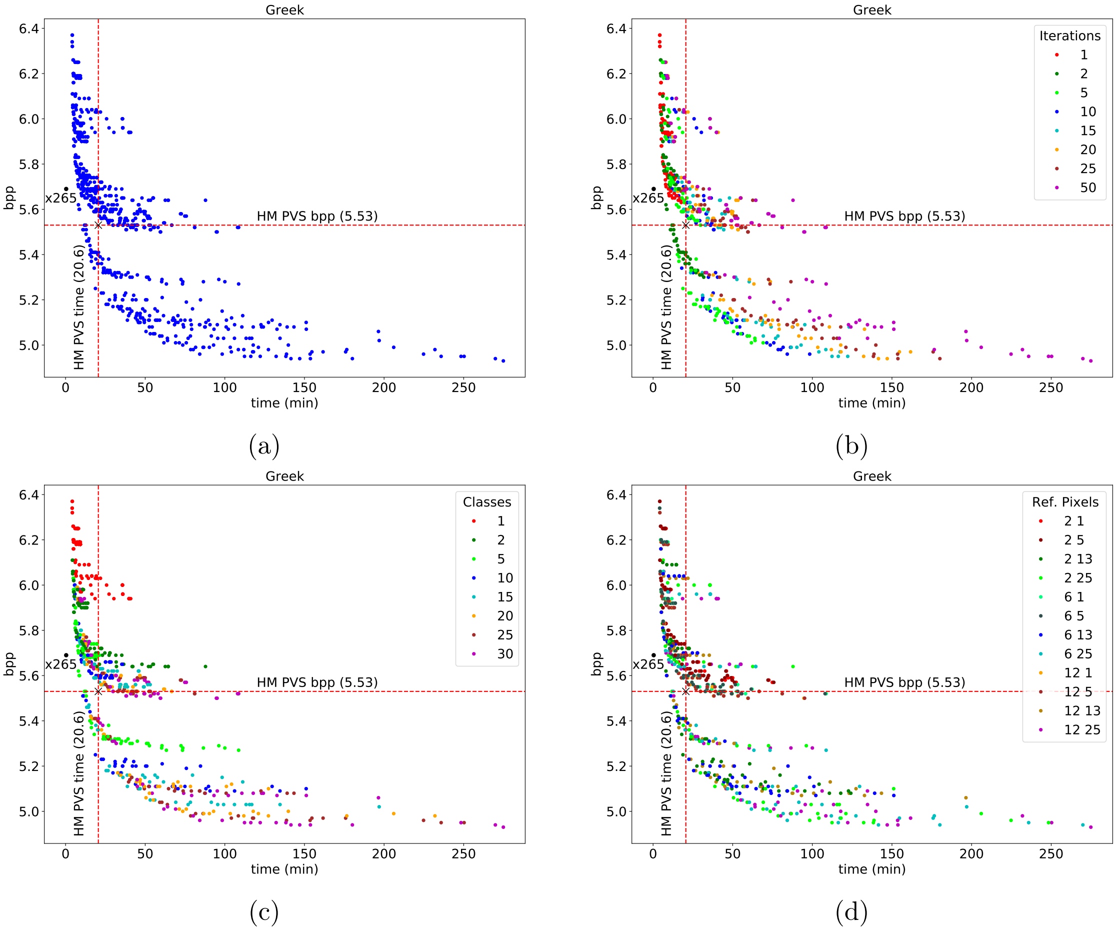
(c)

(d)

Figure 5.9: Operational points for Stone Pillars Outside image of EPFL dataset (a), (b) the operational points with the number of iterations sorted by colour, (c) the operational points with the number of classes sorted by colour, and (d) the operational points with the number of reference pixels pair (intra and inter) sorted by colour.

Figures 5.9a and 5.10a show all the resulting operational points when varying the previously described parameters. For an easier analysis in Figures 5.9b and 5.10b the number of iterations are represented by different colours, the same is done for the number of classes in Figures 5.9c and 5.10c, and for the number of reference pixels in Figures 5.9d and 5.10d. These figures show that with a proper selection of the coding parameters, M-MRP can achieve competitive coding efficiency, while maintaining a similar complexity to HM. The computational complexity was measured as the running time of the encoding operations for the codecs, using a dedicated server running Ubuntu Server 18.04 with an Intel Xeon Silver 4114 @ 2.20GHz CPU and 62.6 GB of RAM.

Considering the number of classes for both datasets, it is observed in Figures 5.9c and 5.10c that using just one class generally leads to higher bpp (*i.e.*, lower compression ratios). This was expected, as using just one class greatly reduces the adaptability of the encoder. For the remaining number of classes the analysis is not straightforward. However, considering the encoding time of



(a) (b)

(c) (d)

Figure 5.10: Operational points for Greek image of HCI dataset (a), (b) the operational points with the number of iterations sorted by colour, (c) the operational points with the number of classes sorted by colour, and (d) the operational points with the number of reference pixels pair (intra and inter) sorted by colour.

HEVC as reference, it can be seen that for both cases the lowest bpp (*i.e.*, highest compression efficiency) is obtained for 10 to 15 classes.

In regard to the number of iterations, the results show that using a lower number of iterations or classes (1 or 2), generally leads to lower compression efficiency, as can be seen for the Greek image. Once again, considering the encoding time of HEVC as a benchmark, using five iterations provide the highest compression efficiency for both images.

The colour representation of the parameters and visual analysis of the number of reference pixels does not result in sharp conclusions, unlike the case of the number of classes and the number of iterations. Nevertheless, the analysis of the results led to the selection of the encoding parameters shown in Table 5.4.

These M-MRP parameters are selected such that, for a similar complexity, M-MRP achieves higher coding efficiency than HM, originating the results shown in Tables 5.5 to 5.7, with bitrate and complexity results. In these tables, M-MRP stands for the encoding results with the selected parameters and M-MRP *Optimal* the highest compression ratio for M-MRP (non-constrained).

M-MRP *Optimal* represents the M-MRP results presented in Section 5.5.1.

Table 5.4: Parameter selection for the rate-complexity optimisation.

Dataset	Iterations	Classes	References[1]
EPFL	5	10	6, 5
HCI	5	15	6, 13

Table 5.5: Compression results (bpp) for the JPEG-Pleno CTC.

Type	Light Fields	HM	x265	JPEG-LS	M-MRP	M-MRP Optimal
LL	Bikes	13.69	14.59	16.42	11.55	11.47
	Danger	13.03	13.75	16.18	10.93	10.80
	Fountain	14.43	15.54	16.66	12.24	12.06
	Stone	13.44	14.10	16.77	11.30	11.20
HDCA	Greek	5.53	5.69	7.25	5.09	4.78
	Sideboard	8.16	8.75	12.19	7.99	7.62
Average		11.38	12.07	14.24	9.85	**9.66**

Table 5.6: Bitrate savings for the JPEG-Pleno CTC when compared to HM.

Type	Light Fields	x265	JPEG-LS	M-MRP	M-MRP Optimal
LL	Bikes	6.6%	19.9%	-15.6%	-16.2%
	Danger	5.5%	24.2%	-16.1%	-17.1%
	Fountain	7.7%	15.5%	-15.2%	-16.4%
	Stone	4.9%	24.8%	-16.0%	-16.6%
HDCA	Greek	2.9%	31.0%	-7.9%	-13.8%
	Sideboard	7.3%	49.4%	-2.1%	-6.5%
Average		6.1%	25.2%	-13.5%	**-15.2%**

The experimental results show that both M-MRP and M-MRP *Optimal* present better compression ratios than the other encoders. Although there is a slight loss of compression efficiency in M-MRP when compared to M-MRP *Optimal*, as expected, because the parameter values have been decreased to speed up the encoding time. Thus, M-MRP and M-MRP *Optimal* achieve a compression efficiency that is, on average, 13.5% and 15.2% higher than that of HM (HEVC), respectively.

In regard to the encoding and decoding time, Table 5.7 shows that the encoding time of M-MRP (*i.e.*, 105.5%) is quite similar to HEVC, with M-MRP being just 5.5% slower, on average (note that 100% corresponds to the encoding time of HEVC). This means that the encoding time of M-MRP *Optimal* can be reduced by 27 to 48 times, while still achieving an average compression efficiency gain of 13.5% in comparison with the HEVC Reference implementation (see M-MRP

[1] The references parameter represents the number of reference pixels in the current and in each of the neighbour SAIs.

Table 5.7: Encoding and decoding processing time relative to HM (in %) for the JPEG-Pleno CTC using the selected images.

Type	Light Fields	x265		JPEG-LS		M-MRP		M-MRP Optimal	
		Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
LL	Bikes	2.1%	94.8%	0.6%	150.2%	106.5%	85.6%	3202.4%	146.1%
	Danger	2.0%	95.2%	0.6%	155.5%	106.1%	87.5%	3205.7%	148.9%
	Fountain	2.0%	96.8%	0.6%	149.8%	101.2%	83.7%	3185.9%	143.5%
	Stone	2.1%	95.1%	0.6%	161.5%	102.4%	89.3%	4251.1%	152.8%
HCDA	Greek	1.9%	104.2%	0.7%	193.3%	114.6%	143.2%	4795.1%	307.8%
	Sideboard	1.5%	101.8%	0.7%	186.7%	109.8%	117.3%	2706.3%	271.2%
Average		2.0%	96.5%	0.6%	159.0%	105.5%	92.2%	3494.4%	166.7%

column in Table 5.6). Additionally, Table 5.7 shows that the decoding time of M-MRP is just 0.92 times that of HM, on average. In comparison with HM, the M-MRP *Optimal* average decoding time is 1.67 times higher, which shows that the change of the input parameters have a much lower influence on the decoding process (as expected).

Overall, the results presented in this section establish a trade-off relationship between LF lossless coding efficiency and computational complexity between a non-standard, yet more efficient encoder, and the standard HEVC Reference Encoder. It stands to reason, that these results can be extrapolated to other MRP based encoders, and in particular those presented in this chapter. Therefore, albeit Table 5.3 showing that DT-4D-MRP presents high computational complexity, this can be managed by a proper selection of input parameters.

5.6 Summary

In this chapter, three 4D prediction and partition modes were proposed for LF lossless encoding. These modes allow the encoder to use up to four neighbouring SAIs as prediction references, which leads to a better prediction performance and increases the coding efficiency of the MRP encoding, by exploiting the four-dimensional characteristics of LFs. The three proposals use the same type of 4D prediction, with differences relating to the type of tree partition: (i) in 4D-MRP an hexadeca tree is used, (ii) DT-4D-MRP uses a 4D partition with a quadtree for the spatial dimensions and other for the angular dimensions, and (iii) M-MRP uses a 2D quadtree partition. The 4D-MRP and M-MRP prediction modes present a good compromise in terms of rate-complexity trade-off.

Overall, in comparison with other state-of-the-art encoders, all the proposed methods present higher compression ratios, reducing the average bitrates from 10% to 32%. The highest performance is achieved by DT-4D-MRP which, to the best of the author's knowledge, is the best lossless coding performance for LFs represented in the 4D spatial-angular domain achieved so far. While its computational complexity might be a concern, some adjustments can be made to alleviate this issue. It was shown that the complexity of M-MRP can be reduced, while still maintaining better compression efficiency than other state-of-that-art encoders. Additionally, none of the implementations of the proposed methods were optimised for computational efficiency, which means there is room for improvement. Considering 4D-MRP and M-MRP, while they present lower performance than DT-4D-MRP, competitive compression ratios are nevertheless achieved and can be regarded as a good compromise in terms of rate-complexity trade-off.

CHAPTER 6

Hierarchical Minimum Rate Predictors

Contents

Current state-of-the-art methods, such as those proposed in the previous chapters, are able to efficiently encode light field (LF) images. However, most of the existing solutions lack useful functionalities, such as angular scalability and random access (RA). For LF applications that require lossless compression, such as medical imaging and critical computer vision tasks, the introduction of such capabilities in the encoder opens new possibilities. For instance, angular scalability, *i.e.*, the ability to progressively decode more sub-aperture images, allows the extraction of sub-sampled LFs. This can be useful for LF displays, that would be able to decode SAIs as needed, especially when constraints of storage space, network bandwidth, or even computational limitations are taken into account, and even improve the quality of SAI refocusing when additional information is decoded [120].

Another relevant functionality is random access, which consists in the ability of decoders to access specific regions of the LF without decoding the complete bitstream. This simplifies the navigation through the LF sub-aperture images, reduces the decoding delay and the required computational resources at the decoder. RA may also play a role in the analysis of particular views by physicians in a clinical context, in order to quickly render specific regions, *e.g.*, SAI, for preliminary medical assessment.

This chapter proposes a hierarchical approach based on the Minimum Rate Predictors (MRP) encoding method, that is capable of providing angular scalability and random access functionalities for the lossless compression of LFs. This new encoder, named as Hierarchical Minimum Rate Predictors (H-MRP), allows flexible definition of various hierarchical layers that, together with constraints to the encoding dependencies, can provide RA capabilities. This prediction structure is a contribution to endow conventional MRP lossless encoding schemes [62] with angular scalability and random access functionalities.

The remainder of this chapter is organised as follows: Section 6.1 overviews current solutions for angular scalability and random access in LF coding; Section 6.2 describes the proposed encoder [J1]; in Section 6.3 the experimental results are presented and analysed; and finally, Section 6.4 summarises the chapter.

6.1 Related Work

In this section, a review on state-of-the-art LF encoding methods with angular scalability and RA is presented. Regarding lossless encoding, the only work found in the literature that supports scalability and RA is described in [103]. This work describes a LF codec supporting quality scalability from lossy to lossless. Details about this algorithm were already exposed in Section 3.3. It is worth mentioning that this work only uses non-rectified SAIs, *i.e.*, not obtained through the JPEG-Pleno processing chain and, therefore, requires both extra information and processing steps at the decoder side.

As the literature on lossless LF encoders with RA and angular scalability is rather limited, the remainder of this review refers only to research on lossy encoders. The methods presented below are divided into two categories: those supporting angular scalability and those supporting random access. Some works provide both capabilities and so belong in both categories.

In regard to angular scalability, in [121], Conti *et al.* propose to address the compatibility of LFs with legacy displays. The authors use a three-layer approach, where the base layer encodes one or more SAIs using standard 2D encoders. The first enhancement layer encodes the SAIs in a stereo or multiview fashion using prediction references from the same layer and also from the previous one. Finally, the second enhancement layer encodes the remaining information by exploiting the spatial and geometric multiview information, and the micro images (MIs) LF information, taking advantage of the overall redundancy. This work was extended in [122], where the number of enhancement layers is flexible, and extra inter-layer coding tools are proposed. In [123], Yun Li *et al.*, propose to encode focused LFs by using a sparse set of micro images and their calculated disparities. This information is used to interpolate the remaining LF, using inpainting where necessary. This interpolated LF is used as the prediction for the original one. The selected MIs are encoded with a modified High Efficiency Video Coding (HEVC) encoder, as well as four disparity maps, using lossless coding. The sparse set of micro images act as the first layer, with the interpolated LF being the second and the third the residues resulting from the previous interpolation. This work is expanded in [124] resorting to lossy encoding of the disparity maps.

Although the previous works provide some form of angular scalability, most of them only use two or three layers, which still poses limitations. Thus, a more flexible approach capable of dealing with a variable the number of layers will be useful for enlarging the scope of possible applications, and in particular, to support a larger range of displays resolutions. The H-MRP algorithm proposed in this chapter, presents a contribution to alleviate this problem by allowing the user to flexibly configure the number of scalability layers and the inter-layer prediction structure.

Concerning random access, *i.e.*, the ability to decode selected parts of the LF without decoding the whole information, several methods can be found in the literature and some of these also present scalability capabilities. In [125], the SAIs are divided into 25 groups and a fixed dependency scheme is defined between the groups. A new pseudo-sequence-based prediction structure is proposed and applied to the Multiview High Efficiency Video Coding (MV-HEVC) in [126]. In [127], it is proposed to interpret the LF as multiview sequences, by using a quadratic spiral scan to form multiple pseudo-video sequences (PVSs). The result is encoded with MV-HEVC, where several dependency layers are tested to achieve better RA capabilities. In [128], Pratapa proposes a hierarchical scheme to encode LFs, where the top level layers encompass the merged information of lower level layers, which is achieved by computing new SAIs through lower levels SAIs clustering. These new SAIs are called representative key views (RKVs). This operation is performed at each level until only one RKV exists in the last level. The top level RKVs are encoded with JPEG2000 and the lower level SAIs are encoded using RKVs in higher levels as reference. The bitstream is organised to enable RA and progressive decoding of the pixels.

In [129], a study on the performance of different dependency structures for LFs in MV-HEVC is carried out. Finally, in [130] a LF coding framework with flexible viewpoint scalability and random access is proposed. It encodes the light field using a PVS representation, with enhanced reference picture selection algorithms. The angular scalability is achieved by defining six layers, each one with an increasing number of SAIs, that define the coding order. Random access is achieved by setting the size of the reference picture list, the number of viewpoint regions, and the maximum dependency layer – *i.e.*, the scalability layers that can be used as reference. In order to further improve the RA capabilities, the authors group the LF SAIs to create regions that are self contained in terms of prediction references called viewpoint regions, *i.e.*, that can be decoded independently.

A common limitation of most works presented above is the low RA flexibility to cope with different requirements imposed by the rigid prediction dependencies, and the fact that they are not designed to cope with lossless compression. The present work addresses these limitations by proposing a hierarchical lossless LF encoder with flexible random access capabilities.

6.2 Hierarchical Lossless Encoding with MRP

The proposed H-MRP algorithm is a versatile encoder that allows to finely tune the encoding parameters with the goal of achieving high compression efficiency, while also adding enhanced functionalities to the bitstream, such as random access and scalability. H-MRP relies on a hierarchical decomposition of the LF representation data, which organises the LF sub-aperture images into multiple layers defined by the configuration parameters. The most preeminent aspect of the encoder is the hierarchical coding structure defined for inter SAI prediction, where the SAIs in each layer are encoded in succession, resorting to reference SAIs that lie either in the same layer or in lower ones within the hierarchical structure. Two other key aspects for increasing the H-MRP versatility are (i) the selection of the reference SAIs, which allows the encoder to use the most appropriate SAIs as references; and (ii) the disparity compensation, which is especially relevant to minimise the disparity of SAI references in lower layers – that are farther from the current SAI in the LF 4D space. The encoder uses a set of linear predictors adapted to the SAIs characteristics to provide high compression rates.

6.2.1 The H-MRP Encoding Scheme

An overview of the processing flow of the proposed Hierarchical Minimum Rate Predictors is provided in Figure 6.1. The H-MRP algorithm starts by pre-processing the LF by applying an RCT and histogram packing, as was done in the case of Four-Dimensional Minimum Rate Predictors (4D-MRP) in Chapter 5, for instance. This step ensures that the LF colour components are decorrelated, thus increasing the compression efficiency. After the pre-processing step, the SAI to be encoded is selected according to the sequence order defined by the hierarchical structure. The encoder compresses each hierarchical layer separately. Further details about the hierarchical coding structure are given in Section 6.2.2. Then, the reference SAIs for the prediction are selected based on their distance to the current SAI in the LF 4D space. This process, which allows for the encoder to use the most appropriate prediction references, is described in Section 6.2.3. The selection of reference SAIs might also depend on the random access region (RAR) partitions, with the 4D LF divided into these RARs, such that each SAI can only be predicted with references from the same region. Section 6.2.4 provides details of the RARs and its potential influence on H-MRP performance. Finally, before starting the MRP encoding steps, disparity compensation is applied to the reference SAIs. For each of the selected references, the disparity vectors are calculated and used in the disparity compensated prediction, as detailed in Section 6.2.5.

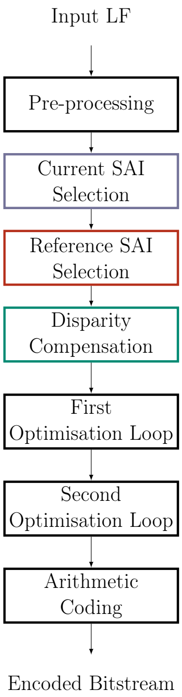


Figure 6.1: Functional diagram of the H-MRP algorithm.

After the *Disparity Compensation* module, the encoding algorithm follows the MRP steps, as presented in Algorithm 6.1. Overall, this algorithm is based on the optimised computation of linear predictors, each associated to a corresponding class. These classes are characterised by a set of coefficients computed to minimise the bitrate needed to encode the prediction residuals and side information. Thus, as defined in Algorithm 6.1, the processing flow is organised as a sequence of optimisation loops: one using blocks of fixed size to perform the initial operations and then another one using variable block sizes that refines the computations and optimisation of the previous operations, in the same way as described in Section 3.4.

First Optimisation Loop: The first optimisation loop, which divides the SAIs into blocks of 8×8 pixels, starts by sorting the blocks into M classes, according to its pixels variances, providing an initialisation for the algorithm. The pixels in these blocks are predicted using a class-specific linear predictor. The prediction in H-MRP is similar to that of the MRP based encoders, where the prediction of a pixel p_0 is computed as a weighted sum of pixels from the reference SAIs, as defined by Equation 6.1,

$$\hat{s}(0) = \sum_{i=0}^{N_R} \sum_{k=1}^{K_i} a_m(i,k) \cdot s(i,k), \tag{6.1}$$

where $s(i,k)$ is the k-th pixel value for the current (for $i = 0$) or the reference SAIs (for $i = 1, \dots, N_R$), N_R is the number of reference SAIs, and $a_m(i,k)$ is the coefficient of the m-th class linear prediction models for the current and reference SAIs. The reference pixels from the current and reference SAI are illustrated in Figure 6.2. The coefficients a_m are calculated in the *Predictors*

Algorithm 6.1 Hierarchical MRP high level algorithm.

```
Input: light field
Apply pre-processing
for layer: 1 to MAX_LAYER do
    for each SAI in layer do
        Reference SAIs selection ($N_R$)
        Calculate disparity vectors for each reference ($\overrightarrow{D}_\chi$)
#Initial block classification
        Sort and classify of $8 \times 8$ blocks by pixel variance
#1st optimisation loop using fixed block size
        for 1 to MAX_ITERATIONS do
            #Predictors design
            for each class do
                Compute the prediction coefficients ($a_m$)
                Calculate residuals
            Calculate prediction residue encoding cost ($B_r$)
#Groups optimisation
            for each class do
                Compute $C$ quantisation thresholds to minimise $B_r$
            Calculate $B_r$
#Classification optimisation
            for each $8 \times 8$ block do
                Move neighbour blocks classes to front of list
                Calculate residuals for all classes
                Select class that minimises $B_r$
            Calculate $B_r$
            if 10 iterations without improvement then
                end for loop
#2nd optimisation loop using variable block size
        for 1 to MAX_ITERATIONS do
            Calculate prediction coefficients encoding cost ($B_a$)
#Groups optimisation
            for each class do
                Compute $C$ quantisation thresholds to minimise $B_r$
                Update shape parameter in probability models
            Calculate quantisation thresholds cost ($B_t$)
#Classification optimisation using VBS
            for Each $32 \times 32$ block do
                ClassOptimisation
            Calculate cost $J$
            if 10 iterations without improvement then
                end for loop
        Remove non-utilised classes
        Run arithmetic coding
Output: Encoded bitstream
```

Design step, by solving a set of Yule-Walker equations, using the pixel correlations considering all the pixels in the blocks associated with class m. K_i represents the number of pixels that support the prediction in the current (for $i = 0$) and reference SAIs (for $i = 1, \ldots, N_R$), respectively.

Algorithm 6.2 Classification optimisation procedure.

```
procedure ClassOptimisation
    Move neighbouring blocks classes to front of list
    Calculate residuals
    Select class that minimises $B_r$
    if level > 0 then
        Calculate cost of not partitioning block ($J_1$)
        Partition block in quadtree fashion
        for each resulting block do
            ClassOptimisation
        Calculate sum of cost of partitioned blocks ($J_2$)
        if $J_2 < J_1$ then
            Partition block
    Return: Cost, partition structure and class selection
```

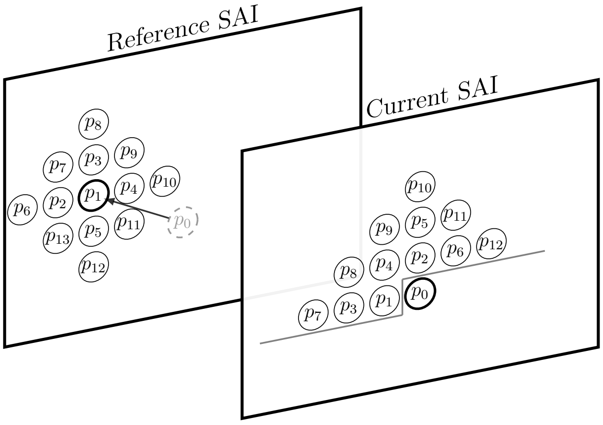


Figure 6.2: H-MRP prediction using disparity vectors.

Then, in the *Groups Optimisation* step, the algorithm calculates a set of thresholds, usually 15, to be used for quantisation of the prediction residuals context. The quantised contexts, resulting from the thresholds, define 16 groups, each with a different probability model, defined by the parameters of a Gaussian probability density function, for the arithmetic coding of the prediction residuals. The context C is calculated as:

$$C = \sum_{i=0}^{N_R} \sum_{k=1}^{K_i} \frac{1}{\delta(i,k)} e(i,k), \tag{6.2}$$

where $e(i,k)$ represents the prediction error of the reference pixel p_k in the reference SAI i, given by $e(i,k) = |s(i,k) - \hat{s}(i,k)|$ and $\delta(i,k)$ are weighting factors proportional to the Euclidean distance between the current pixel (p_0) and each reference pixel (k). The $\delta(i,k)$ weights are defined as:

$$\delta(i,k) = \frac{\sqrt{d_v(k)^2 + d_u(k)^2 + \lambda(i)^2}}{64}, \quad \text{with} \quad \lambda(i) = \begin{cases} 0, & i = 0 \\ 1, & i \neq 0 \end{cases}, \tag{6.3}$$

where d_v and d_u are the spatial distances, measured in pixels, between a reference pixel p_k and the pixel to encode. $\lambda(i)$ represents the distance between SAIs and is set to 1 for the reference

SAIs, and set to 0 for the current SAI.

The first optimisation loop is concluded by the *Classification Optimisation* step, which refines the initial block classification into the newly calculated linear models of the classes. Instead of directly transmitting the actual class index for each block, the encoder uses a lookup table that is updated as each block is processed by using a move-to-front method. The class index associated with the upper, left, and upper right neighbour blocks are placed at the top of the lookup table, in this order. In this manner, as neighbour blocks are correlated, the indices to transmit should generally have small values, which leads to a more efficient compression of these indices (*i.e.*, different classes can be represented by the same lookup table index). Then, the prediction residuals of each block are computed using all the predictors to find the associated class that minimises the encoding cost B_r of prediction residuals, calculated as

$$B_r = \sum_{p_0} L\left(e|\hat{s}\left(0\right), n\right), \tag{6.4}$$

where $L(e|\hat{s}(0), n)$ represents the encoding cost of the prediction error, which is computed by using the probability density estimate of the prediction errors belonging to the n-th group, assumed to follow a Gaussian distribution. The first optimisation loop ends when the algorithm reaches the maximum number of iterations, or 10 consecutive iterations without improvement.

Second Optimisation Loop: The second optimisation loop refines the operations of the previous loop using blocks with variable sizes (VBS), and fits the probability models to the prediction errors. This optimisation takes into account the cost of encoding both the signalling and the prediction errors:

$$J = B_a + B_m + B_t + B_r, \tag{6.5}$$

where B_a, B_m, and B_t are the encoding costs of the prediction coefficients, class selection, and context modelling threshold values, respectively. For every iteration where J is calculated, all its sub costs B are recalculated. In the previous loop, at the *Groups Optimisation* step, only the threshold for the quantisation of C was optimised. In this phase, the algorithm also updates the shape parameter of the generalised Gaussian probability density functions, which are used to model the prediction residuals of each group. Then, in the *Classification Optimisation* step, the H-MRP divides the SAI into blocks that are partitioned in a quadtree fashion, selecting for each block the class whose linear predictions models were designed in the first loop, and that minimises the encoding cost. The initial block size in the variable block size (VBS) optimisation is 32 × 32 pixels, which can be partitioned down to blocks with 2 × 2 pixels. The block sizes define quadtree levels ranging from 4 to 0. For each quadtree level, the decision whether a block is to be partitioned depends on whether or not the sum of the sub-blocks costs (J_2) is smaller than the cost of the parent non partitioned block (J_1). The class choice is signalled by the same process described in the previous optimisation loop.

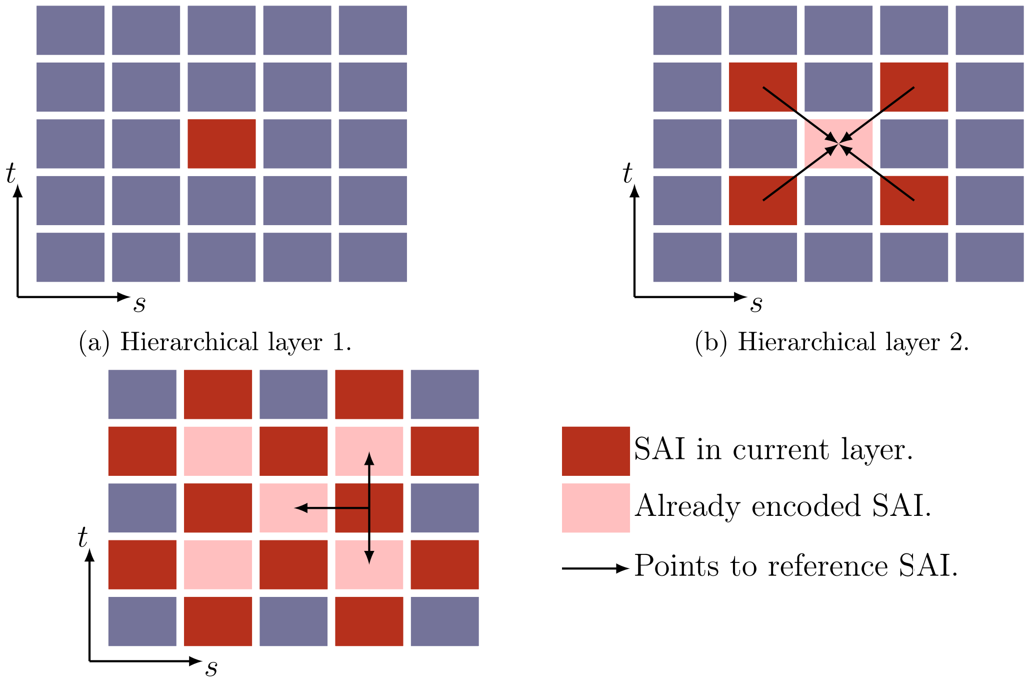
The encoding process is finalised by coding all the generated information using a range coder [68].

More details about the encoding process of the MRP family algorithms can be found in [62, 68].

6.2.2 Hierarchical Coding Structure

For hierarchical encoding, the input data is divided into layers where the predictions only depend on information that is available in lower hierarchical levels. Generally, this concept is used to progressively add information, for example, by increasing resolution or adding more frames (or SAIs) when encoding/decoding layers that are higher in the hierarchy. For instance, an encoder might compress a sub-sampled version of an image in the lower layer, and the difference to the full resolution, in the higher level layer. In some applications, this allows the decoder to decode a sub-sampled version without decoding the remaining full image.

The hierarchical architecture of H-MRP (which enables the angular scalability) is attained by allocating different SAIs of the LF to each hierarchical layer. Figure 6.3 presents a possible selection of SAI references in H-MRP. This figure presents hierarchical layers of a LF with 5 × 5 SAIs, where SAIs in red belong to the respective hierarchical layer, those in pink represent previously encoded SAIs, and the blue ones are the SAIs in higher hierarchical layers. The arrows in the figure point to the reference SAIs. As can be seen in this figure, layers at lower hierarchical levels (*e.g.*, Hierarchical layer 1) contain less information (SAIs in this case) than those at higher levels (*e.g.*, Hierarchical layer 3).



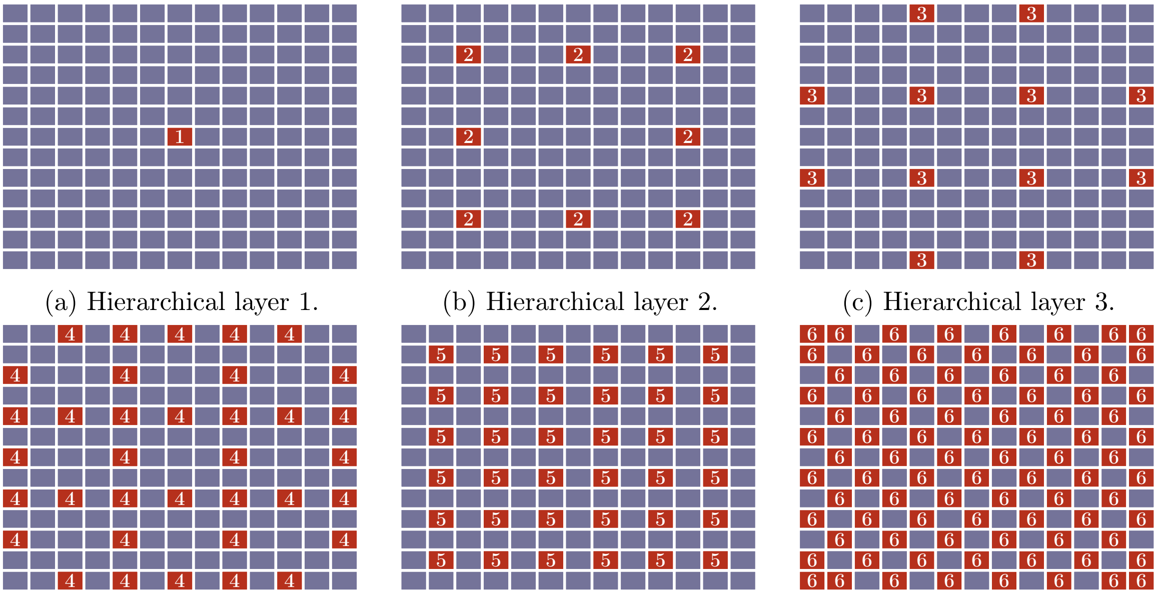
(a) Hierarchical layer 1.

(b) Hierarchical layer 2.

(c) Hierarchical layer 3.

Figure 6.3: Example of a possible hierarchical coding scheme of H-MRP: (a) Hierarchical layer 1 containing a single SAI; (b) Hierarchical layer 2 containing four SAIs, all using the same reference; and (c) Hierarchical layer 3 containing twelve SAIs, showing a potential selection of three references.

The hierarchical structure used in this thesis for H-MRP and the SAIs allocation to each of the six layers, for a 15 × 15 LF, are shown in Figure 6.4. The lower level hierarchical layer has only one SAI, the second layer has eight SAIs placed around the SAI in the first layer, *etc.* If the layers are progressively decoded, the partial output resulting from the decoding of the first layer can be interpreted as a 1×1 ($t \times s$) LF. Then, the partial output of the second layer decomposition will be interpreted as a 3×3 LF. Depending on the SAIs allocation to each hierarchical layer, the decoder can construct a progressively larger 4D LF, that can be considered a sub-sampled version of the full 4D LF.



(a) Hierarchical layer 1.

(b) Hierarchical layer 2.

(c) Hierarchical layer 3.

(d) Hierarchical layer 4.

(e) Hierarchical layer 5.

(f) Hierarchical layer 6.

Figure 6.4: Representation of the used hierarchical layers (inspired by [78], for the case of 13×13 LFs).

In order to achieve random access in a hierarchical scheme, it must be possible to decode each layer independently. Alternatively, as in the case of H-MRP, information from lower hierarchical levels might also be used as references for the current one, establishing some level of dependency. A trade-off between compression efficiency and random access capabilities is achieved, depending on whether SAIs from the same layer are used as reference. H-MRP uses up to N_R reference sub-aperture images as defined by Equation 6.1. The random access capabilities of H-MRP allow easier navigation through the light field, enabling any SAI to be decoded without the need to decode the whole compressed stream. However, the compression efficiency might be penalised, because it is not guaranteed that the optimal references are always chosen when there are restrictions to the reference SAIs selection. Additionally, SAIs in higher hierarchical layers have larger decoding costs in a random access sense, since more SAIs need to be decoded to be used as references.

The coding structure proposed in this work allows for a versatile allocation of the SAIs to the various hierarchical layers and also the possibility of defining the scanning order of the SAIs in each layer.

6.2.3 Selection of Reference SAIs

The proper selection of the reference SAIs for the prediction is an important aspect due to its impact in the prediction efficiency. Therefore, in H-MRP, a flexible structure to select the reference SAIs is adopted. Such flexible structure is attained by adding previously encoded SAIs of lower layers to a list of candidate references[1]. Then, the euclidean distance d between the candidate references and the current SAI is calculated:

$$d = \sqrt{(t_r - t_c)^2 + (s_r - s_c)^2}, \tag{6.6}$$

where (t_r, s_r) and (t_c, s_c) represent the angular coordinates of the reference and current SAIs. The list is sorted by increasing distance and all candidate SAIs whose distances are greater than M_d (set to 8 in this work) are removed, to discard references that are too distant from the SAI to be encoded. From the remaining SAIs, up to N_R references are selected.

6.2.4 Random Access Regions

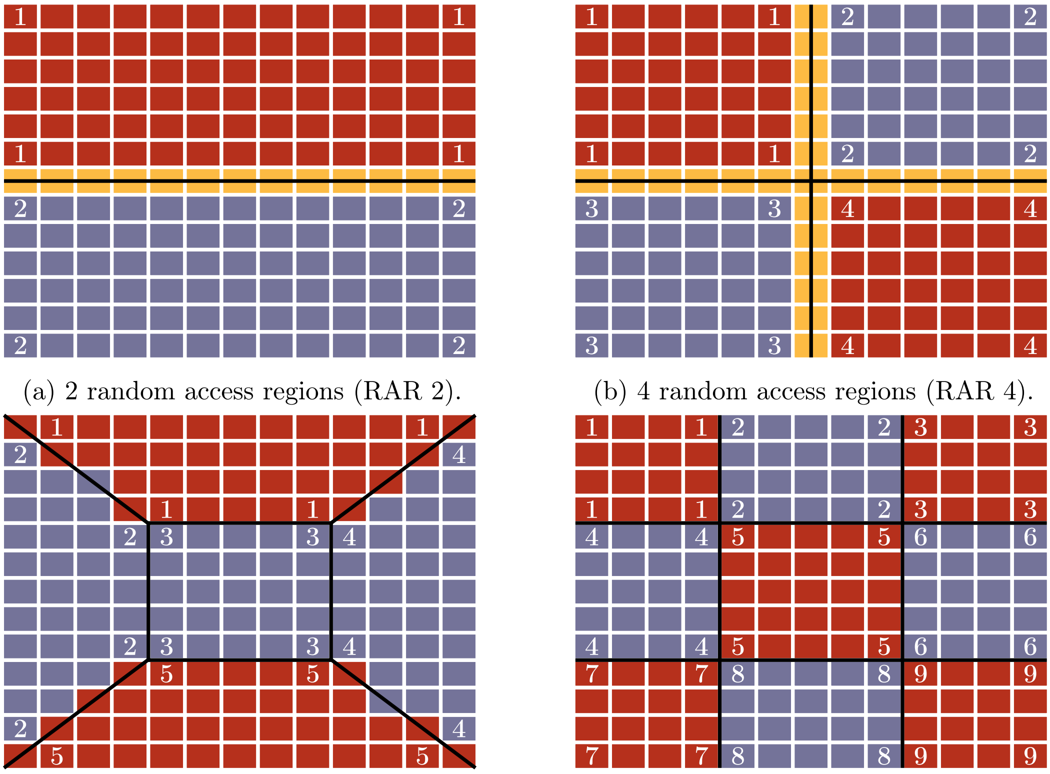
To improve the random access capabilities of H-MRP, random access region are introduced. This concept is inspired by the viewpoint regions of [130]. The random access regions define limits for the reference SAIs selection, such that an SAI in any given RAR can only use references of the same region. Therefore each random access region can be encoded and decoded independently, without requiring information from other regions. This restriction of references is useful not only for improving the RA capabilities, since references will necessarily be closer to the SAI to encode, but also to facilitate navigation through the neighbouring SAIs. Figure 6.5 shows the RAR divisions used in H-MRP, where the black lines indicates the regions borders and the SAIs in yellow are shared between the regions connected to it, *i.e.*, belong to all neighbouring regions.

These RAR divisions, when combined with the hierarchical coding structure, described in Section 6.2.2, force the encoder to have, at least, one referenceless SAI in each region, *i.e.*, an intra SAI. Thus increasing the number of intra SAIs, slightly reducing the compression efficiency while improving the random access capabilities.

6.2.5 Disparity Compensation

Due to the characteristics of LFs, the visual information in the SAIs, which represent slightly different viewpoints of the scene, is not completely aligned, owing to the inter-SAI disparity. The disparity between an SAI and its references should be taken into account in the encoder, to improve the quality of the prediction. Thus, disparity compensation is beneficial for the encoding process. Considering the 4D representation of LFs, denoted in Equation 2.2 and Equation 2.4,

[1] SAIs of the same level can also be added to the list if this option is enabled, as stated in the previous section.



(a) 2 random access regions (RAR 2).

(b) 4 random access regions (RAR 4).

(c) 5 random access regions (RAR 5).

(d) 9 random access regions (RAR 9).

Figure 6.5: Random access region divisions used in this work, as seen in [130].

and taking into account the disparity, a disparity compensated prediction SAI can be formed from a reference SAI as:

$$\widetilde{\mathcal{L}}^{t,s}(v,u) = \mathcal{L}^{t,s}(v - \mathcal{D}_{\mathcal{V}}(v,u), u - \mathcal{D}_{\mathcal{H}}(v,u)), \tag{6.7}$$

where $\mathcal{D}_{\mathcal{V}}(v,u)$ and $\mathcal{D}_{\mathcal{H}}(v,u)$ represent the vertical and horizontal disparity, respectively. Consequently, the disparity vector of a given sub-aperture image, χ, is given by:

$$\overrightarrow{D}_{\chi} = (\mathcal{D}_{\mathcal{V}}(v,u), \mathcal{D}_{\mathcal{H}}(v,u))\,. \tag{6.8}$$

The disparity is estimated on a block-wise basis, for each of the reference SAIs. The search range for the disparity is limited to M_D, for blocks of size B_D, such that $|\mathcal{D}_{\mathcal{V}}(v,u)|$, $|\mathcal{D}_{\mathcal{H}}(v,u)| \in [0,...,M_D]$. Based on empirical experiments, in this work these values were set to $B_D = 8$ and $M_D = 8$. The appropriate vector, $\overrightarrow{D}_{\chi}$, is selected to minimise the sum of squared differences:

$$SSD = \sum_{i=1}^{B_D} \sum_{j=1}^{B_D} \left[R(i + \mathcal{D}_{\mathcal{V}}(v,u), j + \mathcal{D}_{\mathcal{H}}(v,u)) - C(i,j)\right]^2, \tag{6.9}$$

where $R(i,j)$ and $C(i,j)$ represent the pixel value at position (i,j) for the reference and current SAIs, respectively. These disparity vectors define the displacement of the prediction template on the reference sub-aperture images relative to the co-located position, as shown in Figure 6.2.

The disparity vectors are encoded using prediction and arithmetic coding with an adaptive probability model. Prior the encoding, the vectors are transformed by subtracting the median of the disparity vectors of neighbouring blocks, as indicated in Figure 6.6. The difference vectors resulting from the prediction are given by:

$$\vec{R}_{\vec{D}_{\chi}} = \vec{D}_{\chi} - med\left(\vec{D}_A, \vec{D}_B, \vec{D}_C\right). \tag{6.10}$$

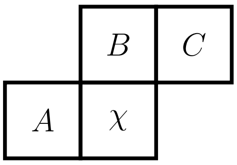


Figure 6.6: Vector prediction.

6.3 Experimental Evaluation

A number of experiments were conducted to characterise the performance of H-MRP and its dependence on all the parameters and prediction structures. The analysis of the experimental results, obtained with the proposed method, is divided in three parts. In the first part (Section 6.3.1) the influence of the H-MRP input parameters on its compression efficiency is assessed. This study defines two configurations for H-MRP which are used in a comparison with state-of-the-art encoders in the second part (Section 6.3.2): (i) maximum compression efficiency and (ii) minimum random access penalty (RAP). The third, and final, part (Section 6.3.3) of this analysis studies the influence of the RAR, mainly concerning in improving the encoder RA capabilities.

6.3.1 Empirical Studies on the Parameters Influence

A set of experiments were performed to assess the influence of the available options on both the compression efficiency and random access capabilities. This section reports on these experimental results, with the objective of finding the configurations that maximise the coding efficiency and also provide the highest RA capabilities, *i.e.*, low RAP. The number of reference pixels used in the linear prediction models (*i.e.*, H-MRP classes) was empirically determined. The following values were found to provide the best performance for each dataset:

- École Polytechnique Fédérale de Lausanne (EPFL): $K_i = 12$ pixels for the current SAI and $K_i = 13$ pixels per reference SAI.
- Heidelberg Collaboratory for Image Processing (HCI): $K_i = 6$ pixels for the current SAI and $K_i = 25$ pixels per reference SAI.

- Light field image dataset of skin lesions (SKINL2): $K_i = 12$ pixels for the current SAI and $K_i = 25$ pixels per reference SAI.

The experiments described in this subsection were designed to evaluate the influence of five different encoding configuration parameters on the performance of H-MRP (in terms of compression efficiency and RAP):

- the number of reference sub-aperture images (N_R: one, two, or four);
- the number of hierarchical layers (H_L: ranging from two to six, for which the configurations are shown in Figure 6.4);
- the use of disparity compensation, identified as: With (W/) or Without (W/o) DC;
- the use of references in the same hierarchical layer, identified as: With or Without Same Layer;
- the scan order, when using references in the same layer (Raster or Spiral).

The influence of the SAI scanning order was only evaluated considering references in the same hierarchical layer, as the scanning order does not influence the compression process otherwise. This is due to the fact that each SAI is independently encoded, using only SAIs selected from lower hierarchical layers, independently of the scanning order. These configuration parameters are listed in Table 6.1.

Table 6.1: Summary of the configurations used for the H-MRP experiments.

Configuration Parameters	Options
Maximum number of references (N_R)	1, 2, or 4
Number of hierarchical layers (H_L)	2, 3, 4, 5, or 6
Disparity compensation (DC)	With or Without
Same layer references	With or Without
Scan order	Raster or Spiral

The results of the parameter optimisation studies are shown in Table 6.2 for the EPFL dataset, in Table 6.3 for the HCI dataset, and in Table 6.4 for the SKINL2 dataset. In these tables, *A. BPP* represents the average bits-per-pixel (bpp) needed to encode the LFs, and *M. RAP* represents the average maximum RAP value. The maximum RAP is calculated for the SAI that needs more bits to be decoded, including the cost of decoding the reference SAIs. N_R represents the number of SAI references, DC indicates the use of disparity compensation, and H_L the number of hierarchical layers. For example, H_L 3, means three layers are used: layer 1 and 2 as seen in Figure 6.4 (a) and (b), and a third layer containing all the remaining SAIs. For both bpp and RAP, lower values indicate better performance. The best results in each column are highlighted in bold.

Table 6.2: Parameter optimisation for the EPFL dataset.

		W/o Same Layer				W/ Same Layer Raster Scan				W/ Same Layer Spiral Scan			
		W/o DC		W/ DC		W/o DC		W/ DC		W/o DC		W/ DC	
N_R	H_L	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP
1	2	14.64	**0.01**	14.85	**0.01**	12.85	0.15	12.81	0.14	12.53	0.08	12.73	0.08
	3	13.72	0.02	14.03	0.02	12.95	0.16	12.92	0.16	12.64	0.06	12.84	0.06
	4	13.35	0.03	13.59	0.03	13.09	0.10	13.07	0.10	12.78	**0.04**	12.99	**0.04**
	5	13.24	0.03	13.48	0.03	13.12	**0.04**	13.27	**0.04**	13.00	**0.04**	13.23	**0.04**
	6	13.23	0.04	13.48	0.04	13.22	**0.04**	13.48	**0.04**	13.21	**0.04**	13.46	**0.04**
2	2	-	-	-	-	11.85	1.00	**12.02**	1.00	11.70	1.00	12.00	1.00
	3	13.47	0.03	14.11	0.03	11.93	1.00	12.13	1.00	11.80	1.00	12.12	1.00
	4	12.83	0.04	13.40	0.04	12.01	0.73	12.24	0.73	11.93	0.29	12.27	0.28
	5	12.02	0.07	12.48	0.06	11.95	0.26	12.37	0.26	11.98	0.15	12.41	0.15
	6	12.48	0.07	12.90	0.07	12.46	0.11	12.90	0.11	12.44	0.11	12.88	0.11
4	2	-	-	-	-	11.37	1.00	11.98	1.00	**11.19**	1.00	**11.93**	1.00
	3	13.44	0.03	14.68	0.03	11.41	1.00	12.04	1.00	11.25	1.00	11.99	1.00
	4	12.62	0.07	13.84	0.07	11.46	0.94	12.11	0.94	11.30	0.80	12.07	0.80
	5	11.65	0.12	12.67	0.12	11.32	0.94	12.15	0.94	11.30	0.87	12.12	0.86
	6	**11.26**	0.14	**12.11**	0.13	**11.22**	0.24	12.06	0.24	11.21	0.19	12.05	0.19

Table 6.3: Parameter optimisation for the HCI dataset.

		W/o Same Layer				W/ Same Layer Raster Scan				W/ Same Layer Spiral Scan			
		W/o DC		W/ DC		W/o DC		W/ DC		W/o DC		W/ DC	
N_R	H_L	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP
1	2	8.12	**0.03**	7.50	**0.03**	6.65	0.23	6.82	0.23	6.63	0.11	6.80	0.11
	3	7.31	0.04	7.16	0.04	6.80	0.15	6.88	0.15	6.80	0.10	6.86	0.10
	4	7.10	0.04	7.04	0.04	6.88	0.14	6.91	0.14	6.88	0.08	6.89	0.08
	5	7.10	0.06	7.05	0.05	6.97	**0.07**	6.95	**0.07**	6.96	0.07	6.93	0.06
	6	7.10	0.07	7.04	0.07	7.09	**0.07**	7.02	**0.07**	7.08	**0.05**	7.01	**0.05**
2	2	-	-	-	-	**6.32**	1.00	**6.62**	1.00	**6.29**	1.00	**6.60**	1.00
	3	7.11	0.06	7.06	0.05	6.45	0.82	6.65	0.82	6.47	0.51	6.66	0.50
	4	6.87	0.07	6.93	0.07	6.54	0.78	6.69	0.77	6.55	0.41	6.69	0.40
	5	6.72	0.10	**6.82**	0.09	6.57	0.34	6.69	0.33	6.61	0.24	6.71	0.23
	6	6.76	0.13	6.84	0.12	6.73	0.19	6.81	0.17	6.73	0.19	6.81	0.17
4	2	-	-	-	-	6.35	1.00	6.96	1.00	6.31	1.00	6.93	1.00
	3	7.20	0.07	7.43	0.06	6.45	0.92	6.96	0.92	6.46	0.75	6.95	0.75
	4	6.85	0.13	7.20	0.12	6.52	0.92	6.97	0.92	6.51	0.75	6.95	0.75
	5	6.67	0.24	7.08	0.22	6.50	0.92	6.92	0.92	6.50	0.83	6.91	0.82
	6	**6.55**	0.27	6.93	0.25	6.52	0.29	6.91	0.28	6.52	0.28	6.91	0.26

Experiments Without Same-Layer References

The analysis of these results can be divided into two cases, in terms of the use of reference SAIs in the same layer. The case where SAIs on the same layer are not used as reference is addressed first. The analysis of the results is divided by the H-MRP configuration parameters. Figure 6.7 presents these results in a graphical format for better visualisation of the ensuing analysis.

Number of hierarchical layers: The results show that the compression efficiency increases when more hierarchical layers are used. This observation generically holds for all datasets, irrespective of the number of reference SAIs and the use of disparity compensation. Taking into account the hierarchical layers shown in Figure 6.4, it is obvious that when more layers are used, the list of potential reference SAIs increases. A consequence of this is that closer SAIs, as defined by Equation 6.6, are available to be used. In the same way, the average of the maximum RAP increases when more hierarchical layers are used, as more SAIs may need to be decoded

Table 6.4: Parameter optimisation for the SKINL2 dataset.

		W/o Same Layer				W/ Same Layer Raster Scan				W/ Same Layer Spiral Scan			
		W/o DC		W/ DC		W/o DC		W/ DC		W/o DC		W/ DC	
N_R	H_L	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP	A. BPP	M. RAP
1	2	9.49	**0.03**	7.95	**0.03**	7.21	0.23	7.20	0.23	7.14	0.12	7.15	0.12
	3	8.41	0.04	7.59	0.04	7.44	0.16	7.27	0.15	7.40	0.10	7.24	0.10
	4	7.96	0.04	7.46	0.04	7.56	0.14	7.31	0.14	7.53	0.07	7.28	0.08
	5	7.99	0.06	7.45	0.06	7.76	**0.07**	7.36	**0.07**	7.73	0.07	7.34	0.07
	6	7.99	0.07	7.44	0.07	7.98	**0.07**	7.43	**0.07**	7.96	**0.06**	7.43	**0.06**
2	2	-	-	-	-	**6.82**	1.00	**6.87**	1.00	6.78	1.00	**6.85**	1.00
	3	8.19	0.06	7.38	0.06	7.04	0.82	6.93	0.82	7.06	0.52	6.95	0.51
	4	7.79	0.08	7.27	0.07	7.17	0.78	6.97	0.77	7.18	0.42	6.98	0.40
	5	7.55	0.11	7.11	0.10	7.32	0.35	6.98	0.33	7.33	0.25	7.01	0.24
	6	7.59	0.13	7.14	0.12	7.57	0.20	7.12	0.18	7.56	0.19	7.12	0.18
4	2	-	-	-	-	**6.82**	1.00	6.95	1.00	**6.77**	1.00	6.92	1.00
	3	8.23	0.07	7.49	0.07	7.00	0.92	6.97	0.92	7.01	0.76	6.99	0.75
	4	7.73	0.14	7.26	0.12	7.10	0.92	6.98	0.92	7.10	0.76	6.98	0.75
	5	7.41	0.26	**7.10**	0.23	7.14	0.92	6.93	0.92	7.14	0.83	6.93	0.82
	6	**7.24**	0.29	6.94	0.26	7.22	0.31	6.91	0.29	7.21	0.30	6.91	0.27

to be used as reference. This can be observed, for instance, in the columns labelled W/o DC of Table 6.2.

Number of reference sub-aperture images: In the same way, the compression efficiency increases with the increasing maximum number of reference SAIs, N_R, when the disparity compensation is not used. However, a few exceptions exist that are worth mentioning. One of these exceptions occurs when using $N_R = 2$, with $H_L = 6$, regardless of use of disparity compensation, lower compression performance is achieved when compared to the same configuration using $H_L = 5$. This observation might be partially explained due to the fact that when more hierarchical layers are used the references are usually more distant. This is observed for the three datasets, in Table 6.2, Table 6.3 and Table 6.4.

Disparity compensation: When considering the disparity compensation, further analysis of the results shows that its benefits are not consistent across the datasets. For instance, in the case of the EPFL dataset, the compression efficiency decreases when using the disparity compensation, which is explained by its low inherent disparity. This can be observed in Figure 6.7, for the EPFL plots, by comparing the red lines ($N_R = 1$ and $N_R = 1$ DC), for instance. In this dataset, the disparity is less than one pixel between neighbour SAIs. The low disparity prevents the algorithm from finding a better match than the co-located position. Thus, the additional overhead of encoding the disparity vectors results in a reduction in the compression efficiency. In contrast, both the HCI and SKINL2 datasets generally benefit from the use of the disparity compensation, except for the HCI dataset when using $N_R = 4$. This can be noted in Table 6.4 by comparing the W/o and W/ disparity compensation columns of the configuration without using references on the same layer, for instance, for any N_R and H_L configuration. The benefits of using the disparity compensation are highly correlated with the inherent disparity of the datasets. For this reason, the coding efficiency of the EPFL dataset, that presents the lowest average disparity, is penalised. On the other hand, the SKINL2 dataset, which presents the highest average disparity, benefits more from using the disparity compensation in all configurations.

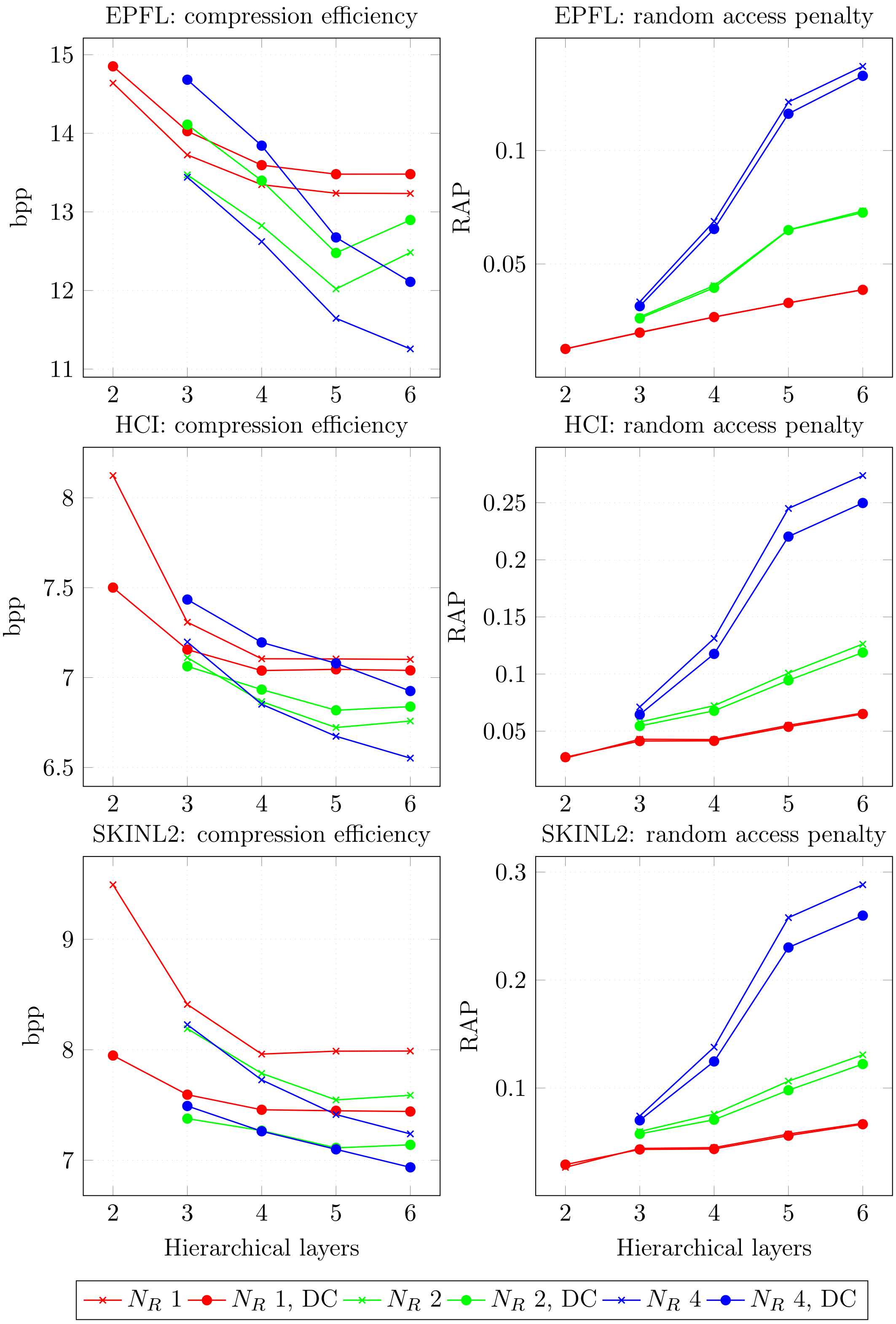


Figure 6.7: Plots of the bpp and RAP values shown in Table 6.2, Table 6.3 and Table 6.4 for the configurations with no references on the same layer.

Experiments With Same-Layer References

For the case where references of the same hierarchical layer are used, an additional option is considered: the scan order for the SAIs compression. Figure 6.8 shows the results for the configurations where H-MRP uses references in the same layer with a spiral scan.

Sub-aperture images scan order: As explained before, the scanning order is relevant in this case as the set of SAIs available to be used as references in the same layer change with the different scanning orders. Two scanning orders are considered: raster and spiral, as detailed in Section 4.1. The spiral scan generally results in the highest compression efficiency. This fact is particularly clear for the EPFL dataset, with differences of up to 0.3 bpp. This can be seen in Table 6.2 by comparing the W/o or the W/ DC columns for both scans, when using references in the same layer. These results are coherent with what is reported in Section 4.4.1. The differences between the results of both scans diminish as the number of hierarchical layers increase. One can mention, for instance, the $N_R = 4$, $H_L = 6$, without disparity compensation configuration in Table 6.2, where the difference between the raster and the spiral scans results is 0.01 bpp.

Number of references and hierarchical layers: When analysing the influence of the number of SAI references and the number of hierarchical layers, the conclusions are quite different from the case where references on the same layer are not used. In fact, the compression efficiency decreases with the increase of the number of hierarchical layers, which also happens to the average maximum RAP. This observation is supported by the slope of the plots on the left side of Figure 6.8. In this case, all the previously encoded SAIs can be used as references. Therefore, the increase in the number of hierarchical layers limits the list of available references, which reduces the amount of data to be decoded prior reaching any of the SAIs, but also penalises the compression efficiency. In fact, when looking at the maximum RAP in the right-side plots of Figure 6.8, it can be observed that values as high as 1.00 are reached when two hierarchical layers are used. According to the definition of RAP, given by Equation 2.8), this means that the whole LF needs to be decoded to access the mostly costly SAI.

Disparity compensation: In regard to the disparity compensation, once again, there is not a clear consistent tendency across the three datasets and various H-MRP configurations. In fact, while in most configurations using disparity compensation decreases the compression efficiency of H-MRP, there are some exceptions. Evidence that using disparity compensation results in decreased performance is shown on the left side of Figure 6.8, where plots with circles (*i.e.*, DC is used) usually present higher bpp than plots of the corresponding colour with crosses (*i.e.*, DC is not used). The exceptions occur when the number of hierarchical layers increase, in which case the conclusions are consistent with those taken without the use of references on the same layer. As the number of hierarchical layers increase, references are further from the encoding SAI, and, consequently, the disparity is higher. This happens when using references in the same layer, because when increasing the number of layers, the neighbouring SAIs tend to be in different, not yet encoded, hierarchical levels. Therefore, disparity compensation allows H-MRP to minimise the differences between a SAI and its references.

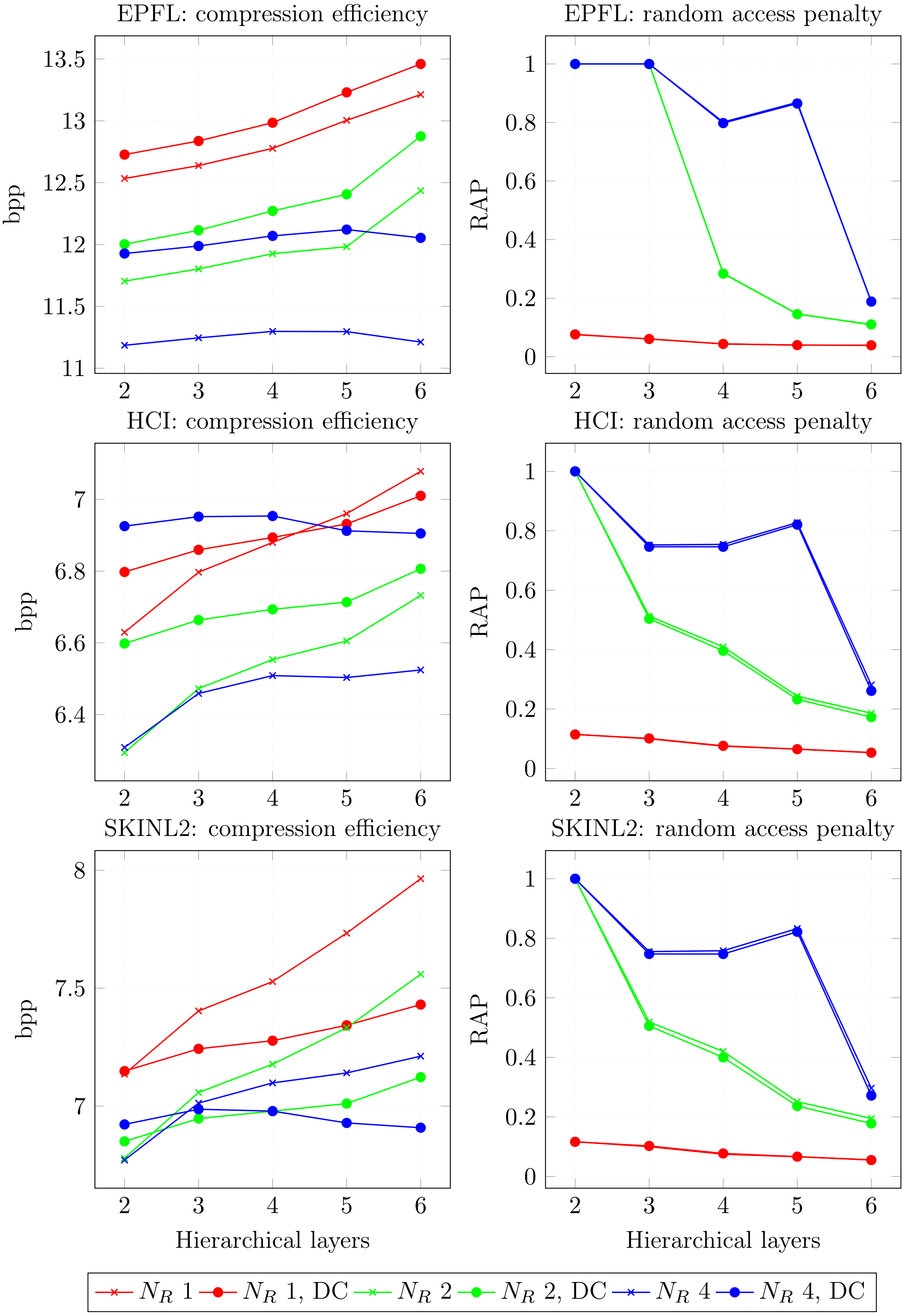


Figure 6.8: Plots of the bpp and RAP values shown in Table 6.2, Table 6.3 and Table 6.4 for the configurations with references on the same layer and using a spiral scan.

Definition of Encoding Configurations

The previous presented studies on the influence of the H-MRP encoding parameters led to the definition of two configurations with different goals: maximum compression efficiency and minimum RAP. The goal of the maximum compression efficiency is to provide the minimum bpp, *i.e.*, the highest compression ratio, when encoding LFs. To this effect, for each dataset, the parameters were selected as those in the previous tables that reached the lowest bpp. For all the used datasets the optimal compression efficiency is achieved when using references in the same layer, encoding the SAIs in a spiral scan, and using two hierarchical layers (which minimises the distance between the current SAI and its references). Considering the number of reference SAIs the optimal efficiency is reached when using $N_R = 4$ for the EPFL and the SKINL2 datasets, and $N_R = 2$ for the HCI dataset.

The main goal of the minimum RAP configuration is to provide a good trade-off between the encoding performance and the RA capabilities of the encoder (*i.e.*, the lowest RAP). The selection of the parameters for this configuration is not as straightforward as in the previous case. One common factor for the three datasets is that using references on the same hierarchical layer greatly increases the RAP, thus SAI references on the same layer are not used in this configuration. Another common factor is that increasing the number of references increases the RAP, thus a good compromise between high coding efficiency and low RAP is reached when setting $N_R = 2$. Due to the higher disparity of the HCI and SKINL2 datasets, when references on the same layer are not used, the disparity compensation results in higher compression ratios, thus this setting was used for the minimum RAP configuration. The selection of the number of hierarchical layers to use is not as simple, and is highly influenced by the need to select better compression efficiency or lower RAP. This parameter was set for each dataset in comparison with other state-of-the-art encoders to achieve a better performance, as will be shown later. Therefore, the EPFL and HCI datasets use $H_L = 4$ and the SKINL2 dataset uses $H_L = 3$.

6.3.2 Comparison with State-of-the-Art Encoders

While the performance evaluation discussed in the previous section referred to compression efficiency and random access capabilities, in this section, the performance of H-MRP against state-of-the-art encoders is presented and discussed. Two configurations are selected: maximum compression efficiency and minimum RAP. The dataset specific configurations are the following:

- Maximum compression efficiency:
  - EPFL: $H_L = 2$, $N_R = 4$, w/ same layer, spiral scan.
  - HCI: $H_L = 2$, $N_R = 2$, w/ same layer, spiral scan.
  - SKINL2: $H_L = 2$, $N_R = 4$, w/ same layer, spiral scan.

- Minimum RAP:
  - EPFL: $H_L = 4$, $N_R = 2$, w/o same layer.
  - HCI: $H_L = 4$, $N_R = 2$, w/o same layer, w/ DC.
  - SKINL2: $H_L = 3$, $N_R = 2$, w/o same layer, w/ DC.

The following state-of-the-art lossless encoders are used in this comparative assessment: MRP [62], HEVC [50], and Versatile Video Coding (VVC) [60], which encode the LF in a PVS format. Additionally, comparisons are made with Dual-Tree 4D Minimum Rate Predictors (DT-4D-MRP) and HEVC-SLF-RA [130] adapted for lossless compression. HEVC-SLF-RA is a LF encoder specifically designed with a focus on random access capabilities, its implementation and used configurations are listed in Table 6.5. The DT-4D-MRP encoder is the only codec in this list without RA capabilities.

Table 6.5: Configurations of the HEVC-SLF-RA encoder used in this work.

Encoder	Software	Configuration
HEVC-SLF-RA	Software from [130]	• Main RExt profile (Profile=main-RExt) • Lossless cost mode • QP 0 (zero) • Transform quantization bypass (TransquantBypassEnable=1) • Configurations by dataset: – EPFL: RPL 4, MDL 6 and NVPR 1, for maximum efficiency configuration, and RPL 2, MDL 9, and NVPR 4, for the maximum random access. – HCI and SKINL2: RPL 4, MDL 6 and NVPR 1, for maximum efficiency configuration, and RPL 2, MDL 2, and NVPR 5, for the maximum random access.

Table 6.6 and Table 6.7 present the results for the comparison with state-of-the-art encoders for compression efficiency and maximum RAP, respectively. In these tables, the best result for each row is in bold. In regard to the state-of-the-art encoders, the highest compression efficiency is achieved by the DT-4D-MRP codec, with 8.72 bpp, averaged over the three datasets. However, this encoder has the worst random access capabilities, with a maximum RAP of 1.0, which means that the whole LF must be decoded to access the most costly SAI. The HEVC-SLF-RA, described in [130], allows some versatility in configuration of the encoding process, permitting the trade-off between a higher compression ratio and better random access capabilities. In these experiments, two configurations of the HEVC-SLF-RA are used, labelled maximum efficiency and minimum RAP[2]. This encoder surpasses the HEVC and VVC encoders for the EPFL and HCI datasets in terms of compression ratio for its maximum efficiency configuration. However, its compression performance is the worst for the SKINL2 dataset. Considering the minimum RAP configuration,

[2] In this context minimum RAP refers to the encoder configuration that minimises the maximum RAP.

the HEVC-SLF-RA is able to produce lower RAP values than the aforementioned encoders, namely HEVC and VVC, for the three datasets. The MRP encoder, which compresses the LF in a pseudo-video sequence using a spiral scan, outperforms the HEVC, VVC, and the maximum efficiency configuration of HEVC-SLF-RA in terms of compression efficiency and RAP. However, this encoder was not built to achieve high random access performance, and does not allow for flexible random access configurations. Consequently it is unable to outperform the minimum RAP configuration of HEVC-SLF-RA in terms of random access capabilities.

Table 6.6: Coding efficiency: state-of-the-art encoders *vs.* H-MRP (bpp).

Type	Light Fields	HEVC	VVC	MRP	H-SLF-RA Max. Eff.	H-SLF-RA Min. RAP	DT-4D-MRP	H-MRP Max. Eff.	H-MRP Min. RAP
LL	Bikes	13.69	13.72	12.73	13.08	13.31	11.34	**11.25**	12.89
	Danger	13.03	12.97	11.89	12.36	12.63	10.70	**10.59**	12.25
	Fountain	14.43	14.52	13.28	13.89	14.10	12.00	**11.84**	13.41
	Stone	13.44	13.38	12.36	12.79	13.02	11.16	**11.05**	12.75
	Img 1	8.65	8.17	8.06	9.49	10.47	6.84	**6.68**	7.24
	Img 2	9.10	8.74	7.81	9.92	10.90	6.97	**6.79**	7.48
	Img 3	9.18	8.79	8.00	9.69	10.66	7.09	**6.85**	7.40
HDCA	Greek	5.55	5.36	5.40	5.72	5.85	**4.78**	4.93	5.40
	Sideboard	8.17	8.09	8.43	8.52	8.93	**7.59**	7.66	8.47
Average		10.58	10.42	9.77	10.61	11.10	8.72	**8.63**	9.70

Table 6.7: Maximum RAP: state-of-the-art encoders *vs.* H-MRP.

Type	Light Fields	HEVC	VVC	MRP	H-SLF-RA Max. Eff.	H-SLF-RA Min. RAP	DT-4D-MRP	H-MRP Max. Eff.	H-MRP Min. RAP
LL	Bikes	0.05	0.05	0.17	0.18	**0.03**	1.00	1.00	0.04
	Danger	0.05	0.05	0.17	0.18	**0.03**	1.00	1.00	0.04
	Fountain	0.05	0.04	0.16	0.18	**0.03**	1.00	1.00	0.04
	Stone	0.05	0.05	0.17	0.18	**0.03**	1.00	1.00	0.04
	Img 1	0.12	0.10	0.21	0.27	**0.06**	1.00	1.00	**0.06**
	Img 2	0.11	0.10	0.21	0.27	**0.06**	1.00	1.00	**0.06**
	Img 3	0.11	0.10	0.21	0.27	**0.06**	1.00	1.00	**0.06**
HDCA	Greek	0.11	0.09	0.19	0.38	**0.03**	1.00	1.00	0.07
	Sideboard	0.11	0.10	0.22	0.38	**0.03**	1.00	1.00	0.07
Average		0.09	0.08	0.19	0.25	**0.04**	1.00	1.00	0.05

In regard to the maximum efficiency configuration, the proposed H-MRP encoder outperforms all the state-of-the-art encoders, with 8.63 bpp and differences ranging from 0.09 for DT-4D-MRP to 1.98 bpp for the HEVC-SLF-RA, on average. These differences are achieved by subtracting the columns DT-4D-MRP and HEVC-SLF-RA Max. Eff. from column H-MRP Max. Eff. in Table 6.6. This means that the proposed method outperforms the DT-4D-MRP by approximately 1%, on average. This might be explained by the flexibility in the references selection, as the fixed reference positions DT-4D-MRP might not always be available. When compared with DT-4D-MRP, the results show that higher differences are obtained for the EPFL and SKINL2 datasets, as for the HCI dataset the proposed encoder is less efficient. The results obtained for the maximum efficiency configuration show that the H-MRP encoder achieves the highest compression efficiency in comparison with all lossless LF encoders reported so far in the literature, surpassing DT-4D-MRP and establishing the new state-of-the-art.

In terms of the minimum RAP configuration, the proposed method achieves slightly higher RAP when compared with HEVC-SLF-RA, however it reaches higher compression efficiency, with a difference of 1.4 bpp (12.6%), on average. This difference is achieved by subtracting the column HEVC-SLF-RA Min. RAP from column H-MRP Min. RAP in Table 6.7. It is also worthy of note, that the H-MRP with the minimum RAP configuration achieves higher compression efficiency than all other encoders, with the exception of DT-4D-MRP, providing a good trade-off between compression and random access capabilities. In terms of RAP, when compared with HEVC-SLF-RA, H-MRP is only able to achieve equal performance for the SKINL2 dataset, while for the HCI dataset, the RAP achieved by H-MRP more than doubles that of HEVC-SLF-RA. However, due to the higher compression performance, it stands to reason that with slight adjustments in the configuration H-MRP could reach a more competitive performance when compared with HEVC-SLF-RA. One way to do this would be through the use of *random access regions*, as it will be shown in the next section.

6.3.3 Random Access Regions in H-MRP

In order to improve the random access capabilities, the concept of random access regions was introduced to H-MRP in Section 6.2.4. The HEVC-SLF-RA resorts to a similar technique to improve its own random access capabilities. The experimental results on the impact of using this technique are shown in Table 6.8 and Table 6.9, for the compression efficiency and RAP, respectively, for the four types of RARs shown in Figure 6.5.

Table 6.8: Effect of the use of RARs on H-MRP: compression efficiency.

Type	Light Fields	Maximum Efficiency Configuration					Minimum RAP Configuration				
		No RAR	RAR 2	RAR 4	RAR 5	RAR 9	No RAR	RAR 2	RAR 4	RAR 5	RAR 9
LL	Bikes	11.25	11.26	11.37	11.68	11.81	12.89	12.89	13.05	13.41	13.44
	Danger	10.59	10.60	10.72	11.06	11.20	12.25	12.25	12.43	12.86	12.90
	Fountain	11.84	11.86	11.96	12.26	12.38	13.41	13.41	13.56	13.87	13.89
	Stone	11.05	11.06	11.20	11.54	11.68	12.75	12.75	12.94	13.39	13.42
	Img 1	6.68	6.68	6.73	7.02	7.14	7.24	7.24	7.39	7.72	7.71
	Img 2	6.79	6.79	6.84	7.13	7.28	7.48	7.48	7.62	7.93	7.93
	Img 3	6.85	6.85	6.91	7.18	7.33	7.40	7.40	7.52	7.90	7.90
HDCA	Greek	4.93	4.96	4.98	5.09	5.14	5.40	5.40	5.41	5.58	5.55
	Sideboard	7.66	7.65	7.71	8.00	8.18	8.47	8.47	8.57	9.23	9.14
Average		8.63	8.63	8.71	9.00	9.13	9.70	9.70	9.83	10.21	10.21

The results show that, for both configurations the compression efficiency decreases as the number of RARs increases and the RAP decreases, *i.e.*, the random access capabilities are improved as the number of relative bits required to access a given SAI decreases, as expected. These conclusions are explained by the constraints that the RARs impose to the selection of reference SAIs. Interestingly, the results show that using only two random access regions does not affect the

Table 6.9: Effect of the use of RARs on H-MRP: random access penalty.

Type	Light Fields	Maximum Efficiency Configuration					Minimum RAP Configuration				
		No RAR	RAR 2	RAR 4	RAR 5	RAR 9	No RAR	RAR 2	RAR 4	RAR 5	RAR 9
LL	Bikes	1.00	1.00	0.53	0.23	0.14	0.04	0.04	0.04	0.03	0.02
	Danger	1.00	1.00	0.53	0.23	0.14	0.04	0.04	0.04	0.03	0.02
	Fountain	1.00	1.00	0.52	0.23	0.14	0.04	0.04	0.04	0.03	0.02
	Stone	1.00	1.00	0.53	0.23	0.15	0.04	0.04	0.04	0.03	0.02
	Img 1	1.00	1.00	0.55	0.30	0.29	0.06	0.06	0.06	0.04	0.03
	Img 2	1.00	1.00	0.55	0.30	0.29	0.06	0.06	0.06	0.04	0.03
	Img 3	1.00	1.00	0.55	0.30	0.29	0.06	0.06	0.06	0.04	0.03
HDCA	Greek	1.00	1.00	0.55	0.30	0.30	0.07	0.07	0.07	0.04	0.04
	Sideboard	1.00	1.00	0.55	0.30	0.30	0.07	0.07	0.07	0.04	0.04
Average		1.00	1.00	0.54	0.27	0.23	0.05	0.05	0.05	0.04	0.03

encoding performance of H-MRP. This might be explained by the large number of SAIs in each of the two regions, which are able to provide appropriate references for the prediction. Despite the general diminishing tendency of the compression efficiency with the increasing number of RARs, there is an exception. For the *minimum RAP* configurations, using nine random access regions achieves equal or higher compression performance than when using five regions of random access, *i.e.*, RAR 5. This happens for both the SKINL2 and the HCI datasets. These are the datasets with the highest average disparities between neighbour SAIs. These results may be explained by the fact that when fewer SAIs are available in each region, the references are necessarily closer, providing better predictions.

For the maximum efficiency configuration case, it can be seen that, for any number of RARs, the proposed encoder still achieves higher compression efficiency than all other encoders, with the exception of DT-4D-MRP and H-MRP without random access regions. Considering both the compression ratio and the random access capabilities, it is shown that for the RAR 5 and RAR 9 configurations, the proposed encoder is able to outperform HEVC-SLF-RA in the maximum efficiency configuration. Achieving differences of approximately 1.50 bpp for the same RAP value, around 0.25.

For the minimum RAP configuration, H-MRP achieves a good compromise in terms of compression ratio, outperforming HEVC, VVC, and both configurations of HEVC-SLF-RA for any number random access regions. The best random access performance in Table 6.7 is achieved by HEVC-SLF-RA with an average RAP equal to 0.04. Table 6.9 shows that the RAR 5 configuration presents the same average RAP as HEVC-SLF-RA in Table 6.6 and configuration RAR 9 presents an RAP of 0.03, representing an improvement of 25% in comparison with HEVC-SLF-RA. Taking into account both the random access capabilities and the compression ratio, it is clear that H-MRP is able to achieve similar or lower RAP for a bitrate that is lower by approximately 1.00 bpp, on average (10.21 bpp), when compared with HEVC-SLF-RA (11.10 bpp).

Overall, the above results show that the use of random access regions increases the flexibility of H-MRP, improving its random access capabilities, with minimal compression efficiency loss.

Therefore, despite the loss in compression performance, H-MRP with random access regions is still competitive in this regard.

6.4 Summary

In this chapter, a new hierarchical encoder, H-MRP, was proposed to combine efficient compression with angular scalability and random access capabilities in light field coding. The proposed approach is based on three major tools: (i) hierarchical coding using multiple layers; (ii) flexible selection of the reference SAIs; and (iii) disparity compensation. An extensive evaluation procedure was carried out to fully assess the capabilities of the proposed encoder under two distinct configurations: (i) maximum compression efficiency and (ii) minimum RAP. The proposed H-MRP encoder, in its maximum compression efficiency configuration, outperformed all the state-of-art lossless encoders by 0.10 to 2.00 bpp, for DT-4D-MRP and HEVC-SLF-RA, respectively, on average. For the minimum RAP configuration, H-MRP achieved the best compromise in terms of compression efficiency and random access capabilities amongst the tested encoders, obtaining a compression efficiency gain of at least 1.00 bpp and an average improvement of 25% in the random access capabilities when compared to its best competitor. Overall, the H-MRP encoder is a flexible solution to encode light fields efficiently, allowing trading-off coding efficiency for random access capabilities or vice-versa, as required by different applications.

CHAPTER 7

Conclusions and Future Work

Contents

This thesis addresses the problem of efficient lossless coding for light fields (LFs). LFs have plenty of applications, ranging from the simple image acquisition for refocusing or extraction of various viewpoints, to more sophisticated applications like the acquisition of skin lesions LFs for medical diagnosis. However, as LFs comprise a much richer information than traditional image formats, they require huge amounts of storage capacity and bandwidth to be handled, especially when compared to conventional 2D images or even video sequences. Consequently, the development of efficient LF coding methods might contribute to a more widespread adoption of this technology. Lossless compression in particular is a requirement for applications that cannot afford the loss of information, such as medical imaging or precision measurements in industry, which need high fidelity in the reconstruction of images. LF imaging technology works by creating a 3D colour representation of the images scene through the capture of light rays emanating from the scene object in different directions. The previous chapters presented new lossless LF coding algorithms developed to address the open challenges in LF lossless compression.

This chapter summarises the main achievements and contributions of this thesis, highlighting the most relevant results obtained with the proposed algorithms (Section 7.1). It is finalised with a discussion about some research problems identified during this work and not yet solved, and with an outline of new research directions in the scope of this work (Section 7.2).

7.1 Conclusions

This thesis proposes new methods for efficient lossless compression of light fields. To provide useful solutions, the methods are based on a common 4D LF representation. Reproducibility of the results presented is ensured through the adoption, adapted for the lossless case, of the Common Test Conditions of the JPEG-Pleno (JPEG-Pleno CTC) for the empirical performance evaluations. Throughout the previous chapters, this thesis discussed the status of LF lossless coding as reported in the literature and presented new algorithms to address some of the shortcomings of previous works. In order to fulfil the purpose of this thesis, three major objectives were planned and carried out:

1. **Research pre-processing algorithms for light field compression and representation arrangements beyond the raw uncompressed RGB format, enabling efficient compression.**

2. **Research efficient prediction-based lossless coding methods adapted to light fields.**

3. **Research methods for light field lossless coding with enhanced capabilities, such as scalability and random access (RA).**

The initial steps of this research investigated pre-processing methods to improve the compression of LFs and the optimal data arrangements for generic codecs compliant coding. This research direction emphasises the use of generic state-of-the-art lossless codecs to efficiently compress this type of data. As these types of codecs are not prepared to take advantage of and handle efficiently the 4D LF data, three types of compatible data arrangements were studied, namely: lenslet arrays, encompassing a rectified array of micro images (MIs); stacks of sub-aperture images (SAIs) that form a pseudo-video sequence (PVS), where the order of the SAIs can be defined; and stacks of epipolar plane images (EPIs). This methodology allowed to find the best data representation format for use with state-of-the-art encoders, which can promote early adoption of LFs as these encoders are more readily available.

The empirical studies, described in Chapter 4, showed that the pair Minimum Rate Predictors (MRP) with the PVS arrangement (using a spiral scan) achieves the highest compression ratio, surpassing the High Efficiency Video Coding (HEVC) by 4.8%, on average, for the tested LFs. Additionally, this thesis also found that the common RGB colour-space is not optimal for compression, due to the inherent redundancies present in its components. Therefore, several reversible colour transforms were studied to assess their influence in the encoding performance. Results showed that the RCT provides the highest compression efficiency of all the studied reversible colour transforms for the lossless compression of LFs. Furthermore, it was noted that by using the RCT the performance of MRP (for the PVS arrangement) is increased by 10.9%, on average, when compared with the RGB case.

Another contribution of this work was the development of a fully reversible disparity compensation method, which is able to minimise the disparity between the LF SAIs prior to the encoding process. However, due to the increment in the number of pixels, that needed to be encoded, lossless encoders do not benefit from this type of disparity compensation. Nevertheless, it was shown that 4D transform-based LF lossy encoders like Multidimensional Light Field Encoder (MuLE) benefits significantly from using this method, with an average Bjøntegaard delta (BD)-rate improvement of 44% for high density camera array (HDCA) acquired LFs.

During this thesis second major activity, three new algorithms which exploit the 4D redundancies of LFs were proposed, namely: Four-Dimensional Minimum Rate Predictors (4D-MRP), Dual-Tree 4D Minimum Rate Predictors (DT-4D-MRP), and Multi-reference Minimum Rate Predictors (M-MRP). In these methods, a SAI based 4D prediction algorithm was introduced, enabling to exploit not only the spatial redundancies in SAIs but also the similarities between neighbouring SAIs, *i.e.*, the angular redundancies. These algorithms differ mostly in the block partition approach used. 4D-MRP and DT-4D-MRP both use 4D blocks, with the partitioning into sub-blocks of 4D-MRP resulting from splitting the four dimensions at the same time and in DT-4D-MRP separating the partition of the spatial dimensions from that of the angular dimensions, while M-MRP uses 2D blocks, following the methodology already used in MRP, for the selection of prediction classes.

The three proposed algorithms were extensively tested against other state-of-the-art methods and compared with each other. The results clearly demonstrate the superiority in terms of coding efficiency of the proposed algorithms when compared to other state-of-the-art encoders, like HEVC and MRP, for the experimental conditions set in Section 2.4, which follow the JPEG-Pleno CTC. Overall, DT-4D-MRP presents the highest compression efficiency of the three, surpassing MRP by 10.2% and HEVC by 15.8%, on average, in terms of bitrate, and thus surpassing the previous state-of-the-art of lossless LF compression. The performance of 4D-MRP and M-MRP is slightly lower that than of DT-4D-MRP, by up to 0.1 bits-per-pixel (bpp), or roughly 0.9%. One aspect that differentiates these algorithms is their computational complexity. Due to its dual-quadtree, the DT-4D-MRP is, on average, 11 times slower than M-MRP a fact that has to be taken into account when selecting one of these methods for specific applications. A study on the computational performance of M-MRP was conducted, which revealed that its encoding time can be reduced by about 33 times, while keeping the increase of bitrate lower than 2%. Assuming that these conclusions can be extrapolated for the DT-4D-MRP case, it stands to reason that the higher computational complexity of DT-4D-MRP can be managed with the proper selection of input parameters, whilst keeping a high compression ratio.

The last contribution of this doctoral work was the Hierarchical Minimum Rate Predictors (H-MRP), a new efficient lossless LF coding algorithm with useful features, such as enhanced scalability and random access functionalities. These capabilities simplify the navigation through LF sub-aperture images, reducing the decoding delay and the computational resources required at the decoder, allowing easier access to specific regions of the LF without full decoding of the bitstream. A comprehensive experimental study was carried out to thoroughly assess the performance of H-MRP, both in terms of compression efficiency and random access capabilities. Two configurations for H-MRP were used in this study, maximum coding efficiency and minimum random access penalty (RAP). The first configuration prioritises the compression ratio while the second gives preference to low RAP. In its first configuration, H-MRP is able to surpass the compression efficiency of the encoder that previously had the best performance on the literature, DT-4D-MRP, by 1%, on average. Regarding the minimum RAP configuration, results show that H-MRP can surpass other state-of-the-art encoders, like HEVC-SLF-RA, both in terms of compression efficiency and RA capabilities, with a result that is, at least, 1 bpp lower and an improvement of 25%, in terms of RAP, when compared to its best competitor, HEVC-SLF-RA.

One limitation of the experimental studies presented in this thesis arise from the lack of comparison with other LF lossless encoders. This shortcoming is due to the fact that competing solutions do not use a common experimental setup, such as the JPEG-Pleno CTC set by the JPEG-Pleno standardisation initiative, which makes fair comparison between methods difficult. This problem is aggravated by the lack of publicly available implementations (by the authors or others) of the competing methods.

Nevertheless, the objectives set for these doctoral studies were successfully achieved, as can be seen by the reports in previous paragraphs. The contributions of this thesis helped to expand on the state-of-the-art of lossless compression of light fields, often setting the lowest bitrates for the compressed images (using the JPEG-Pleno CTC) in the literature. Overall, the results of this doctoral work can be subsumed by the following three major contributions:

Contribution 1. Compatible methods for the lossless compression of LFs:

Three types of LF data arrangements were proposed and evaluated in this thesis (Chapter 4), which can be used as the input of common encoders (published in J3). These methods also benefit from the introduction of the RCT colour transform for compression, which greatly increases the compression efficiency due to the decorrelation of the colour components (published in J4 and C5).

Contribution 2. MRP based 4D LF lossless codecs:

This contribution is described in Chapter 5, where three codecs that are able to exploit the inherent redundancies of the 4D LFs are proposed. The DT-4D-MRP codec is a noteworthy contribution as it sets a new state-of-the-art for the lossless compression of LFs (published in J2, C2, C3, and C4)

Contribution 3. Scalable and random access lossless LF codec:

The H-MRP codec, introduced in Chapter 6, is, to the best of the author's knowledge, the only lossless LF codec with scalability and random access capabilities. These capabilities are achieved through the use of an hierarchical coding architecture on the encoder, that was shown to be able to surpass other RA encoders primarily designed for LF lossy compression, but also the previous state-of-the-art DT-4D-MRP in terms of compression efficiency (published in J1).

7.2 Future Work

The work developed on these doctoral studies established new benchmarks for future research on LF lossless coding, as can be inferred from an analysis of the results reported consistently outperforming those of several state-of-the-art methods. However, there are still some open issues that are worth investigating. This section proposes some research directions that may strengthen and improve upon the proposed methods.

7.2.1 Improve the Computational Efficiency of DT-4D-MRP

The computational complexity of DT-4D-MRP is a problem that needs to be solved. As described in Section 5.5.3, the encoding parameters of M-MRP can be adjusted to reduce the computational complexity of the encoder, while maintaining a high compression ratio. Thus, in the future, a similar study shall be done for DT-4D-MRP to understand if a reduction in computational complexity can be achieved in the same way. It is arguable that some optimisations can be introduced in the DT-4D-MRP code to increase its computational efficiency, without compromising the compression efficiency. One possibility, is the inclusion of parallel computation of the MRP linear prediction coefficients, for instance. An extensive profiling of its code is necessary for a proper understanding of the algorithm time allocations, leading to a further suggestions for computational efficiency improvements.

7.2.2 Improve the Selection of Reference SAIs in H-MRP

Another research topic that may be explored is the selection of reference SAIs in H-MRP. Currently, the encoder selects the references to the SAI to be encoded according to their distance, in the (t, s) dimensions of the 4D space. This selection can be refined by introducing similarity metrics, like the sum of squared differences, to compare the potential references to the current SAI, taking also into account the influence of the disparity compensation algorithm for the similarity metrics computation. These improvements would probably allow the H-MRP algorithm to increase its compression efficiency, especially for datasets with higher disparity, like that of Heidelberg Collaboratory for Image Processing (HCI).

7.2.3 Improvements to the MRP Algorithm

A different exploration worth pursuing in the future is related to the improvement of the inner workings of the family of MRP encoders. Two important operations that are part of these algorithms are the prediction and the residuals entropy coding. Considering the prediction, the use of neural networks and deep learning can be useful to enhance its efficiency leading to a better reduction of redundancies from the encoding images and consequently to prediction residues with lower entropy. Furthermore, these machine learning techniques can also help to tackle the classification of blocks in MRP which is one of the most time intensive tasks in the algorithm.

The largest part of the MRP bitstream is taken by the representation of the prediction residuals, as shown in Chapter 4. Therefore, there might be room for improvement on the efficiency of the entropy coding. Two ways might be mentioned to address this issue: one by improving the entropy probability models that are used, either by using adaptive context modelling or through the development of probabilistic models that more closely reflect the statistic of residues generated by MRP. Another way is to replace the current entropy coder with a more recent technique, like the context-adaptive binary arithmetic coder used in recent standards like HEVC and Versatile Video Coding (VVC).

Bibliography

[1] G. Lippmann, “Epreuves reversibles donnant la sensation du relief,” *J. Phys. Theor. Appl.*, vol. 7, no. 1, pp. 821–825, Jan. 1908.

[2] J. Wang, X. Xiao, H. Hua, and B. Javidi, “Augmented reality 3D displays with micro integral imaging,” *IEEE/OSA J. Display Technol.*, vol. 11, no. 11, pp. 889–893, Nov. 2015.

[3] J. Arai, M. Kawakita, T. Yamashita, H. Sasaki, M. Miura, H. Hiura, M. Okui, and F. Okano, “Integral three-dimensional television with video system using pixel-offset method,” *Opt. Express*, vol. 21, no. 3, pp. 3474–3485, Feb. 2013.

[4] D. Shin, M. Cho, and B. Javidi, “Three-dimensional optical microscopy using axially distributed image sensing,” *Opt. Lett.*, vol. 35, no. 21, pp. 3646–3648, Nov. 2010.

[5] P. M. M. Pereira, R. Fonseca-Pinto, R. P. Paiva, P. A. A. Assuncao, L. M. N. Tavora, L. A. Thomaz, and S. M. M. Faria, “Dermoscopic skin lesion image segmentation based on local binary pattern clustering: Comparative study,” *Biomedical Signal Processing and Control*, vol. 59, p. 101765, May 2020.

[6] ——, “Skin lesion classification enhancement using border-line features – the melanoma vs nevus problem,” *Biomedical Signal Processing and Control*, vol. 57, p. 101765, Mar. 2020.

[7] W. Zhou, E. Zhou, G. Liu, L. Lin, and A. Lumsdaine, “Unsupervised monocular depth estimation from light field image,” *IEEE Trans. Image Process.*, vol. 29, pp. 1606–1617, Oct. 2020.

[8] K. Mishiba, “Fast depth estimation for light field cameras,” *IEEE Trans. Image Process.*, vol. 29, pp. 4232–4242, Feb. 2020.

[9] R. Lourenco, P. A. A. Assuncao, L. M. N. Tavora, R. Fonseca-Pinto, and S. M. M. Faria, “Silhouette enhancement in light field disparity estimation using the structure tensor,” in *IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, Oct. 2018, pp. 2580–2584.

[10] H. Jung, H. Lee, and C. E. Rhee, “Flexibly connectable light field system for free view exploration,” *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 980–991, Apr. 2020.

[11] Y. Wang, J. Yang, Y. Guo, C. Xiao, and W. An, “Selective light field refocusing for camera arrays using bokeh rendering and superresolution,” *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 204–208, Jan. 2019.

[12] Stanford University Computer Graphics Laboratory, “The (New) Stanford Light Field Archive,” http://lightfield.stanford.edu/, 2008, [Accessed 16 August 2021].

[13] C. Zhou and S. K. Nayar, “Computational cameras: Convergence of optics and processing,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3322–3340, Dec. 2011.

[14] H. G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. W. Tai, and I. S. Kweon, “Accurate depth map estimation from a lenslet light field camera,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1547–1555.

[15] M. Harris, “Focusing on everything,” *IEEE Spectr.*, vol. 49, no. 5, pp. 44–50, May 2012.

[16] D. Donatsch, S. A. Bigdeli, P. Robert, and M. Zwicker, “Hand-held 3D light field photography and applications,” *The Visual Computer*, vol. 30, no. 6-8, p. 897, Jun. 2014.

[17] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Linear volumetric focus for light field cameras,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, pp. 1–20, Mar. 2015.

[18] S. Baghdadchi, K. Liu, J. Knapp, G. Prager, S. Graves, K. Akrami, R. Manuel, R. Bastos, E. Reid, D. Carson, S. Esener, J. Carson, and Y.-T. Liu, “An innovative system for 3D clinical photography in the resource-limited settings,” *Journal of translational medicine*, vol. 12, no. 1, pp. 1–8, Jun. 2014.

[19] J. K. Makanjuola, A. Aggoun, M. Swash, P. C. Grange, B. Challacombe, and P. Dasgupta, “3D-Holoscopic Imaging: A New Dimension to Enhance Imaging in Minimally Invasive Therapy in Urologic Oncology,” *Journal of Endourology*, vol. 27, no. 5, pp. 535–539, May 2013.

[20] R. Marshall, I. Styles, E. Claridge, and K. Bongs, “Plenoptic imaging of the retina: can it resolve depth in scattering tissues?” in *Biomedical Optics.* Optical Society of America, 2014, pp. BM3A–60.

[21] R. Decker, A. Shademan, J. Opfermann, S. Leonard, P. C. W. Kim, and A. Krieger, “Performance evaluation and clinical applications of 3D plenoptic cameras,” in *Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*, M. Blowers, D. Popa, and M. B. J. Wijesundara, Eds., vol. 9494, International Society for Optics and Photonics. Baltimore, MD, United States: SPIE, Jun. 2015, pp. 62 – 72.

[22] Seema, A. Kumar, and G. S. Gill, "Automatic fruit grading and classification system using computer vision: A review," in *International Conference on Advances in Computing and Communication Engineering*, Dehradun, India, May 2015, pp. 598–603.

[23] A. Landstrom, M. J. Thurley, and H. Jonsson, "Sub-millimeter crack detection in casted steel using color photometric stereo," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Hobart, TAS, Australia, Nov. 2013, pp. 1–7.

[24] X. Huang, E. Uffelman, O. Cossairt, M. Walton, and A. K. Katsaggelos, "Computational Imaging for Cultural Heritage: Recent developments in spectral imaging, 3-D surface measurement, image relighting, and X-ray mapping," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 130–138, Sep. 2016.

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

[26] "JPEG Pleno Call for Proposals on Light Field Coding," International Organization for Standardization (ISO), Chengdu, China, Tech. Rep. ISO/IEC JTC 1/SC29/WG1 N73013, Oct. 2016.

[27] S. M. M. de Faria, J. N. Filipe, P. M. M. Pereira, L. M. N. Távora, P. A. A. Assunção, M. O. Santos, R. Fonseca-Pinto, F. Santiago, V. Dominguez, and M. Henrique, "Light field image dataset of skin lesions," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, Jul. 2019, pp. 3905–3908.

[28] M. Vestergaard, P. Macaskill, P. Holt, and S. Menzies, "Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting," *British Journal of Dermatology*, vol. 159, no. 3, pp. 669–676, Jul. 2008.

[29] K. Korotkov and R. Garcia, "Computerized analysis of pigmented skin lesions: A review," *Artificial Intelligence in Medicine*, vol. 56, no. 2, pp. 69–90, Oct. 2012.

[30] P. M. M. Pereira, L. A. Thomaz, L. M. N. Tavora, P. A. A. Assuncao, R. Fonseca-Pinto, R. P. Paiva, and S. M. M. Faria, "Skin lesion classification using features of 3D border lines," in *Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, Guadalajara, Mexico, Jul. 2021.

[31] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot, "Impact of light field compression on focus stack and extended focus images," in *European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 898–902.

[32] J. Taquet and C. Labit, "Hierarchical oriented predictions for resolution scalable lossless and near-lossless compression of CT and MRI biomedical images," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2641–2652, May 2012.

[33] "Strategic Document," Digital Imaging and Communications in Medicine (DICOM), Salt Lake City, UT, USA, Tech. Rep., Jun. 2010, [Revised March 2017]. [Online]. Available: http://dicom.nema.org/dicom/geninfo/Strategy.pdf

[34] "Data Structures and Encoding," Digital Imaging and Communications in Medicine (DICOM, Arlington, VA, USA, Tech. Rep. PS3.5 2021c, 2021. [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf

[35] "Health informatics – Digital imaging and communication in medicine (DICOM) including workflow and data management," International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO 12052:2017, Aug. 2017.

[36] A. Naït-Ali and C. Cavaro-Ménard, *Compression of Biomedical Images and Signals*, ser. ISTE. London, United Kingdom: Wiley, 2008.

[37] "The adoption of lossy image data compression for the purpose of clinical interpretation," Royal College of Radiologists, London, United Kingdom, Tech. Rep. BFCR(08)11, Apr. 2008.

[38] "A guideline for the use of image compression in diagnostic imaging," Royal Australian and New Zealand College of Radiologists, Sidney Australia, Tech. Rep., Aug. 2020. [Online]. Available: http://www.ranzcr.edu.au/documents-download/document-library-2/document-library-3/574-a-guideline-for-the-use-of-image-compression-in-diagnostic-imaging

[39] C. Zhang and T. Chen, *Light Field Sampling*, 1st ed. Williston, VT, USA: Morgan & Claypool, 2006.

[40] M. Levoy and P. Hanrahan, "Light field rendering," in *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, New York, NY, USA, Aug. 1996, p. 31–42.

[41] C. Hahne, A. Aggoun, S. Haxha, V. Velisavljevic, and J. C. J. Fernández, "Light field geometry of a standard plenoptic camera," *Opt. Express*, vol. 22, pp. 26 659–26 673, Nov. 2014.

[42] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *IEEE International Conference on Computational Photography (ICCP)*, San Francisco, CA, USA, Apr. 2009, pp. 1–8.

[43] M. Rerabek and T. Ebrahimi, "New Light Field Image Dataset," in *International Conference on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, Jun. 2016, pp. 1 – 2.

[44] Stanford University Computer Graphics Laboratory, "The Stanford Multi-Camera Array," http://graphics.stanford.edu/projects/array/, 2002, [Revised January 2011; accessed 16 August 2021].

[45] ——, “A light field of Michelangelo’s statue of Night,” https://accademia.stanford.edu/mich/lightfield-of-night/, May 1999, [Accessed 16 August 2021].

[46] 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)*, Portland, OR, USA, Jun. 2013, pp. 1027–1034.

[47] D. Dansereau, “Light Field Toolbox,” https://www.mathworks.com/matlabcentral/fileexchange/75250-light-field-toolbox, Apr. 2020, [Accessed 16 August 2021].

[48] S. Wanner, J. Fehr, and B. Jähne, *Generating EPI Representations of 4D Light Fields with a Single Lens Focused Plenoptic Camera.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 90–101.

[49] “JPEG Pleno Light Field Coding Common Test Conditions v3.3,” International Organization for Standardization (ISO), Brussels, Belgium, Tech. Rep. ISO/IEC JTC 1/SC 29/WG 1 N84025, Jul. 2019.

[50] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Sep. 2012.

[51] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, “A dataset and evaluation methodology for depth estimation on 4D light fields,” in *Asian Conference on Computer Vision (ACCV)*, Taipei, Taiwan, Nov. 2017, pp. 19–34.

[52] G. Bjøntegaard, “Calculation of average PSNR differences between RD curves,” International Telecommunication Union (ITU), Austin, TX, USA, Tech. Rep. VCEG-M33, Apr. 2001.

[53] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, “Standardized extensions of high efficiency video coding (HEVC),” *IEEE Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[54] “Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding,” International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO/IEC 23008-2:2020, Aug. 2020.

[55] “Information technology – JPEG 2000 image coding system – Part 1: Core coding system,” International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO/IEC 15444-1:2019, Oct. 2019.

[56] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[57] “Information technology — Lossless and near-lossless compression of continuous-tone still images,” International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO/IEC 14495-1:1999, Dec. 1999.

[58] X. Wu and N. Memon, “Context-based, adaptive, lossless image coding,” *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[59] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, “Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 4–19, Sep. 2016.

[60] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, and Y. K. Wang, “Developments in international video coding standardization after avc, with an overview of versatile video coding (VVC),” *Proc. IEEE*, pp. 1–31, Jan. 2021.

[61] “Information technology - Coded representation of immersive media - Part 3: Versatile video coding,” International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO/IEC 23090-3:2021, Aug. 2020.

[62] J. M. Santos, A. F. R. Guarda, L. A. da Silva Cruz, N. M. M. Rodrigues, and S. M. M. Faria, “Compression of medical images using MRP with bi-directional prediction and histogram packing,” in *Picture Coding Symposium (PCS)*, Nuremberg, Germany, Dec. 2016, pp. 1–5.

[63] Université de Louvain (UCL), “Openjpeg JPEG2000 encoder v. 2.1.2,” http://www.openjpeg.org/, Sep. 2016, [Accessed 16 August 2021].

[64] T. Richter, “JPEG-LS encoder implementation,” https://github.com/thorfdbg/libjpeg, 2012, [Accessed 16 August 2021].

[65] M. Briano and J. Serra-Sagristà, “CALIC, 3D CALIC and M-CALIC customizable software implementation,” https://prezi.com/j2ichhip4ls7/calic-implementation/, Jul. 2014, [Accessed 16 August 2021].

[66] Fraunhofer Heinrich Hertz Institute, “HEVC reference software encoder,” https://hevc.hhi.fraunhofer.de/, Dec. 2012, [Accessed 16 August 2021].

[67] ——, “VVC reference software encoder,” https://vvc.hhi.fraunhofer.de/, Apr. 2018, [Accessed 16 August 2021].

[68] I. Matsuda, Y. Umezu, N. Ozaki, J. Maeda, and S. Itoh, “A lossless coding scheme using adaptive predictors and arithmetic code optimized for each image,” *Systems and Computers in Japan*, vol. 38, no. 4, pp. 1–11, Mar. 2007.

[69] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[70] ——, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, Oct. 1948.

[71] W. A. Pearlman and A. Said, *Digital signal compression: Principles and Practice*, 1st ed. Cambridge Univ. Press, 2011, includes bibliographical references and index.

[72] K. Sayood, *Introduction to Data Compression (2nd Ed.)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

[73] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[74] A. Said, “Chapter 5 - arithmetic coding,” in *Lossless Compression Handbook*, ser. Communications, Networking and Multimedia, K. Sayood, Ed. San Diego: Academic Press, 2003, pp. 101 – 152.

[75] S. Golomb, “Run-length encodings (corresp.),” *IEEE Trans. Inf. Theory*, vol. 12, no. 3, pp. 399–401, Jul. 1966.

[76] S. C. Sahinalp and N. M. Rajpoot, “Chapter 6 - dictionary-based data compression: An algorithmic perspective,” in *Lossless Compression Handbook*, ser. Communications, Networking and Multimedia, K. Sayood, Ed. San Diego: Academic Press, 2003, pp. 153 – 167.

[77] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, “A 4D DCT-based lenslet light field codec,” in *IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, Oct. 2018, pp. 435–439.

[78] 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)*, Tampere, Finland, Nov. 2018, pp. 1–6.

[79] ——, “Coding of light fields using disparity-based sparse prediction,” *IEEE Access*, vol. 7, pp. 176 820–176 837, Dec. 2019.

[80] P. Helin, P. Astola, B. Rao, and I. Tabus, “Minimum description length sparse modeling and region merging for lossless plenoptic image compression,” *IEEE Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1146–1161, Oct. 2017.

[81] A. Aggoun, “A 3D DCT compression algorithm for omnidirectional integral images,” in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, vol. 2, Toulouse, France, May 2006, p. II.

[82] 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 *IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, Sep. 2011, pp. 961–964.

[83] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Aug. 2003.

[84] C. Conti, P. Nunes, and L. D. Soares, “Inter-Layer Prediction Scheme for Scalable 3-D Holoscopic Video Coding,” *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 819–822, Aug. 2013.

[85] 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)*, Orleans, France, Nov. 2015, pp. 494–497.

[86] C. Perra and P. Assunção, “High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement,” in *IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, Seattle, WA, USA, Jul. 2016, pp. 1–4.

[87] A. Aggoun, “Compression of 3D Integral Images Using 3D Wavelet Transform,” *IEEE/OSA J. Display Technol.*, vol. 7, no. 11, pp. 586–592, Nov. 2011.

[88] S. Shi, P. Gioia, and G. Madec, “Efficient compression method for integral images using multi-view video coding,” in *IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, Sep. 2011, pp. 137–140.

[89] N. C. Francisco, N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, and S. M. M. de Faria, “Video compression using 3D multiscale recurrent patterns,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Beijing, China, May 2013, pp. 1412–1415.

[90] L. F. R. Lucas, C. Conti, P. Nunes, L. D. Soares, N. M. M. Rodrigues, C. L. Pagliari, E. A. B. d. Silva, and S. M. M. de Faria, “Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC,” in *European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, Sep. 2014, pp. 11–15.

[91] C. Conti, L. D. Soares, and P. Nunes, “Dense light field coding: A survey,” *IEEE Access*, vol. 8, pp. 49 244–49 284, Mar. 2020.

[92] C. Perra, “Lossless plenoptic image compression using adaptive block differential prediction,” in *IEEE International Conference Acoustics Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 1231–1234.

[93] D. Salomon, *Data Compression: The Complete Reference*, 4th ed. New York, NY, USA: Springer-Verlag New York, 2007.

[94] I. Tabus and P. Helin, “Microlens image sparse modelling for lossless compression of plenoptic camera sensor images,” in *European Signal Processing Conference (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 1907–1911.

[95] P. Helin, P. Astola, B. Rao, and I. Tabus, “Sparse modelling and predictive coding of subaperture images for lossless plenoptic image compression,” in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Hamburg, Germany, Jul. 2016, pp. 1–4.

[96] I. Tabus and E. Palma, “Lossless compression of plenoptic camera sensor images,” *IEEE Access*, pp. 31 092–31 103, Feb. 2021.

[97] 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)*, Chiang Mai, Thailand, Jan. 2018, pp. 1–4.

[98] I. Schiopu and A. Munteanu, “Macro-pixel prediction based on convolutional neural networks for lossless compression of light field images,” in *IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, Oct. 2018, pp. 445–449.

[99] ——, “Deep-learning-based macro-pixel synthesis and lossless coding of light field images,” *APSIPA Transactions on Signal and Information Processing*, vol. 8, pp. 1–20, Jul. 2019.

[100] ——, “Residual-error prediction based on deep learning for lossless image compression,” *Electronics Letters*, vol. 54, no. 17, pp. 1032–1034, Aug. 2018.

[101] I. Tabus, I. Schiopu, and J. Astola, “Context coding of depth map images under the piecewise-constant image model representation,” *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4195–4210, 2013.

[102] I. Tabus and P. Astola, “Sparse prediction for compression of stereo color images conditional on constant disparity patches,” in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Budapest, Hungary, Jul. 2014, pp. 1–4.

[103] I. Tabus, P. Helin, and P. Astola, “Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000,” in *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, Sep. 2017, pp. 4567–4571.

[104] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, “Occlusion-aware depth estimation using light-field cameras,” in *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 3487–3495.

[105] 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)*, Copenhagen, Denmark, Jun. 2017, pp. 1–4.

[106] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, May 2012.

[107] I. Schiopu, M. Gabbouj, A. Iosifidis, B. Zeng, and S. Liu, “Subaperture image segmentation for lossless compression,” in *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Montreal, Canada, Nov. 2017, pp. 1–6.

[108] M. U. Mukati and S. Forchhammer, “EPIC: Context adaptive lossless light field compression using epipolar plane images,” in *Data Compression Conference (DCC)*, Snowbird, UT, USA, Mar. 2020, pp. 43–52.

[109] ——, “Epipolar plane image-based lossless and near-lossless light field compression,” *IEEE Access*, vol. 9, pp. 1124–1136, Dec. 2021.

[110] G. N. N. Martin, “Range encoding: an algorithm for removing redundancy from a digitized message,” in *Video and Data Recording Conference*, Southampton, UK, Mar. 1979.

[111] A. J. Pinho, “On the impact of histogram sparseness on some lossless image compression techniques,” in *IEEE International Conference on Image Processing (ICIP)*, vol. 3, Thessaloniki, Greece, Oct. 2001, pp. 442–445.

[112] R. Starosolski, “Compressing high bit depth images of sparse histograms,” *AIP Conference Proceedings*, vol. 1060, no. 1, pp. 269–272, 2008.

[113] W. Pratt, “Spatial Transform Coding of Color Images,” *IEEE Trans. Commun. Technol.*, vol. 19, no. 6, pp. 980–992, Dec. 1971.

[114] T. Strutz, “Adaptive selection of colour transformations for reversible image compression,” in *European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 1204–1208.

[115] R. Starosolski, “New simple and efficient color space transformations for lossless image compression,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1056–1063, 2014.

[116] “Information technology — JPEG XR image coding system — Part 2: Image coding specification,” International Organization for Standardization (ISO), Geneva, Switzerland, Tech. Rep. ISO/IEC 29199-2:2020, May 2020.

[117] JPEG, “JPEG Pleno Light Field Datasets according to common test conditions,” https://jpeg.org/plenodb/lf/pleno_lf/, 2018, [Accessed 16 August 2021].

[118] P. J. S. G. Ferreira and A. J. Pinho, “Why does histogram packing improve lossless compression rates?” *IEEE Signal Process. Lett.*, vol. 9, no. 8, pp. 259–261, Aug. 2002.

[119] 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*, A. G. Tescher and T. Ebrahimi, Eds., vol. 11137, International Society for Optics and Photonics. SPIE, 2019, pp. 402 – 413.

[120] C. Perra, W. Song, and A. Liotta, “Effects of light field subsampling on the quality of experience in refocusing applications,” in *International Conference on Quality of Multimedia Experience (QoMEX)*, Cagliari, Italy, May 2018, pp. 1–3.

[121] C. Conti, P. Nunes, and L. D. Soares, “Inter-Layer Prediction Scheme for Scalable 3-D Holoscopic Video Coding,” *IEEE Signal Process. Lett.*, vol. 20, no. 8, pp. 819–822, Aug. 2013.

[122] C. Conti, L. D. Soares, and P. Nunes, “Light field coding with field-of-view scalability and exemplar-based interlayer prediction,” *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2905–2920, Nov. 2018.

[123] Yun Li, M. Sjöström, and R. Olsson, “Coding of plenoptic images by using a sparse set and disparities,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Turin, Italy, Jun. 2015, pp. 1–6.

[124] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, “Scalable coding of plenoptic images by using a sparse set and disparities,” *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, Jan. 2016.

[125] H. Amirpour, A. Pinheiro, M. Pereira, F. J. P. Lopes, and M. Ghanbari, “Light field image compression with random access,” in *Data Compression Conference (DCC)*, Snowbird, UT, USA, Mar. 2019, pp. 553–553.

[126] N. Mehajabin, S. R. Luo, H. Wei Yu, J. Khoury, J. Kaur, and M. T. Pourazad, “An efficient random access light field video compression utilizing diagonal inter-view prediction,” in *IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 3567–3570.

[127] P. Gomes and L. A. da Silva Cruz, “Pseudo-sequence light field image scalable encoding with improved random access,” in *European Workshop on Visual Information Processing (EUVIP)*, Roma, Italy, Oct. 2019, pp. 16–21.

[128] S. Pratapa and D. Manocha, “RLFC: Random access light field compression using key views and bounded integer sequence encoding,” in *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Montreal, QC, Canada, May 2019, pp. 1–10.

[129] V. Avramelos, J. Praeter, G. Wallendael, and P. Lambert, “Random access prediction structures for light field video coding with MV-HEVC,” *Multimedia Tools and Applications*, vol. 79, p. 12847–12867, May 2020.

[130] R. J. Monteiro, N. M. Rodrigues, S. M. Faria, and P. J. Nunes, “Light field image coding with flexible viewpoint scalability and random access,” *Signal Processing: Image Communication*, vol. 94, p. 116202, May 2021.

APPENDIX A

Contributions

This appendix presents a list of the published papers, resulting from the research work done during this work.

A.1 Journal Publications

J1. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Hierarchical Lossless Coding of Light Fields with Improved Random Access, Signal Processing: Image Communication, vol. 105, pp. 116687-116687, 2022, doi: 10.1016/j.image.2022.116687.

J2. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Coding of Light Fields based on 4D Minimum Rate Predictors, IEEE Transactions on Image Processing, vol. 31, pp. 1708-1722, 2022, doi: 10.1109/TIP.2022.3146009.

J3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Coding of Light Field Images based on Minimum-Rate Predictors, Journal of Visual Communication and Image Representation, Vol. 54C, pp. 21 - 30, April, 2018, ISSN 1047-3203, doi: 10.1016/j.jvcir.2018.03.003.

J4. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Performance evaluation of light field pre-processing methods for lossless standard coding, IEEE COMSOC MMTC Communications - Frontiers, Vol. 12, No. 4, pp. 44 - 49, July, 2017.

A.2 Peer Reviewed Conference Publications

C1. João M. Santos, Lucas A. Thomaz, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Sérgio M. M. de Faria, Disparity compensation of light fields for improved efficiency in 4D transform-based encoders, IEEE International Conference on Visual Communications and Image Processing (VCIP), Macau, China, December, 2020, pp. 112-115, doi: 10.1109/VCIP49819.2020.9301829.

C2. Lucas A. Thomaz, João M. Santos, Pekka Astola, Sérgio M. M. de Faria, Pedro A. A. Assunção, Murilo B. de Carvalho, Eduardo A. B. da Silva, Ioan Tabus, Márcio P. Pereira, Gustavo Alves, Fernando Pereira, Vanessa Testoni, Pedro Freitas, Ismael Seidel, Visually Lossless Compression of Light Fields, EURASIP European Light Field Imaging Workshop (ELFI), Borovets, Bulgaria, June, 2019.

C3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Rate-Complexity Trade-Off in Minimum Rate Predictors Light Field Lossless Encoding, EURASIP European Light Field Imaging Workshop (ELFI), Borovets, Bulgaria, June, 2019.

C4. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless compression of Light Fields using multi-reference Minimum Rate Predictors, Data Compression Conference (DCC), Snowbird, Utah, United States, March, 2019, pp. 408-417, doi: 10.1109/DCC.2019.00049.

C5. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. Távora, Rui Fonseca-Pinto, Sérgio M. M. de Faria, Lossless Light-Field Compression Using Reversible Colour Transformations, International Conference on Image Processing Theory, Tools and Applications IPTA, Montreal, Canada, pp. 1-6, November, 2017, doi: 10.1109/IPTA.2017.8310154.

A.3 JPEG Documents

D1. Lucas A. Thomaz, João M. Santos, Pedro A. A. Assunção, Sérgio M. M. Faria, "WG1M84001-CTQ-JPEG Pleno Light Field ES4.1 IT-LEIRIA partial replacement report", 84th JPEG Meeting, Brussels, Belgium.

D2. Lucas A. Thomaz, João M. Santos, Pedro A. A. Assunção, Sérgio M. M. Faria, "WG1M83038 - C&Q-JPEG Pleno Light Field ES4.1 IT-LEIRIA report", 83rd JPEG Meeting, Geneve, Switzerland.

D3. João M. Santos, Pedro A. A. Assunção, Luís A. da Silva Cruz, Luís M. N. Távora, Rui Fonseca-Pinto, Sérgio M. M. Faria, "WG1M82036 - Lossless compression of Light Fields using multi-reference Minimum Rate Predictors", 82nd JPEG Meeting, Lisbon, Portugal.