COPPE UFRJ

Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

PREDICTIVE CODING ALGORITHMS FOR LOSSY IMAGE AND VIDEO COMPRESSION

Luís Filipe Rosário Lucas

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Sérgio Manuel Maciel de Faria

Rio de Janeiro
Janeiro de 2016

PREDICTIVE CODING ALGORITHMS FOR LOSSY IMAGE AND VIDEO COMPRESSION

Luís Filipe Rosário Lucas

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Sérgio Manuel Maciel de Faria, Ph.D.

Prof. Sergio Lima Netto, Ph.D.

Prof. Ricardo Lopes de Queiroz, Ph.D.

Prof. Fernando Manuel Bernardo Pereira, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2016

Lucas, Luís Filipe Rosário

Predictive coding algorithms for lossy image and video compression/Luís Filipe Rosário Lucas. – Rio de Janeiro: UFRJ/COPPE, 2016.

XXIII, 211 p.: il.; $29,7$cm.

Orientadores: Eduardo Antônio Barros da Silva
Sérgio Manuel Maciel de Faria

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 197 – 211.

1. Image and video compression. 2. Directional intra prediction. 3. Linear prediction. 4. Least-squares optimisation. 5. Sparse representation. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

The only source of knowledge
is experience.

Albert Einstein

Acknowledgements

I would like to express my gratitude to my supervisors Prof. Eduardo da Silva, Prof. Sérgio Faria, Prof. Nuno Rodrigues and Prof. Carla Pagliari, without whom this thesis would not be made possible. I am thankful for their support and wise guidance during this journey; for their suggestions and fruitful discussions providing new insights and inspiring ideas that enriched this research work; for their availability to review and improve the papers and this thesis; and also for their friendship. I would also like to acknowledge Prof. Krzysztof Wegner, for helping to implement the VSO algorithm and for reviewing the journal paper on depth map coding.

I would like to thank Fundação para a Ciência e Tecnologia, which provided the financial support for this PhD. I am also grateful to the Instituto de Telecomunicações in Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria and also to the Laboratório de Sinais, Multimídia e Telecomunicações of Universidade Federal do Rio de Janeiro, which provided the computer resources and laboratory facilities with excellent conditions for my research work in Leiria and Rio de Janeiro, and provided some funds for participating in scientific conferences.

I would like to acknowledge my research group colleagues of IT, namely Sylvain, Lino, Carreira, Ricardo, Santos and André, for the nice working environment, for sharing new ideas, and also for the moments outside the laboratory. I am also thankful to Danillo, Nelson and Diego, for sharing their knowledge and experience, since my first years as researcher. To all my colleagues of SMT-UFRJ, in particular Anderson, Flávio, Tadeu, Luiz and Renam, I express my gratitude for receiving me in Brazil, and for helping me with the statistics and algebra courses.

I also thank all my friends, that accompanied me during this journey, for the distracting and social moments, running nights in Leiria and pleasant bike rides.

I must thank all those people and communities around the world that contribute for the development of many free and open-source tools used in this research work, namely the GNU/Linux/KDE software, distributed by Ubuntu and Gentoo communities. I also acknowledge Carreira for the PlaYUVer software, and its contribution to the Remoterun scripts, which made our lives as researchers much easier.

Last but not least, I would like to thank my family and Ana for the support and love, mainly in those moments I was absent and more involved with this work.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

ALGORITMOS DE PREDIÇÃO PARA COMPRESSÃO DE IMAGEM E VÍDEO COM PERDAS

Luís Filipe Rosário Lucas

Janeiro/2016

Orientadores: Eduardo Antônio Barros da Silva
Sérgio Manuel Maciel de Faria

Programa: Engenharia Elétrica

Nesta tese são investigadas técnicas de predição intra para os atuais algoritmos estado-da-arte de compressão de imagem e vídeo. São propostos vários métodos de predição que permitem reduzir a redundância espacial presente em vídeos 2D e 3D, usando algoritmos baseados em diferentes paradigmas de compressão.

A primeira proposta desta tese consiste em um esquema de predição mais eficiente para o algoritmo *Multidimensional Multiscale Parser* (MMP) baseado em casamento de padrões. Com a utilização de um maior número de modos de predição direcional o algoritmo MMP melhora significativamente, atingindo um desempenho competitivo com os atuais padrões de codificação baseados em transformada.

O estudo de novos métodos de predição para o MMP levou ao desenvolvimento de um novo algoritmo de compressão de mapas de profundidade que combina predição direcional com segmentação flexível e aproximações lineares. Os resultados experimentais, baseados na qualidade das vistas sintetizadas, apresentam ganhos taxa-distorção consistentes sobre o padrão 3D-HEVC (*High Efficiency Video Coding*).

Em relação à predição linear, são apresentadas novas abordagens que usam modelos lineares com restrições de esparsidade. A predição *Locally Linear Embedding* é investigada com sucesso para compressão de imagens holoscópicas usando o padrão HEVC. É também desenvolvido um novo método de predição linear esparsa, que melhora o desempenho do HEVC, em particular para imagens com estruturas complexas e repetidas. Por último, é proposto um novo método de predição generalizado que unifica as predições direcional e linear, com base em modelos esparsos óptimos. Os resultados experimentais mostram a vantagem do novo esquema de predição em relação à atual predição intra do padrão HEVC.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PREDICTIVE CODING ALGORITHMS FOR LOSSY IMAGE AND VIDEO COMPRESSION

Luís Filipe Rosário Lucas

January/2016

Advisors: Eduardo Antônio Barros da Silva
Sérgio Manuel Maciel de Faria

Department: Electrical Engineering

This thesis investigates efficient intra prediction techniques for current state-of-the-art image and video compression algorithms. Several prediction methods are proposed to reduce the spatial redundancy present in 2D and 3D video signals, using several compression algorithms based on different coding paradigms.

The first proposal of this thesis consists in an improved prediction framework for a pattern-matching-based image encoder known as Multidimensional Multiscale Parser (MMP). It is demonstrated that the MMP algorithm significantly benefits from using an increased number of directional prediction modes, achieving a competitive performance in comparison to the transform-based standards.

In the context of the prediction methods investigated for the MMP, a new highly predictive algorithm is presented for depth map coding, combining directional prediction with flexible block partitioning and linear fitting. Based on the quality of the synthesised views, the experimental results show consistent rate-distortion gains relative to the current 3D-HEVC (High Efficiency Video Coding) standard.

Regarding linear prediction, new approaches that enforce sparsity constraints on linear models are presented. The Locally Linear Embedding-based prediction method is investigated for holoscopic image coding using the HEVC standard with successful results. A new sparse linear prediction method is also developed, improving the coding performance of HEVC, particularly for images with complex and repeated structures. Finally, a new generalised intra prediction framework is proposed, unifying the directional and linear prediction methods, based on optimal sparse models. Experimental results show the advantage of the new prediction framework over the current intra prediction of HEVC standard.

Contents

List of Figures

List of Tables

List of Abbreviations

k-NN	k-Nearest Neighbours, p. 29
3DTV	3D Television, p. 57
AAC	Adaptive Arithmetic Coding, p. 52
ACT	Adaptive Colour Transform, p. 48
AIC	Akaike Information Criterion, p. 162
AMP	Asymmetric Motion Partitioning, p. 43
AMR	Adaptive Motion Resolution , p. 48
AMVP	Advanced Motion Vector Prediction, p. 68
AVC	Advanced Video Coding, p. 3
BDPSNR	Bjontegaard Delta PSNR, p. 54
BDRATE	Bjontegaard Delta Rate, p. 54
BIC	Bayesian Information Criterion, p. 162
BMA	Block Matching Algorithm, p. 20
CABAC	Context Adaptive Binary Arithmetic Coding, p. 47
CALIC	Context-based Adaptive Lossless Image Codec, p. 14
CB	Coding Block, p. 42
CDMM	Constrained Depth Modelling Mode, p. 112
CPV	Constant Partition Value, p. 95
CTB	Coding Tree Block, p. 42
CTC	Common Test Conditions, p. 70

CTU	Coding Tree Unit, p. 42
CU	Coding Unit, p. 42
CV	Cross-Validation, p. 162
DBBP	Depth-Based Block Partitioning, p. 68
DBF	Deblocking Filter, p. 41
DCP	Disparity-Compensated Prediction, p. 66
DCT	Discrete Cosine Transform, p. 30, 46
DFT	Discrete Fourier Transform, p. 30
DIBR	Depth Image Based Rendering, p. 62
DLT	Depth Lookup Table, p. 93
DMM	Depth Modelling Modes, p. 93
DPB	Decoded Picture Buffer, p. 45
DPCM	Differential Pulse Code Modulation, p. 11
DST	Discrete Sine Transform, p. 46
DVB-T	Digital Video Broadcasting - Terrestrial, p. 1
DWT	Discrete Wavelet Transform, p. 65
DoNBDV	Depth-oriented Neighbouring Block-based Disparity Vector, p. 68
EDP	Edge-Directed Prediction, p. 14
EN	Elastic Net, p. 160
FTV	Free-viewpoint Television, p. 57
GOP	Group Of Pictures, p. 38
GOSP	Generalised Optimal Sparse Predictors, p. 173
GSC	Geometrical Stopping Criterion, p. 163
HD	High Definition, p. 2
HEVC	High Efficiency Video Coding, p. 2

HVS	Human Visual System, p. 10
IBC	Intra Block Copy, p. 48
JCT-VC	Joint Collaborative Team on Video Coding, p. 40
JPEG	Joint Photographic Experts Group, p. 9
KNP	K Non-zero Predictors, p. 162
LAR	Least Angle Regression, p. 6
LASSO	Least Absolute Shrinkage and Selection Operator, p. 6
LCB	Largest Coding Unit, p. 42
LLE	Locally Linear Embedding, p. 29
LPC	Linear Predictive Coding, p. 13
LSP	Least-Squares Prediction, p. 3, 23
LVD	Layered Depth Video, p. 64
MCP	Motion Compensated Prediction, p. 20
MFV	Most Frequent Value, p. 50
MMP	Multidimensional Multiscale Parser, p. 3
MPEG	Motion Picture Experts Group, p. 2
MPI	Motion Parameter Inheritance, p. 69
MPM	Most Probable Mode, p. 18
MP	Matching Pursuit, p. 29
MRP	Minimum Rate Predictor, p. 14
MSE	Mean Squared Error, p. 54
MVC	Multiview Video Coding, p. 61
MVD	Multiview Video+Depth, p. 62
NBDV	Neighbouring Block-based Disparity Vector Derivation, p. 67
NMF	Non-negative Matrix Factorisation, p. 29

OLS	Ordinary Least-Squares, p. 159
OMP	Orthogonal Matching Pursuit, p. 30
PB	Prediction Block, p. 42
PDC	Predictive Depth Coding, p. 91
POC	Picture Order Count, p. 45
PSNR	Peak Signal-to-Noise Ratio, p. 53
PU	Prediction Unit, p. 42
QP	Quantisation Parameter, p. 46
RD	Rate-Distortion, p. 39
RGB	Red, Green and Blue, p. 10
RMD	Rough Mode Decision, p. 45
ROI	Region Of Interest, p. 92
RPS	Reference Picture Set, p. 45
SAE	Sum of Absolute Errors, p. 100
SAO	Sample Adaptive Offset, p. 41
SCB	Smallest Coding Unit, p. 42
SCC	Screen Content Coding, p. 48
SDC	Segment-wise DC, p. 93
SLSP	Sparse-LSP, p. 138
SSE	Sum of Squared Errors, p. 51
SS	Self-Similarity, p. 66
SVDC	Synthesised View Distortion Change, p. 96
TB	Transform Block, p. 42
TM	Template Matching, p. 3
TU	Transform Unit, p. 42

TW	Training Window, p. 157
TZS	Test Zone Search, p. 46
UHD	Ultra High Definition, p. 1
URQ	Uniform-Reconstruction Quantiser, p. 46
VCEG	Video Coding Experts Group, p. 2, 40
VSO	View Synthesis Optimisation, p. 93
VSP	View Synthesis Prediction, p. 68
VTE	Variation of the Training Error, p. 162
WPP	Wavefront Parallel Processing, p. 44
YCbCr	Luminance, Blue Chrominance and Red Chrominance, p. 10
dB	Decibels, p. 53

Chapter 1

Introduction

This chapter presents an introduction to the research work developed in this thesis. The motivations for proposing new efficient predictive coding techniques are introduced and the outline of the document structure is briefly presented.

1.1 Motivation

Over the last years, the proliferation of digital devices, associated to the ever growing bandwidth availability and speed of digital networks, has increased the demand for digital multimedia contents. Due to the technological advances in consumer electronics, novel devices that capture, store and display visual information with higher resolutions are launched everyday. Simultaneously, most of the existing analogue multimedia systems have migrated to digital technology.

Digital representation presents numerous advantages over the analogue one, including easier processing, storage and transmission of multimedia contents. The amount of digital video applications and services provided through the global Internet network has been increasing continuously, specially with the spread of computer devices and mobile handsets, such as smartphones and tablets. These devices are used by millions of users, allowing them to capture visual information in digital format and immediately publish it into the cloud, for instance, using social networks or video sharing websites. Furthermore, the evolution of mobile networks, that recently entered in fourth generation systems, continues making the Internet even more accessible for portable devices.

However, digital multimedia systems go beyond the Internet scope. Recently, in many countries, the traditional analogue television system has been replaced by digital system television standards, such as the Digital Video Broadcasting - Terrestrial (DVB-T). In regard to television and home entertainment, digital technology has brought a more immersive viewing experience, namely through the use of Ultra High Definition (UHD) and three-dimensional (3D) video systems. Other applications of digital multimedia include video telephony, surveillance, virtual libraries or personal digital cameras and video recorders.

Such a worldwide digital revolution has been enabled not only by the advances of digital devices and infrastructures, but also by the advent of the image and video compression standards. The compression algorithms aim to provide reduced bitrate representations of the multimedia contents with minimum quality losses. Thus, the research of efficient compression algorithms has been a constant challenge over the last years, mainly due to the continuous increase of bitrate requirements. Currently, the UHD and 3D video formats are the most challenging technologies, due to the higher amount of data associated to higher resolutions and multiple views. In order to efficiently cope with such increased bitrate requirements, the compression algorithms exploit the increasingly available processing power and take advantage of the existing parallel processing architectures.

The High Efficiency Video Coding (HEVC) [1, 2] is the most recent video compression standard, developed with focus on two main issues: the compression of increased video resolutions beyond High Definition (HD) format, and the use of parallel processing architectures. The ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Motion Picture Experts Group (MPEG) are the organisations responsible for the standardisation of HEVC, as well as the previous developed compression standards.

Since the H.261 recommendation [3], all the existing video coding standards are based on a hybrid coding architecture, which consists in the picture prediction, transform-based residue coding and entropy coding. The designing details of these techniques is what mainly distinguishes the various video coding standards developed over the last years. In the case of the most recent compression algorithms, the coding techniques tend to be more sophisticated and complex, usually providing a superior rate-distortion performance at the cost of an increased computational complexity.

Picture prediction is extremely important in the design of video coding algorithms, since it provides the main tools to exploit the redundancy present in video signals. Prediction methods are classified either as intra or inter methods, depending whether they exploit the spatial or temporal redundancy, respectively. Nevertheless, prediction methods have experienced significant improvements over the various video coding standard generations. For reducing temporal redundancy, the motion-compensated prediction based on block-matching algorithm has been the main approach adopted by video coding standards. Regarding intra prediction methods, the current solution relies on a directional prediction framework.

Intra prediction methods have been under constant evolution, with the number of available modes increasing significantly in the most recent video coding standard. A noticeable improvement has been observed from the MPEG-2 [4] to the H.264/AVC (Advanced Video Coding) [5, 6] standards. While MPEG-2 only used the DC coefficient prediction, H.264/AVC introduced the Plane mode and eight directional modes in addition to DC prediction. This tendency has been also observed in the recent H.265/HEVC standard [1, 2], which further increased the number of intra directional modes to 35. Due to these improvements on prediction methods, as well as the enhancements on transform-based residue coding and entropy coding techniques, the recent H.265/HEVC standard has shown approximately 50% of bitrate saving, for the same subjective quality, when compared to its predecessor, the H.264/AVC standard.

The importance of prediction techniques for image compression has been also demonstrated with alternative coding paradigms proposed in literature, such as the pattern matching-based Multidimensional Multiscale Parser (MMP) image encoder [7, 8]. Proposals of efficient prediction methods for the MMP algorithm include the directional prediction methods similar to the ones of H.264/AVC [7], the Least-Squares Prediction (LSP) method [9, 10] and the Template Matching (TM) algorithm [11] for intra image coding. Moreover, in the context of a previous research work on stereo image coding using the MMP algorithm [12], the LSP method and the TM algorithm were proposed for inter prediction, specifically for disparity compensation.

All these observations, related to the importance and evolution of image and video prediction methods in the state-of-the-art video compression algorithms, encouraged the research of new improved prediction methods, aiming to provide new performance benchmarks for generic image and video compression, and responding to the most recent challenges on 3D video compression.

1.2 Main objectives and contributions

With the advent of 3D video formats and higher video resolutions, new video compression algorithms have been developed in the last years and further research continues targeting new applications and content types. The predictive methods have shown to be an important component in the design of image and video compression algorithms, becoming increasingly accurate over the video coding generations, usually at the cost of a higher computational complexity.

In this context, this research work addresses the increasing necessity to further compress image and video multimedia contents, including the 2D and recent 3D content formats. The main objective of this thesis is the investigation of new prediction algorithms for 2D and 3D image coding, using the existing state-of-the-art image encoders, namely the MMP algorithm and the HEVC standard. The investigated algorithms are based on directional prediction, as well as on more computationally complex solutions, namely the linear prediction and sparse representation techniques. Beyond still image coding application, this research work addresses the compression of stereo images, holoscopic images, and depth map data as part of the multiview video plus depth format.

The most relevant contributions of this thesis, related to the presented objective, can be summarised as follows:

- **Improving the MMP algorithm for still and stereo image coding using predictive methods.**

  The first contribution of this research work is an improved intra prediction scheme for the Multidimensional Multiscale Parser algorithm. Previous research works using MMP algorithm have demonstrated the importance of intra predictive methods for this coding paradigm, which has shown a superior coding performance in comparison to the H.264/AVC standard, for still and stereo image coding. The recent developments for the HEVC standard, namely the ones regarding the intra prediction framework, motivated the proposed improvements to the MMP algorithm. Instead of the eight directional modes, the proposed MMP algorithm uses 33 angular prediction modes, based on the ones of the HEVC standard. A larger block size and improved block partitioning tree were also investigated for the MMP algorithm.

  Experimental results demonstrated the advantage of the coding tools introduced in MMP algorithm for the compression of still images. Regarding the stereo image coding scenario, the new MMP algorithm presented very competitive results relative to the state-of-the-art MV-HEVC standard. For some images, mostly synthetic ones, the MMP algorithm was able to overcome the MV-HEVC standard. This work on predictive methods for the MMP algorithm resulted in the conference paper C3 and it constitutes part of the submitted paper S2 (see Appendix B).

- **The Predictive Depth Coding: a new highly predictive algorithm for depth map coding.**

  In the context of the multiview video plus depth format, a new algorithm, named by Predictive Depth Coding (PDC), was developed in this thesis for the compression of depth maps. The PDC algorithm was motivated not only by the issues of the existing transform-based algorithms for the compression of depth maps, but also by the efficient predictive capabilities observed for the directional prediction framework proposed for the MMP algorithm. The use of directional intra prediction with the flexible block partitioning scheme of MMP algorithm, which recursively divides one block either in horizontal or vertical direction, showed to provide an efficient representation for depth map signals.

Thus, the proposed PDC algorithm combines a flexible intra predictive scheme with a straightforward residue coding method based on linear approximation. Beyond the directional prediction framework based on 33 modes, the PDC algorithm may use a simplified depth modelling mode that represents depth edges that cannot be well predicted by angular modes. The whole encoder was designed to minimise its computational complexity, while keeping an efficient depth map compression performance.

Experimental results comparing the PDC algorithm relative to the state-of-the-art 3D-HEVC standard using all-intra configuration for depth map coding, demonstrated the superiority of PDC when the quality of the texture views synthesised using the encoded depth maps is evaluated. Furthermore, an experiment validating the high potential of PDC to improve the 3D-HEVC standard under the common test conditions was performed by using PDC to encode the anchor I-frames of 3D-HEVC. This research topic resulted in a journal paper J1, an MPEG document D1 and conference papers C1, C2, C4 and C6 (see Appendix B).

- **Developing an efficient 3D holoscopic image encoder based on HEVC and sparse representation methods.**

  The main limitation of directional intra prediction methods is that only the first line of samples defined along the left and top margins of the block can be exploited in order to generate the prediction output. Due to these characteristics, directional prediction tends to present some issues for predicting complex areas made of repeated patterns, such as the high redundant structure of micro-images observed in 3D holoscopic content. Thus, in order to exploit useful information from a larger causal region for block prediction, dictionary-based methods using sparse representations have been proposed in literature.

  One of these prediction methods, studied in this research work is known as Locally Linear Embedding (LLE). Due to its ability to exploit correlations between pixels at larger spatial distances, we investigated LLE method for the compression of 3D holoscopic images. A full encoder based on the HEVC technology and LLE-based prediction resulted from this work.

  Experimental results using the developed method for 3D holoscopic image coding demonstrated its advantage over other HEVC-based algorithms proposed in literature for the same purpose. The developed algorithm and achieved results were described in the conference paper C5 (see Appendix B).

- **The sparse-LSP: a least-squares prediction method using adaptive sparse models.**

  In the context of the LLE-based prediction method, an improved linear prediction technique for generic image coding was proposed in this thesis. While the LLE-based prediction method is able to exploit previously encoded information based on patches extracted from a causal search window, it does not allow to predict one sample from the closer causal neighbouring samples, similarly to traditional least-squares-based linear prediction methods. Thus, we developed the sparse-LSP algorithm as a new linear prediction method based on adaptive sparse models, that extends the LLE-based prediction method.

  The sparse-LSP and LLE-based prediction methods are closely related because they use the same procedure for predictor selection. The main advantage of sparse-LSP is the ability to select closer neighbouring predictors in the causal area, providing a mixed functionality between LLE and traditional LSP methods. The experimental tests using sparse-LSP method in the HEVC standard, with a fixed number of predictors, presented significant improvements for still image coding, specially for images presenting complex textures and repeated structures. This research work resulted in the conference paper C7 (see Appendix B).

- **Developing a new intra prediction framework based on generalised optimal sparse predictors.**

  The last contribution of this thesis is a new intra prediction framework that generalises the previous investigated prediction methods, namely the directional prediction and the sparse-LSP. We presented an alternative interpretation of directional intra prediction, based on first-order linear filters combined with geometric transformations, which served as basis for the proposed prediction solution.

  The result of this research work is a prediction framework based on sparse models optimally estimated in a causal training area, improved by a geometric transformation procedure that is explicitly selected and signalled to the decoder. In order to select and estimate the sparse linear models several algorithms were investigated, such as Matching Pursuit, Least-Angle Regression (LAR) and Least Absolute Shrinkage and Selection Operator (LASSO).

  Experiments using the developed prediction framework in HEVC standard by replacing the angular prediction modes demonstrated the advantage of this method to exploit the causal information in the prediction process. When compared with other methods, the developed solution provided a superior coding performance for most images. This investigation work was described in the submitted paper S1 (see Appendix B).

1.3 Outline

This thesis is organised as follows. The current chapter introduces the main research topics of this thesis, describing the motivations and objectives to achieve with this work, as well as the original contributions of this thesis.

Chapter 2 describes some of the most relevant predictive coding techniques reported in the literature for image and video compression and used along this research. These techniques include the current prediction methods used in state-of-the-art video coding standards, namely the directional intra prediction and the motion-compensated prediction of the HEVC standard. A description of the most important linear prediction and sparse representation methods investigated in this thesis is also presented.

Chapter 3 presents a brief overview of the image and video coding algorithms used in this thesis, in particular the current state-of-the-art HEVC standard and the MMP algorithm. In this chapter, the main 3D video technologies are also presented and the recent standardised 3D-HEVC codec is briefly described. Some benchmark experiments illustrating the performance of the existing methods for 2D and 3D video coding are presented.

Chapter 4 investigates an improved intra prediction framework for the MMP algorithm, for both intra and stereo image coding applications. The motivation for this work is to increase the prediction accuracy of MMP algorithm, mainly for high resolution images. The experimental tests for intra image coding demonstrate the advantage of the proposed methods, that allow for a consistent improvement on the MMP performance. The proposed MMP algorithm is also compared with the state-of-the-art transform-based standards for stereo image coding.

Chapter 5 presents a novel algorithm for the compression of depth maps, which is denominated as Predictive Depth Coding algorithm. In its essence, the PDC algorithm combines the directional intra prediction framework with a highly flexible block partitioning scheme for efficient representation of depth maps. An additional depth modelling mode, designed to represent some edges difficult to predict by directional modes, is used in PDC algorithm. The residual signal of PDC is represented using a straightforward linear fitting method. Experimental results show the advantage of PDC over the state-of-the-art 3D-HEVC standard for depth map coding. The compression efficiency is evaluated based on the quality of the synthesised texture views using the encoded depth maps.

Chapter 6 proposes new prediction methods based on sparse representation and least-squares prediction techniques, to improve the intra prediction framework of the current state-of-the-art HEVC standard. First, we investigate an intra prediction method for 3D holoscopic image coding using the HEVC standard, known as Locally Linear Embedding (LLE). Experimental results demonstrate the ability of LLE-based prediction to exploit the highly redundant structure of 3D holoscopic images which is based on a regularly spaced array of micro-images. As a second proposal, this chapter presents a new least-squares-based algorithm that extends the LLE-based prediction method. The developed prediction method is equivalent to block-based LSP method using a large filter context, subject to a sparsity constraint that limits the number of non-null coefficients. The used predictors, i.e. non-null coefficients, are selected using the same procedure of LLE method. The proposed sparse-LSP method is implemented in HEVC standard and evaluated for natural image coding, being compared with other related prediction methods.

Chapter 7 presents a new intra prediction framework that generalises the prediction techniques investigated along this thesis, namely the directional and the sparse linear prediction methods. We show that intra prediction can be viewed as a particular case of linear prediction, when first-order linear filters are combined with geometric transformations. The proposed generalised prediction algorithm depends on sparse linear models that are optimally estimated using several techniques, such as Matching Pursuit, Least Angle Regression or LASSO. We evaluate the generalised intra prediction framework for still image coding, by replacing the directional prediction framework of HEVC standard with the proposed one. Experiments demonstrate the advantage of the proposed intra predictive solution over the directional framework, as well as other intra prediction methods proposed in literature.

Chapter 8 presents the main conclusions regarding the developed work, as well as the main topics considered for future work.

Appendix A illustrates some of the original test signals used in simulations throughout this thesis, namely test stereo images, grayscale natural test images and holoscopic test images.

Appendix B presents a summary of the published papers, describing the contributions of the research work of this thesis.

Chapter 2

Prediction techniques for image and video coding

Prediction techniques have been widely investigated over the last decades to be used in efficient image and video compression schemes. Almost all systems for image and video compression include some kind of predictive coding. For example, the well-known, 25 year old, Joint Photographic Experts Group (JPEG) image coding standard, predictively encodes the quantised DC transform coefficients. Also in the case of video sequences, compression systems typically exploit the temporal redundancy through the use of motion-compensated prediction.

This chapter describes some of the most important predictive coding techniques used in this thesis, which have been reported in literature and employed in the state-of-the-art image and video encoders. Section 2.1 provides some basic concepts about digital video representation. Section 2.2 presents a general overview on predictive coding for image compression. In Section 2.3, the most important prediction techniques used in the current state-of-the-art video coding standards are described. Section 2.4 presents various linear prediction methods based on least-squares optimisation, while Section 2.5 describes some predictive methods inspired on sparse representation and dimensionality reduction.

2.1 Digital video representation

A digital video sequence is a temporal succession of digital images, also referred to as pictures or frames, which usually presents a significant amount of temporal and spatial correlations. Typically, each image has a rectangular dimension, also called spatial resolution, and it is organised as a matrix of numerical values, denominated by pixels, which means picture elements.

An important aspect related to digital images is the colour representation. In general, image sensing systems generate colour images based on the RGB (Red, Green and Blue) model. In this model, image colours are generated through a linear combination of red (R), green (G) and blue (B) primary colours. The colour image is thus formed by three equally sized pixel matrices, which represent the amount of each primary colour at each pixel position. This contrasts to the monochromatic image representation, which only uses one image channel, whose pixels correspond to the brightness intensity at each position.

One of the main issues of RGB system is the spectral redundancy that usually exists between pixels of the three primary colour channels. In order to provide a more efficient representation of colour images, the YCbCr (Luminance, Blue Chrominance and Red Chrominance) colour system, also known as YUV, has been proposed in ITU-R BT.601 Recommendation [13]. The idea of the YCbCr system is to concentrate the image spatial information in the luminance (Y) component and use a reduced bandwidth for the chrominance (Cb and Cr) components. This colour system exploits the fact that the human visual system (HVS) is much more sensitive to luminance (or luma) information than to colour information, represented by chrominance (or chroma) components.

A common approach used to achieve reduced bandwidth for the chrominance components is through sub-sampling. Chrominance sub-sampling reduces the spatial resolution of colour components, providing a more compact representation of image or video data. Although YCbCr sub-sampling causes unrecoverable loss of colour information, this should not be noticeable by the HVS since it is less sensitive to colour information. The most common sub-sampling patterns are the 4:2:2 and the 4:2:0. The 4:2:2 pattern uses two Cb and two Cr samples for each group of four luma samples, while the 4:2:0 pattern uses one Cb and one Cr samples for each group of four luma samples. In this thesis, we will assume the 4:2:0 sub-sampling pattern, whenever we refer to the YUV colour format.

Due to its losses, the YCbCr colour system can be viewed as a compression technique that removes the irrelevancy associated to the colour information. In order to better represent image and video data, other compression methods that exploit data irrelevancy and also redundancy can be used. While the irrelevancy is related to the information that can be removed without being perceived by the HVS, the redundancy has a statistical nature and its removal does not cause any loss in the original signal. Video compression methods that only exploit data redundancy are denominated lossless compression algorithms, while the coding approaches that exploit both redundancy and irrelevancy are the so-called lossy encoders.

In lossless image and video coding, the compressed signals can be completely recovered after the decoding process, *i.e.* the original and reconstructed signals are exactly the same. Generic lossless video compression algorithms not only exploit the spatial and temporal redundancies of video signals, but also the entropic redundancy. The spatial or intra-frame redundancy of one image is associated to the correlation between neighbouring pixels in the same image, tending to increase for higher spatial resolutions. Temporal or inter-frame redundancy is related to the existing similarities between adjacent temporal frames. Video sequences present higher temporal correlations when their contents are constant in time or the motion in the scene is slow. Such correlations also tend to increase for sequences with higher temporal frame rate, because the motion effect between two consecutive frames is smoother. Entropic redundancy is related to the statistics of the symbols transmitted by the image and video compression algorithms. Symbols with higher probability of occurrence may be more efficiently represented using shorter binary codes.

In addition to redundancy coding, the lossy compression algorithms also exploit data irrelevancy, in order to further compress image and video signals. Contrary to the redundancy coding, the removal of data irrelevancy is not a reversible process, being responsible for the coding losses associated to lossy image and video compression algorithms. Such loss of information is accepted in favour of a much higher compression ratio, as long as the video quality is not significantly degraded. The efficiency of lossy video coding algorithms depends on their ability to remove both the redundant and irrelevant information present in image and video signals, while maintaining an acceptable video quality. This is measured by considering both the compression ratio and the quality of the reconstructed signal.

2.2 Image prediction overview

The prediction process consists of a statistical estimation of future random variables from past and present observable random variables. In image compression scenario, the prediction of an image pixel or a group of pixels may be derived from the previously transmitted pixels. The predicted pixels are subtracted from the original pixels and a residual signal is obtained. In general, the prediction is considered successful if the energy of the residual signal is lower than the energy of the original signal. As a consequence, the residual pixels might be more efficiently encoded by entropy coding algorithms, being transmitted with fewer bits.

Since the early days of image compression, predictive coding has showed to be an effective technique, being widely used for both lossless and lossy image coding. A generic compression approach commonly used to describe predictive coding is known as Differential Pulse Code Modulation (DPCM) [14]. This method consists in transmitting the difference between a pixel and its prediction instead of directly transmitting the pixel's value. The main motivation for DPCM is the fact that most source signals, *e.g.* image and video signals, present a high correlation between

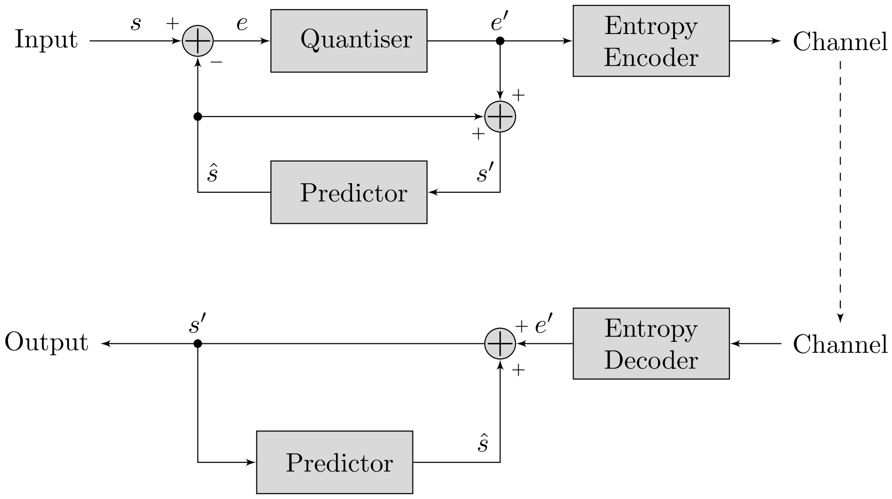


Figure 2.1: DPCM block diagram for lossy image compression.

adjacent pixels.

A representative block diagram of DPCM approach for lossy image compression is outlined in Figure 2.1. The basic principle of DPCM is to predict the current pixel from the previous pixel (or multiple pixels) and to encode the prediction error (or residual) resulting from the difference between the original and predicted pixels. In a lossy coding approach, the residue signal (represented in Figure 2.1 by e) is encoded by means of a scalar quantiser which converts a predefined range of values into a single quantised value. In a final stage the entropy coding is used to efficiently represent the quantised residual (e') into fewer bits, by exploiting the existing statistical redundancy [15]. The compressed quantised residual is transmitted by the encoder and received by the decoder. After entropy decoding the residue signal, the reconstructed signal is obtained by summing the decoded quantised residue signal (e') with the prediction signal ($\hat{s}$), estimated from previous decoded pixels. All the encoding and decoding processes are performed in a pixel-by-pixel basis.

The compression performance of DPCM depends on the accuracy of the used prediction technique. An example of a spatial prediction technique for the pixel-based DPCM approach processed in raster scan order is illustrated in Figure 2.2, where the pixel $X(n)$ is predicted based on a linear combination of the neighbouring pixels indicated on the gray squares. This kind of prediction, based on the linear combination of the previously decoded pixels, is known as linear prediction [16, 17]. Since the coding procedure is done in raster scan order, both encoder and decoder can use the pixels $X'(n-1)$, $X'(n-2)$, $X'(n-3)$ and $X'(n-4)$, available in current and previous pixel rows, in order to generate the prediction for the current pixel, $X(n)$. The decoded (or reconstructed) pixels should be used instead of the original ones, in both encoder and decoder sides, because the decoded pixels may differ from the original pixels due to the quantisation step. Since the original pixels

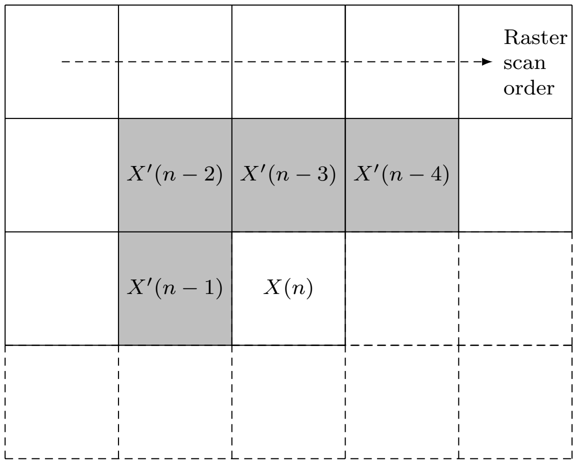


Figure 2.2: Spatial prediction in a raster scan order encoder, *e.g.* DPCM.

are not available in the decoder side, the use of decoded pixels guarantees that the prediction signal is identical in both encoder and decoder sides. Otherwise, if the encoder used the original pixels for prediction, a cumulative mismatching error could be observed between the decoded pixels at the encoder and decoder, leading to an increased distortion in the decoder reconstructed image.

Linear Predictive Coding (LPC) techniques have been extensively studied in literature and successfully used in image, speech and audio processing applications. For the case of image compression, several algorithms based on DPCM, using an adaptive approach of LPC have been presented since the early days of digital image compression [17]. The use of adaptive prediction schemes for image compression has shown to provide better prediction accuracy than the fixed LPC approach. This is mainly because the adaptive methods try to adapt the predictor parameters to the changing local statistics, that characterise the image signal. The main challenge of adaptive prediction is to keep an affordable computational complexity.

For example, adaptive prediction can be achieved by testing a set of different predictors in the encoder and signalling the chosen predictor to the decoder, by using extra bits. However, the trade-off between the prediction efficiency gains and the extra bits required to signal the chosen predictor must be carefully evaluated. Alternatively, implicit approaches, which do not require extra signalling bits, also can be used for adaptive prediction. A discussion on implicit linear prediction techniques for lossy image compression using least-squares optimisation is presented in Section 2.4.

Pixel-based compression schemes using adaptive linear predictive coding techniques have also been successfully applied in lossless image compression. A basic lossless DPCM system can be designed by discarding the quantisation step in the block diagram of Figure 2.1. In such lossless scenario, the prediction stage may directly receive the causal original pixels, since they are equal to the decoded pixels.

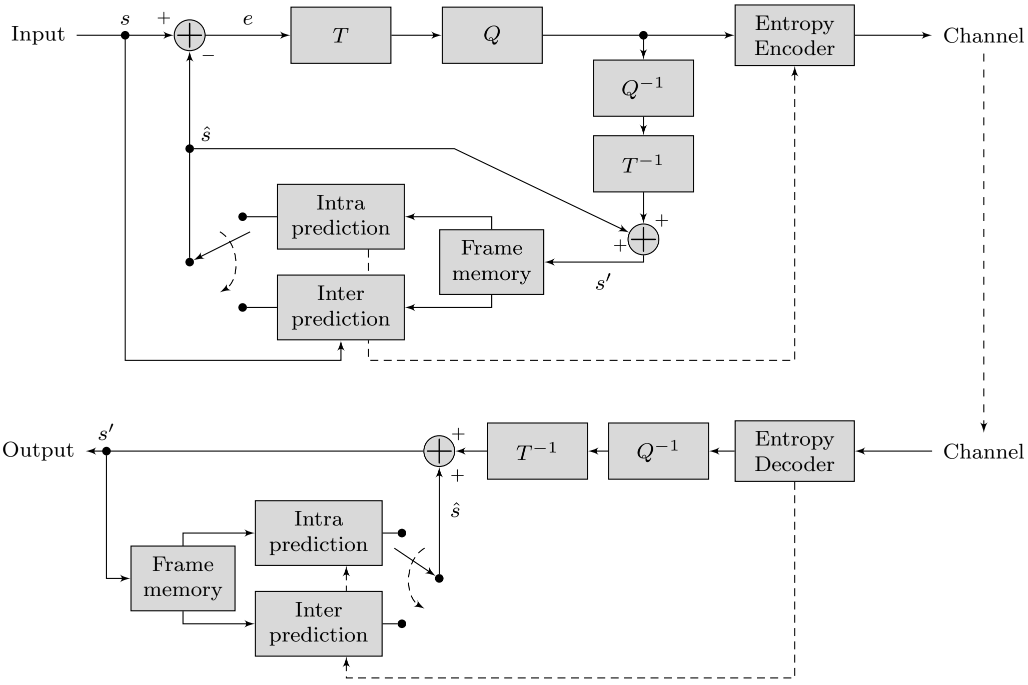


Figure 2.3: Block diagram for a generic transform-based hybrid video compression system.

Some examples of popular pixel-based lossless compressors using adaptive linear and non-linear predictors are the JPEG-LS standard [18], Context-based Adaptive Lossless Image Codec (CALIC) [19], Minimum Rate Predictor (MRP) [20] and Edge-Directed Prediction (EDP) [21].

Although DPCM compression is commonly associated to spatial or intra-frame coding, it can be designed to exploit inter-frame similarities, such as temporal redundancy in the case of video signals. While intra-frame coding usually uses the causal neighbouring pixels belonging to the same frame, the inter-frame coding can be conducted using the causal pixels belonging to another frame, *e.g.* the co-located pixel in the previous coded frame.

Alternatively to the pixel-based approach, DPCM can operate with vectors. In a vector-based DPCM, image blocks can be used as input vectors and the prediction can be performed using block-based methods, such as directional prediction or motion-compensated prediction. These prediction methods, used in the state-of-the-art image and video coding standards, are described in Subsection 2.3. In fact, the vector-based formulation of DPCM works quite similar to the typical transform-based image and video compression standards. The main difference refers to the residue quantiser that is replaced by transform coding and quantisation methods.

Figure 2.3 presents the block diagram for a generic block-based hybrid video compression system using predictive coding and residue coding based on transform (T) and quantisation (Q) methods. Apart from the residue coding module, which may be based on a different coding paradigm, the illustrated block diagram represents the architecture of most image and video compression algorithms studied and developed along this thesis.

The advantage of reducing spatial and temporal redundancies using prediction methods is directly related to the probability distribution of the residual signal. Unlike the original signal, the residue typically presents a highly peaked probability distribution, centred around zero, which is favourable for statistical coding techniques. Regarding the transform coding methods, such probability function is advantageous because the transformed residual presents a lower energy, resulting in less non-null coefficients and producing a more compact representation in the frequency domain. This transform-based representation tends to present a lower entropy being more efficiently encoded with fewer bits.

Besides the prediction of original image pixels, most of image and video compression algorithms use predictive coding to efficiently encode other symbols generated during the encoding process. For example, the motion vectors generated by motion-compensated prediction can be differentially transmitted as a lossless DPCM system. Such predictive coding approach is advantageous because motion vectors typically present high correlation between neighbouring blocks, specially in situations of camera panning.

2.3 State-of-the-art prediction methods

The prediction methods play an important role in image and video coding standards, due to their ability to reduce the signal redundancy based on the previously encoded samples. With the evolution of compression standards, more complex and sophisticated predictive coding techniques have been investigated and adopted. The most recent state-of-the-art H.264/AVC [5, 6] and H.265/HEVC [1, 2] standards use both intra and inter prediction methods. Although these algorithms have been primarily conceived for video compression applications, they also present efficient results for still image coding [22]. In the image coding scenario, only intra prediction methods can be used.

This section describes the main prediction techniques used in the state-of-the-art H.265/HEVC standard, and highlights some differences in relation to the previous H.264/AVC encoder. These techniques include the intra directional prediction for efficient spatial redundancy coding and the motion-compensated prediction for inter-frame temporal redundancy coding. The advantage of these techniques is not only associated to their prediction efficiency but also to their reasonable computational complexity, which make them appropriate for practical applications.

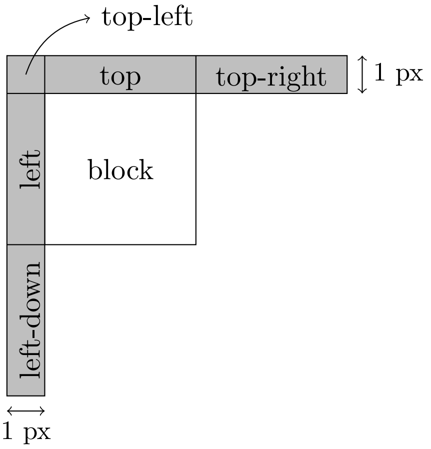


Figure 2.4: Available neighbouring regions used by intra prediction modes in HEVC standard.

2.3.1 Intra-frame prediction

In recent video coding standards, block-based intra prediction techniques are used for efficient spatial redundancy exploitation. Intra predicted blocks are generated based on the pixels of the previously encoded and reconstructed blocks. Directional prediction plus Planar and DC modes were adopted in HEVC standard for luma component prediction. These prediction modes are defined for square block sizes from 4×4 up to 32×32. The encoder tests all the available intra prediction modes and signals the best mode to the decoder. The decoder must generate an identical prediction block, based on the reconstructed pixels from the previously encoded neighbouring blocks.

The reconstructed reference samples used in the prediction process belong to the neighbouring blocks at left-down, left, top-left, top and top-right positions, as shown in Figure 2.4. For an $N \times N$ block, intra prediction requires a row of $2N$ top neighbouring samples, a column of $2N$ neighbouring samples and the top-left corner neighbouring sample, totalling $4N + 1$ reference samples. For some blocks, the left-down or top-right neighbouring blocks may not be available. This also can occur for the top or left neighbouring blocks, *e.g.* at slice or tile boundaries. In order to solve the unavailability issue, HEVC substitutes the unavailable reference samples with the closer available reference sample values, so that all intra prediction modes can be used.

Directional prediction modes

Directional intra prediction was first introduced in the H.264/AVC standard [5]. The idea of directional prediction is to estimate regions with a structured texture or directional edges. In its process, directional prediction projects the reconstructed

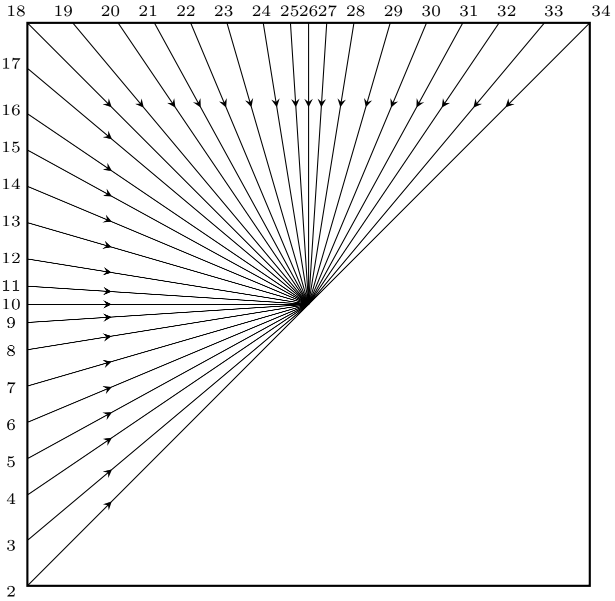


Figure 2.5: Intra prediction angular modes used in HEVC standard.

samples in the block neighbourhood along a specific direction. While H.264/AVC defines 8 possible directional modes, HEVC uses a more sophisticated scheme based on 33 directions, denominated as angular modes [1]. The 33 angular modes in HEVC offer an increased number of possible prediction directions and are able to better predict the larger block sizes available in HEVC. Figure 2.5 illustrates the available prediction directions in the HEVC standard, numbered from 2 to 34.

Angular prediction modes were designed to better cover the angles near the horizontal and vertical directions, being sparser near the diagonal directions. Such angle distribution was observed to be more frequent and useful for prediction. When the projected samples need to be extrapolated, HEVC uses bilinear interpolation of the two nearest samples at integer location with 1/32 sample accuracy.

Planar and DC prediction modes

In addition to the directional modes, HEVC implements the Planar and DC prediction modes, which were also used in H.264/AVC standard. These modes were designed to efficiently predict smooth regions, by using constant or linear planes.

DC mode uses the average value of the neighbouring reference samples to generate a constant prediction for the block. The planar mode works by predicting the block through an amplitude surface with vertical and horizontal slopes derived from the neighbouring reference samples.

Table 2.1: Reference sample smoothing in HEVC standard based on block size and intra prediction modes.

Block size	Intra prediction mode
4×4	No smoothing
8×8	Planar, Angular 2, 18 and 34 smoothing
16×16	Planar and all Angular except 9, 10, 11, 25, 26 and 27 smoothing
32×32	Planar and all Angular except 10 and 26 smoothing

Sample smoothing

HEVC applies sample smoothing in two distinct situations to improve intra prediction results, namely the filtering of reference samples and the filtering of the predicted samples. In H.264/AVC standard, only the reference samples are filtered. In HEVC, this filtering is done in an adaptive manner, depending on the block size and intra prediction mode. Table 2.1 presents the cases in which reference sample smoothing is applied. HEVC uses the same three-tap [1 2 1]/4 filter, as defined in H.264/AVC standard.

The other filtering proposed in HEVC is used to smooth the discontinuities generated along block boundaries due to intra prediction. This filter is applied over the boundary predicted samples for the DC, Angular 10 and Angular 26 modes. In case of DC mode, when the block is smaller than 32×32, the samples of both the first row and first column of the intra predicted block are filtered. A two-tap [3 1]/4 filter fed by the original value and the adjacent reference sample is used. In the case of Angular 10, only the samples of the first column are changed by adding half of the difference between the adjacent reference sample and the top-left reference sample. A similar filtering is done for Angular 26 mode, but only for the first row of predicted samples.

Prediction mode coding

An important aspect in the design of a predictive method is the need for signalling of additional information in the bitstream. In the case of the presented intra predictive scheme for luma prediction, the encoder should choose the best mode and signal it to the decoder.

To better encode the prediction modes, the recent video coding standards estimate the Most Probable Modes (MPMs), which allow to use fewer bits when selected. In the case of H.264/AVC, only one MPM is used, being adaptively initialised from the neighbouring blocks for each predicted block. Due to the increased number of prediction modes, HEVC selects three MPMs to be more efficiently represented. The MPMs are chosen according to the used prediction modes in the top and left

Table 2.2: Assignment of the most probable prediction modes in HEVC.

Block neighbouring modes: A and B	**Most probable modes**		
	1st	2nd	3rd
A != B	A	B	DC, Ang.10, Ang.26
A = B = DC, Planar	A	Planar, DC, Ang.26	Planar, DC, Ang.26
A = B = Angular	A	Angular closest to A	Angular closest to A

neighbouring blocks.

The initialisation of MPMs in HEVC is shown in Table 2.2. It is possible to observe that prediction modes from left and top neighbouring prediction blocks are represented as modes A and B, respectively. When a neighbouring prediction block is unavailable, the DC mode is used instead. If modes A and B are different, they become the two first MPMs, otherwise only the first MPM is assigned. The remaining MPMs, specifically the third one (and also the second one when $A = B$), are defined in such a way that the same mode is not used twice in MPMs list. Table 2.2 explicitly defines a small set of mode candidates that can be used for the third MPM (and also the second MPM when $A = B$). In these cases, the MPM is assigned to the first candidate that does not result in a duplicated MPM.

For encoding purposes, HEVC transmits one flag indicating whether the chosen prediction mode matches any of the MPMs or not. If the matching occurs, the index of the matched MPM (1 out of 3 possible values) is entropy coded and transmitted. Otherwise, a 5-bit fixed length code is used to signal one of the remaining 32 modes (35 modes minus 3 MPMs) to the decoder.

Chroma intra prediction

For chroma component intra prediction, HEVC considers five prediction modes: Planar, Angular 26 (vertical), Angular 10 (horizontal), DC and another mode denominated as Derived mode. The objective of the Derived mode is to use for chroma prediction the same mode as the co-located luma block. Despite allowing only 5 modes, the Derived mode gives a chance to predict the chroma block using any of the 35 previously presented modes for luma prediction.

Derived mode relies on the assumption that corresponding luma and chroma blocks are highly correlated. This is so because some image features, like edges or textures, can be observed in both components. Thus, the Derived mode provides an efficient solution to use the directional modes in chroma component prediction, exploiting some inter-component correlation and using less bits to signal the prediction mode.

For prediction mode signalling, HEVC does not use the MPM approach for chroma component. One flag is transmitted indicating whether or not the chosen chroma prediction mode is the Derived mode. When the chosen mode is not the Derived mode, HEVC uses a 2-bit fixed length code to signal one of the four remaining modes. If any of these remaining modes matches the Derived mode, HEVC replaces it by Angular 34 mode.

2.3.2 Inter-frame prediction

The efficiency of video coding algorithms highly relies on inter-frame predictive techniques used to reduce temporal redundancy. The underlying idea of inter-frame prediction is to estimate the target frame from one or more previously encoded reference frames using block-based Motion Compensated Prediction (MCP). The most common technique used for motion estimation in H.265/HEVC, as well as in H.264/AVC, is the Block Matching Algorithm (BMA).

Inter-frame prediction is also used for disparity compensation in stereo and multiview video coding applications [23, 24]. In addition to temporal redundancy, the stereo and multiview video systems present inter-view redundancy among the multiple views. Similar inter prediction techniques, typically based on BMA, are used for both disparity and motion estimation. Despite being based on BMA, inter-frame prediction methods have received significant improvements, mainly in the most recent video coding standards. In the following, the state-of-the-art techniques proposed for MCP in HEVC standard are discussed.

Motion compensation using Block Matching Algorithm

Motion estimation consists in searching for the causal block with highest similarity with the target block to be predicted, using BMA over the previously encoded frames, *i.e.* the reference frames. Typically, reference frames correspond to past or future temporal frames. The frame encoding order determines which reference frames are selected and whether future frames are available.

In its procedure, BMA subtracts the target block from each equally sized block that exists in a search window defined in the reference frame and computes the resulting difference errors. The reference frame block that produces the lowest matching error is used as prediction to the target block. The motion vector is given by the offset between the best matched block position and the co-located position of the target block. This vector is signalled to the decoder, in order to be able to use the same predictor.

HEVC allows either one or two motion vectors to be chosen to compensate the target block, in a process denominated uni-predictive or bi-predictive coding, respectively. In the case of bi-predictive coding, the average of two motion compensated blocks is performed. These vectors are obtained from frames of the two available reference frame lists. In addition to the motion vectors, HEVC transmits the indices of the used reference frames and, in some cases, the reference picture list associated to each index. Additionally, HEVC allows an enhanced prediction technique, called weighted prediction, which is also present in H.264/AVC standard. This technique consists in applying a scaling and offset to the predictor block. In HEVC, only the explicit mode is used, in which the scale and offset parameters are transmitted to the decoder.

BMA has been evolving over the various video coding standard generations. While first block-based MCP approaches used a fixed block size, more recent algorithms introduced the adaptive block size. The H.264/AVC standard uses adaptive block size with seven possibilities from 4×4 up to 16×16 pixels for MCP, including square and rectangular blocks generated by symmetric block partitioning. A significant evolution is visible on HEVC standard, that uses much more block sizes from 4×4 up to 64×64 pixels, including square and rectangular blocks generated by symmetric and asymmetric block partitioning (see Subsection 3.2.1).

By using larger prediction blocks, HEVC is able to save more bits in the signalling of the chosen motion vector, because more pixels can be predicted using a single translational motion vector. At more complex textured regions, large blocks may not produce efficient prediction results and a more precise motion representation may be required. For these cases, the HEVC block partitioning gives rise to a large number of possible block sizes that may use different motion vectors, leading to better prediction results. However, such flexible block partitioning has the cost of an additional computational complexity and increased bitrate requirements to signal the chosen partitioning tree and motion information associated to each sub-block.

Besides the use of adaptive block size, MCP algorithms adopted in most video coding standards have received some improvements in the motion vector accuracy. Fractional interpolation has been used to generate non-integer sample positions at the reference frames used for motion estimation. Both H.264/AVC and HEVC standards use motion vectors up to quarter-pixel accuracy for luma component. For chroma component, motion vector accuracy depends on the sampling format, being eighth-pixel accuracy for the 4:2:0 sampling format.

In order to generate the luma interpolated pixels, HEVC uses an eight-tap filter for half-pixel positions and a seven-tap filter for quarter-pixel positions, in contrast to the H.264/AVC interpolation based on a six-tap filter for half-pixel positions and two-pixel averaging for quarter-pixel positions. For chroma pixel interpolation, HEVC uses four-tap filters, while H.264/AVC uses two-tap bilinear filtering. By using larger tap filters, HEVC interpolation accuracy is improved. In contrast to the two-stage interpolation process used by H.264/AVC standard, HEVC interpolation uses a simpler architecture based on a single and separable process to interpolate

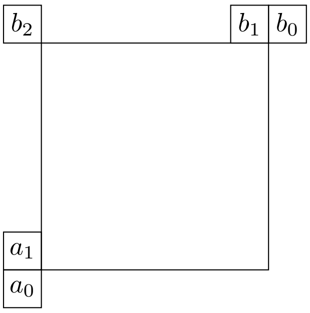


Figure 2.6: Spatial candidates of motion information for Merge mode. Figure taken from [2].

all fractional positions, without requiring any intermediate rounding operations. Detailed information about HEVC interpolation filter can be found in [1, 2].

Merge mode

The Merge mode was introduced in HEVC standard, as a new technique to derive motion information from spatial and temporal neighbouring blocks. This technique extends the concepts of the direct and skip modes used in H.264/AVC, using a more sophisticated algorithm. It provides an alternative solution for motion representation, which selects the motion vector from an adaptive set of candidate vectors and signals it using an index number, that can be more efficiently represented.

A set of five spatial motion vector candidates, with the positions illustrated in Figure 2.6, is considered in Merge mode. Unavailable candidates are not accounted and duplicated candidates (*i.e.* with the same motion information) are removed. According to the order a_1, b_1, b_0, a_0, b_2, after the proper exclusions, the $C-1$ spatial candidates are retained, where C is the maximum number of Merge candidates defined in the slide header. One temporal candidate is also added to fulfil the C Merge candidates. This candidate is obtained from the right bottom pixel outside the co-located block of one reference picture, whose index should be transmitted. If no candidate is available in right bottom position, the middle position of the block is used instead. HEVC Merge mode also considers some mechanisms to derive additional candidates, when the number of available spatial and temporal candidates is inferior to C.

Motion vector prediction

In order to encode motion vectors, in those cases when the Merge mode is not chosen, HEVC uses a differential encoding approach for more efficient representation. This predictive approach for motion vector coding works similarly to a lossless DPCM system, where no loss is admitted. Its advantage is the possibility to encode a residual signal (difference motion vectors) that presents a concentrated probability density function around zero, providing more efficient entropy coding results.

HEVC considers the same spatial candidate pixels of the Merge mode, illustrated in Figure 2.6, to derive the motion vector predictors from which the difference vector is obtained. Among these candidates, only two are chosen to be used as possible predictors. HEVC limits the number of predictors to avoid further increasing of the computational complexity associated to the motion estimation process. The first motion candidate is chosen by picking up the first available candidate in the set of left candidates arranged as a_0, a_1, while the second one is the first available candidate among the set of above candidates arranged as b_0, b_1, b_2.

When the number of available spatial motion vector predictors is less than two, HEVC includes the temporal candidate as predictor. This is one of the mechanisms used by HEVC to guarantee the existence of two motion vector predictors. For the signalling, only a single binary flag is required to indicate which motion vector predictor should be used.

2.4 Least-squares prediction methods

Least-squares methods have been used for linear prediction in many signal processing applications. The modelling and compression of audio and speech signals are successful applications of linear prediction for one-dimensional signals. Currently, linear prediction is used in most speech coding algorithms.

In regard to image signals, several successful applications of linear prediction for lossless image coding using Least-Squares Prediction (LSP) methods have been presented in literature [21, 25–28]. Due to the abrupt changes in image's local statistics, adaptive approaches of linear prediction, such as the context-based adaptive least-squares prediction [21], have shown to perform better than the non-adaptive methods. For the case of lossy image and video compression, there are some research works that demonstrate the advantage of the context-based adaptive prediction for the recent state-of-the-art image and video coding algorithms [9, 29–32].

In this section, some methods based on LSP for efficient image and video compression are described. Initially, the advantage of LSP for edge prediction in both spatial and temporal directions is explained. Then, the LSP algorithm proposed for lossless image coding is described. Finally, this section presents some variants of LSP for block-based prediction in lossy image coding applications.

2.4.1 Linear prediction of images and video using LSP

The principle of LSP is to linearly combine a set of causal neighbouring pixels using coefficients previously estimated in a local causal training window. A common implementation of LSP estimates the filter coefficients in a pixel-by-pixel basis, *i.e.* a new set of coefficients is estimated for each pixel to be predicted. Such procedure allows LSP to adaptively embed the changing local texture characteristics in the linear prediction coefficients. Since the coefficients are estimated in a causal training window, LSP adaptation is implicit and does not require the transmission of the coefficient values.

In [21], it has been demonstrated that context-based adaptive LSP provides an effective modelling of the edges present in natural images. This fact led to a new interpretation of LSP, which is referred to as edge-directed property. The reasonable match of the linear predictor to the edge direction is justified by the higher influence of the pixels around the edges in the least-squares optimisation process.

The problem of edge modelling is of particular interest, due to the large amount of information carried out by edges present in natural images. Typically, smooth regions are easily predicted and compressed. However, the same does not apply to edge areas or complex textured areas. The current state-of-the-art standards use the directional intra prediction to explicitly model image edges, based on a predefined number of fixed directions. Due to the edge-directed property, LSP is able to provide a reasonable prediction of arbitrarily oriented edges using an implicit methodology. The performance of LSP has been evaluated in [21], based on the Edge-Directed Prediction algorithm for lossless image coding. The experiments have shown a superior performance than other state-of-the-art lossless image coding standards.

Modified approaches of LSP algorithm have been also investigated in literature for lossy image coding. The main challenge of these methods is to adapt LSP for block-based image coding algorithms, because pixel-based coding approaches are not efficient for lossy compression. Some of these LSP-based methods have been implemented and evaluated using the state-of-the-art transform-based H.264/AVC standard [29], as well as alternative image coding algorithms based on pattern matching [9].

Regarding video coding applications, proposals of LSP have been investigated for implicit motion compensation. An interesting solution is to use a spatio-temporal prediction approach, in which temporal samples are included in the LSP filter context in order to implicitly learn the motion information. This kind of LSP-based motion compensation approach has been proposed in [31] as an alternative to the explicit block-matching algorithm, widely used for MCP in current video coding

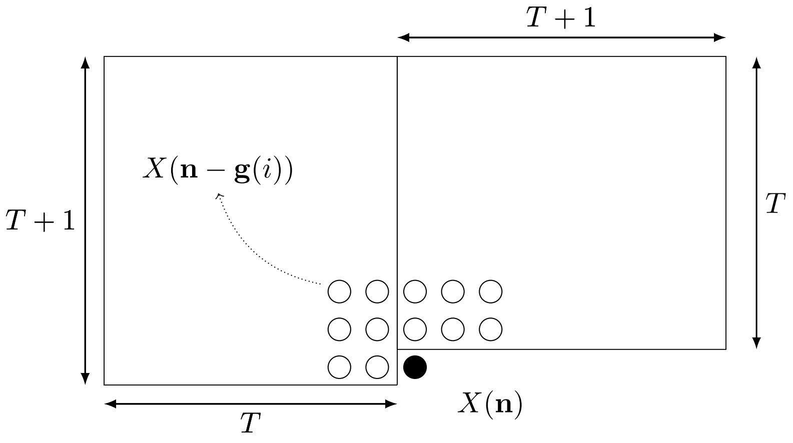


Figure 2.7: LSP filter context (pixels represented by white circles) and associated training window.

standards.

2.4.2 Context-based adaptive LSP

In this section, the context-based adaptive LSP algorithm is described as proposed in [21] for lossless intra image coding. The main challenge of LSP is to develop an efficient predictive model, which fully exploits the information contained in its context.

Let $X(\mathbf{n})$ denote the image pixel to be linearly predicted, where $\mathbf{n}$ is a two-dimensional vector with the spatial coordinates in the image. By using the N nearest spatial causal neighbours, according to a Nth order Markovian model, the predicted pixel is computed as:

$$\hat{X}(\mathbf{n}) = \sum_{i=1}^{N} a_i X(\mathbf{n} - \mathbf{g}(i)), \tag{2.1}$$

where $\mathbf{g}(i)$ gives the relative position of each pixel in the filter context (or support), and a_i are the filter coefficients. An example of the filter context, as proposed in [21], is illustrated in Figure 2.7 for $N = 12$.

In order to avoid the transmission of side information, LSP locally estimates the prediction coefficients using a causal training window, which is available in both the encoder and the decoder. This approach provides an implicit way to adaptively learn the orientation of the edges contained in the training window. In [21], a rectangular training window that contains $M = 2T(T+1)$ elements is proposed, as show in Figure 2.7. The training window pixels can be arranged in a column vector $\mathbf{y} = [X(\mathbf{n} - \mathbf{h}[1]) \ldots X(\mathbf{n} - \mathbf{h}(M))]^T$, where $\mathbf{h}(j)$ represents the relative position of each pixel. With the filter context pixels, the following $M \times N$ matrix can be formed:

$$\mathbf{C} = \begin{bmatrix} X(\mathbf{n}-\mathbf{h}(1)-\mathbf{g}(1)) & \dots & X(\mathbf{n}-\mathbf{h}(1)-\mathbf{g}(N)) \\ \vdots & & \vdots \\ X(\mathbf{n}-\mathbf{h}(M)-\mathbf{g}(1)) & \dots & X(\mathbf{n}-\mathbf{h}(M)-\mathbf{g}(N)) \end{bmatrix}.$$

Note that $X(\mathbf{n}-\mathbf{h}(j)-\mathbf{g}(i))$ represents the i^{th} filter context sample associated to the training window pixel, given by $X(\mathbf{n}-\mathbf{h}(j))$. The filter coefficients, $\mathbf{a} = [a_1 \dots a_N]^T$, can be determined by least-squares optimisation, finding the solution for $\min_{\mathbf{a}}(\|\mathbf{y}-\mathbf{Ca}\|_2^2)$. A well-known closed-form solution for LS problem is given by:

$$\mathbf{a} = (\mathbf{C^TC})^{-1}(\mathbf{C^Ty}). \tag{2.2}$$

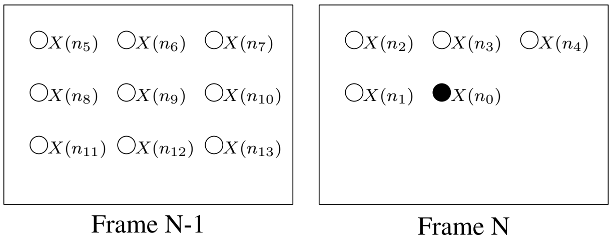
As suggested by the edge-directed property, the LSP estimated predictor coefficients tend to adapt to the orientation of the edges contained in the training window. Because of such adaptation, LSP is able to provide a reasonable prediction of $\hat{X}(\mathbf{n})$ along the estimated edge's orientation.

2.4.3 Block-based LSP

Block-based implementations of LSP have been proposed specifically for lossy image encoders. In [9], an LSP mode based on the previously described algorithm was presented for block-based intra prediction in the MMP algorithm. In it, LSP estimates the weighting coefficients on a pixel-by-pixel basis, using a slightly different training window for each pixel of the block to predict. The training and linear prediction procedures of such lossy-based LSP use not only the reconstructed pixels from previously encoded blocks, but also the available predicted pixels from the current block.

The recursive use of predicted samples in LSP algorithm has some drawbacks, because predicted samples inherently incorporate an error not present in the reconstructed samples. This yields a sort of error propagation. In order to reduce such error propagation, the method in [30] presents a line-based linear prediction scheme for the H.264/AVC standard. Unlike the pixel-based training procedure of [9], the line-based model is fixed for the whole line of the block, being updated after each encoded line.

An alternative LSP method for block-based intra prediction in the H.264/AVC standard has been proposed in [29]. In it, an adaptive training window and filter context are used, depending on the number of available neighbouring blocks. Similarly to the LSP proposal in [9], this approach uses the predicted pixels from the current block in addition to the reconstructed ones of the previously encoded blocks. However, the proposal in [29] performs the training procedure on a block-by-block basis. By performing a single training procedure for each block, the same set of esti-



○$X(n_5)$ ○$X(n_6)$ ○$X(n_7)$
○$X(n_8)$ ○$X(n_9)$ ○$X(n_{10})$
○$X(n_{11})$ ○$X(n_{12})$ ○$X(n_{13})$
Frame N-1
○$X(n_2)$ ○$X(n_3)$ ○$X(n_4)$
○$X(n_1)$ ●$X(n_0)$
Frame N

Figure 2.8: Example of a 13th order spatio-temporal filter context using causal neighbours from the current frame (spatial neighbours) and previous encoded frame (temporal neighbours).

mated linear coefficients is used to predict the entire block. Furthermore, due to the reduced amount of training procedures, this solution presents a lower computational complexity.

Some applications of LSP for linear prediction between colour image components have been also presented using a single least-squares training procedure in the neighbourhood of the block [10, 33, 34]. The work of [33] exploits the fact that the same filter coefficients are used to predict the whole block and investigates an explicit approach that quantises and transmits the linear coefficients.

An LSP prediction mode for inter prediction between stereo pair views has been also proposed in [12]. One interesting feature of this proposal is the usage of multiple filter contexts of different orders and shapes. At the encoder side, eight possible contexts are tested and the one that produces the minimum block prediction error is signalled to the decoder, as side information. A combination of spatial and inter view prediction is used, by including pixels from both the left and right views in the filter context.

2.4.4 Spatio-temporal LSP

The idea of spatio-temporal prediction has been investigated for implicit motion compensation in video coding applications. This interesting approach exploits the LSP edge-directed property for prediction of the motion trajectory along the temporal axis, based on the duality between 2D image contour edges and 3D video motion trajectory [31, 32]. Such duality is given by the similarity between image edges and flow-like patterns that are observed in transversal sectioning of successive video frames along the time direction.

In order to learn the edge information associated to the motion flow pattern along temporal direction, an appropriate filter context that includes causal samples from previous temporal frames is used. Such context configuration allows LSP to implicitly embed motion trajectory in the linear predictor coefficients, based on the training procedure in a causal window. Figure 2.8 illustrates an example of the referred spatio-temporal context for the LSP algorithm. Spatio-temporal context provides some flexibility to exploit motion information from the temporal portion of the context or to employ spatial prediction by assigning more weights to the spatial neighbours.

The filter context of LSP should cover the flow-like patterns observed in 3D video along temporal axis, in order to exploit the existing motion trajectories. Typically, filter contexts as the one illustrated in Figure 2.8 tend to be more effective to predict slow motion, in a variety of types, such as panning, rotations, zoom, and jittering. In order to estimate LSP coefficients, a cube shaped training area that contains the local causal pixels, considered stationary for natural slow motion video, is suggested in [31].

The prediction efficiency of LSP can be further improved by means of an adaptive update of the filter context. In [31], it is proposed to change LSP context configuration according to the motion characteristics. The contexts are designed and chosen based on the statistics of the causal data, not requiring the transmission of auxiliary information. By using different contexts for zooming, jittering or panning, LSP can better learn local motion characteristics and predict unknown samples.

When the video content is characterised by fast motion, the prediction performance of LSP tends to reduce. The inconsistency between the training data and the actual motion trajectory is the main reason for the performance degradation. A solution proposed in [31] for a fast camera panning uses temporal frame warping to compensate it. To minimise the effect of other types of unpredictable events, which reduce the stationary statistics of the signal, an adaptive training window size is also proposed. The main idea is to change the number of temporal frames included in the training window, being updated on a frame-by-frame basis and explicitly transmitted.

2.5 Sparse representation for image prediction

Another important class of algorithms, which has been increasingly considered in literature for intra image coding, is based on sparse representation [35–37]. These approaches are motivated by the assumption that natural image signals are formed by few structural primitives or representative features.Sparse prediction tries to approximate the input signal using a linear combination of a small number of these primitives, selected from a large and redundant basis, known as dictionary. Typically, these methods provide efficient prediction of highly textured areas, where the traditional directional prediction techniques present some issues.

In the context of sparse representation, dimensionality reduction methods have been also recently proposed for image prediction [38, 39]. The main idea of di-

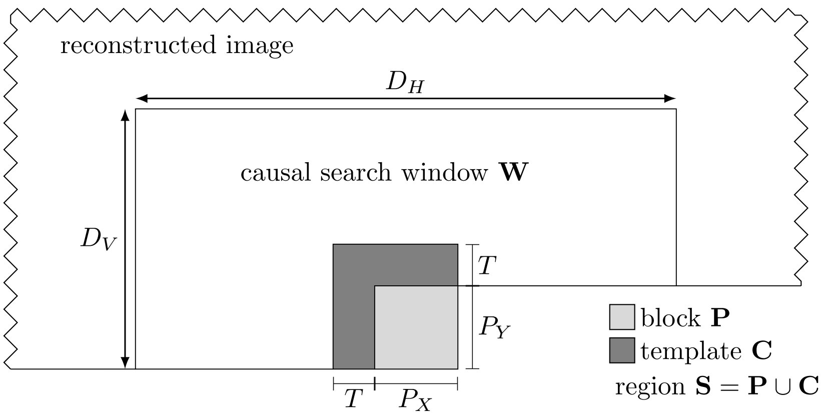


Figure 2.9: Approximation support (template) and search window used in sparse coding and neighbour-embedding methods.

mensionality reduction techniques is to approximate the input unknown block as a linear combination of the k nearest neighbours (k-NN) in terms of the Euclidean distance, determined from an adaptive local dictionary. Such dictionary is formed by elements derived from texture patches present in a causal reconstructed area in the neighbourhood of the unknown block. This kind of prediction can be viewed as a k-sparse representation of the block. In order to avoid the transmission of the nearest neighbours, an approximation to a causal region defined in the neighbourhood of the block (also called template region) is firstly determined. The optimal coefficients estimated for the template are then used to predict the unknown block, by combining the block patches adjacent to the k-NN templates. Since the coefficients for the template can be implicitly determined in the decoder side, no information about the k-NN needs to be transmitted.

In this section, prediction algorithms for image and video coding based on sparse representation are presented. These techniques include Matching Pursuit-based algorithms (MP) and the Template Matching (TM) algorithm. Furthermore, dimensionality reduction methods based on Non-negative Matrix Factorisation (NMF) and Locally Linear Embedding (LLE) methods are described.

2.5.1 Sparse prediction problem formulation

Consider the N-pixel region $\mathbf{S}$, the union of the N_p-pixel block $\mathbf{P}$ to be predicted, and the N_c-pixel approximation support (or template) $\mathbf{C}$, as illustrated in Figure 2.9. Note that the pixels from the block $\mathbf{P}$ to be predicted are unknown, and the ones of the template $\mathbf{C}$ are the previously reconstructed pixels. By using an appropriate dictionary, the sparse prediction method estimates the best linear approximation for the known template $\mathbf{C}$ and keeps the same model to approximate the corresponding unknown block $\mathbf{P}$.

Let vector $\mathbf{b}$ be composed by the N pixel values of region $\mathbf{S}$, stacked in a column (assuming zero value for unknown pixels of block $\mathbf{P}$). Also, let an $N \times M$ matrix $\mathbf{A}$ represent the dictionary matrix, a horizontal concatenation of M basis functions represented as column vectors, $\mathbf{a}_m$, with $m = 0, \ldots, M-1$. An over-complete dictionary is used, *i.e.* the number of atoms (or basis functions) M is greater than the size N of each function.

The dictionary $\mathbf{A}$ can be built in an adaptive way or using fixed basis functions. In [35], fixed dictionaries based on Discrete Cosine Transform (DCT) or Discrete Fourier Transform (DFT) waveforms are used. Adaptive dictionaries are frequently formed by atoms derived from patches present in a local window in the reconstructed causal neighbourhood [36]. However, dictionary learning methods have also been used for sparse prediction [40]. In this description, adaptive dictionaries formed based on the reconstructed causal texture patches are considered. These dictionaries are built by stacking all the patches similar to region $\mathbf{S}$, which exist in a predefined causal search window $\mathbf{W}$, as shown in Figure 2.9.

The dictionary matrix $\mathbf{A}$ (and vector $\mathbf{b}$), can be separated into two vertically concatenated sub-matrices $\mathbf{A_c}$ and $\mathbf{A_p}$ (and two vectors $\mathbf{b}_c$ and $\mathbf{b}_p$), corresponding to the pixels in the spatial location of template $\mathbf{C}$ and predicting block $\mathbf{P}$, respectively. Sparse representation algorithms aim at approximating the template $\mathbf{C}$ (vector $\mathbf{b}_c$), by solving the following optimisation problem:

$$\min_{\mathbf{x}} \|\mathbf{b}_c - \mathbf{A}_c\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq t, \tag{2.3}$$

where $\|\mathbf{x}\|_0$ denotes the L_0 norm of $\mathbf{x}$, *i.e.* the number of non-zero components in $\mathbf{x}$, and t is a bound on sparsity.

Since searching for the sparsest solution for this problem is NP-hard, approximated solutions are usually estimated. The Matching Pursuit (MP) [41] and Orthogonal Matching Pursuit (OMP) [42] algorithms are iterative greedy methods commonly used to find approximate solutions with tractable computational complexity. Once the optimal coefficients are computed, sparse prediction estimates the predicted signal through $\hat{\mathbf{b}}_p = \mathbf{A}_p\mathbf{x}_{opt}$, where $\mathbf{x}_{opt}$ is the sparsest solution of (2.3).

The advantage of the presented approach, based on a template area, is to provide an implicit way to derive the solution of the sparse prediction problem. By solving the problem expressed in (2.3) exclusively based on the causal reconstructed neighbourhood, specifically through $\mathbf{b}_c$ and $\mathbf{A}_c$ data structures, the decoder can implicitly derive the same sparsest solution $\mathbf{x}_{opt}$. This approach avoids the bitrate cost associated to the explicit transmission of the sparse solution. However, since the solution $\mathbf{x}_{opt}$ was optimised for sparse representation of the template $\mathbf{C}$, its prediction performance for the block $\mathbf{P}$ may be reduced. This is so because the solution for the template $\mathbf{C}$ may significantly differ from the optimal sparse solution for the block $\mathbf{P}$. To obtain a minimal difference to the optimal solution, the assumption of stationary statistics among the template $\mathbf{C}$ and block $\mathbf{P}$ should apply.

2.5.2 Matching Pursuit methods

The MP algorithm can be used to compute sparse signal representations by iteratively selecting the dictionary atoms. This method provides a possible approximate solution to the problem in (2.3), which can be rewritten as:

$$\min \left\{ \|\mathbf{x}\|_0 \ : \ \|\mathbf{b}_c - \mathbf{A}_c\mathbf{x}\|_2^2 \leq \rho \right\} , \tag{2.4}$$

with some admissible approximation error $\rho \geq 0$.

MP procedure iteratively generates a sequence of possible solutions $\mathbf{x}_k$, which present an increasing number of non zero components. In first iteration, $\mathbf{x}_0 = \mathbf{0}$ and the initial residual vector is given by $\mathbf{r}_0 = \mathbf{b}_c - \mathbf{A}_c\mathbf{x}_0 = \mathbf{b}_c$. At iteration k, MP selects the basis function, $\mathbf{a}_{c_{m_k}}$ (with m_k referring to the column index number), which has the highest correlation with the residual $\mathbf{r}_{k-1} = \mathbf{b}_c - \mathbf{A}_c\mathbf{x}_{k-1}$. The optimal dictionary atom $\mathbf{a}_{c_{m_k}}$ can be found by

$$m_k = \arg\max_m \frac{|\mathbf{a}_{c_m}^T \mathbf{r}_{k-1}|}{\|\mathbf{a}_{c_m}\|_2}. \tag{2.5}$$

The residual vector at iteration k is then computed as

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \frac{\mathbf{a}_{c_{m_k}}^T \mathbf{r}_{k-1}}{\mathbf{a}_{c_{m_k}}^T \mathbf{a}_{c_{m_k}}} \mathbf{a}_{c_{m_k}} = \mathbf{r}_{k-1} - x_{m_k}\mathbf{a}_{c_{m_k}} \tag{2.6}$$

where x_{m_k} is the weight of the new atom, which is added to $\mathbf{x}_{k-1}$ in order to generate the new sparse solution vector $\mathbf{x}_k$. MP proceeds with this algorithm until $\|\mathbf{b}_c - \mathbf{A}_c\mathbf{x}_k\|^2 \leq \rho$ is satisfied.

As previously stated, the optimal solution estimated for the template area $\mathbf{C}$ may not be the best one for the block $\mathbf{P}$. For a more efficient prediction of block $\mathbf{P}$, some methods propose to explicitly select the solution that provides the best prediction results for block $\mathbf{P}$, from a set of possible solutions. For example, in [35] all the intermediate solutions $\mathbf{x}_k$, for $k = 1, \ldots, k_{max}$, where k_{max} is the value of k that satisfied the maximum error ρ, are saved and tested to predict the block $\mathbf{P}$. The optimum solution $\mathbf{x}_{opt}$ that minimises the block prediction error, given by $\|\mathbf{b}_p - \mathbf{A}_p\mathbf{x}_k\|_2^2$, is selected.

In order to reproduce the prediction process in the decoder side, the iteration k^* associated to the optimum solution $\mathbf{x}_{opt}$ should be explicitly transmitted. The prediction block is calculated as $\hat{\mathbf{b}}_p = \mathbf{A}_p\mathbf{x}_{opt}$. Algorithm 2.1 summarises the described

Algorithm 2.1: Matching Pursuit algorithm for sparse image prediction.

```
Input: $\mathbf{A}_c$ , $\mathbf{b}_c$ , $\mathbf{A}_p$ , $\mathbf{b}_p$ , $\rho$
Output: $\hat{\mathbf{b}}_p$ , $k^*$
initialisation: $k = 0$ , $\mathbf{x}_0 = \mathbf{0}$ , $\mathbf{r}_0 = \mathbf{b}_c - \mathbf{A}_c \mathbf{x}_0 = \mathbf{b}_c$
repeat
    $k = k + 1$
    $m_k = \arg\max_m \frac{|\mathbf{a}_{cm}^T \mathbf{r}_{k-1}|}{\|\mathbf{a}_{cm}\|_2}$
    $x_{m_k} = \frac{\mathbf{a}_{cm_k}^T \mathbf{r}_{k-1}}{\mathbf{a}_{cm_k}^T \mathbf{a}_{cm_k}}$
    $\mathbf{r}_k = \mathbf{r}_{k-1} - x_{m_k} \mathbf{a}_{cm_k}$
    $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{u}_k$ , where $\mathbf{u}_k$ is 1-sparse vector with $x_{m_k}$ on position $m_k$ and
    remaining entries null
until $\|\mathbf{b}_c - \mathbf{A}_c \mathbf{x}_k\|^2 \leq \rho$
$k_{max} = k$
$k^* = \arg \min_{k \in [1, k_{max}]} \|\mathbf{b}_p - \mathbf{A}_p \mathbf{x}_k\|_2^2$
$\mathbf{x}_{opt} = \mathbf{x}_{k^*}$
$\hat{\mathbf{b}}_p = \mathbf{A}_p \mathbf{x}_{opt}$
```

sparse image prediction procedure using MP algorithm.

Orthogonal Matching Pursuit

OMP is a popular extension to the MP algorithm, which is able to provide better results, at the cost of an additional computational complexity [42]. The main difference of OMP relative to MP is that all non-zero coefficients chosen so far are updated at each iteration. In the context of the presented sparse prediction problem, this consists of an orthogonal projection of the template signal, $\mathbf{b}_c$, onto the subspace spanned by the dictionary atoms selected so far.

A sparse image prediction method based on OMP is summarised in Algorithm 2.2. Note that $\mathbf{A}_c^k$ refers to a compacted matrix, which contains a sub-set of the atoms of $\mathbf{A}_c$, specifically all the atoms selected until iteration k. As for stopping condition, alternative approaches that limit the maximum number of iterations can be used.

2.5.3 Template Matching algorithm

The TM algorithm [43] can be interpreted as a particular case of the presented sparse prediction method using MP, in which only one iteration is performed and the weighting coefficient is equal to 1. In practice, TM algorithm can be formulated as a search procedure which compares the reference template with all equally shaped candidate templates existing in a causal search window (the atoms of the dictionary). The prediction of the unknown block is given by the reconstructed block associated

Algorithm 2.2: Orthogonal Matching Pursuit algorithm for sparse image prediction.

```
Input: $\mathbf{A}_c$ , $\mathbf{b}_c$ , $\mathbf{A}_p$ , $\mathbf{b}_p$ , $\rho$
Output: $\hat{\mathbf{b}}_p$ , $k^*$
initialisation: $k = 0$ , $\mathbf{x}_0 = \mathbf{0}$ , $\mathbf{r}_0 = \mathbf{b}_c$ , $\mathbf{A}_c^0 = [\,]$ , $\mathbf{A}_p^0 = [\,]$
repeat
    $k = k + 1$
    $m_k = \arg\max_m \frac{|\mathbf{a}_{c_m}^T \mathbf{r}_{k-1}|}{\|\mathbf{a}_{c_m}\|_2}$
    $\mathbf{A}_c^k = \mathbf{A}_c^{k-1} \cup \{\mathbf{a}_{c_{m_k}}\}$ and $\mathbf{A}_p^k = \mathbf{A}_p^{k-1} \cup \{\mathbf{a}_{p_{m_k}}\}$
    $\mathbf{x}_k = \left(\mathbf{A}_c^{k^T} \mathbf{A}_c^k\right)^{-1} \mathbf{A}_c^{k^T} \mathbf{b}_c = \mathbf{A}_c^{k^+} \mathbf{b}_c$
    $\mathbf{r}_k = \mathbf{b}_c - \mathbf{A}_c^k \mathbf{x}_k$
until $\|\mathbf{b}_c - \mathbf{A}_c \mathbf{x}_k\|^2 \leq \rho$
$k_{max} = k$
$k^* = \arg \min_{k \in [1, k_{max}]} \|\mathbf{b}_p - \mathbf{A}_p \mathbf{x}_k\|_2^2$
$\mathbf{x}_{opt} = \mathbf{x}_{k^*}$
$\hat{\mathbf{b}}_p = \mathbf{A}_p \mathbf{x}_{opt}$
```

(or adjacent) to the candidate template, which resulted in the lowest matching error. Typically, the reconstructed pixels belonging to the neighbourhood of the block to be predicted are used as reference template, equivalently to the area **C**, of Figure 2.9.

Improved variations of TM algorithm have been proposed and implemented in H.264/AVC standard, *e.g.* TM based on the averaging of multiple predictors [44] and TM using adaptive illumination compensation methods [45]. The use of TM algorithm in addition to the traditional intra prediction modes has shown to be advantageous to predict textured areas. Applications of BMA for intra image prediction have also shown to provide effective prediction results [46]. The principle of both methods is to reuse repeated patterns along the image. However, BMA requires some kind of signalling to indicate the optimal matched block in the causal reconstructed area, while TM provides an implicit way to find the matching block in the decoder. Hybrid approaches combining both BMA and TM for intra prediction have also been investigated in the past [47].

2.5.4 Neighbour embedding methods

Neighbour embedding methods form a subset of data dimensionality reduction methods, which have been successfully applied for intra image prediction. The idea of these methods is to search for the NNs (in terms of the Euclidean distance) of the input data point and then to compute the linear combination of the NNs that best approximate the input data point.

Data dimensionality reduction methods rely on the assumption that the real world is sampled from a non-linear low dimensional manifold, which is embedded in the high dimensional space. When applied to image prediction, these methods learn the neighbour embedding of input data, but they do not proceed to the computation of the low dimensional space. Typical proposals find the linear combination of the NN patches which best approximates some known data, *e.g.* the causal template in the block neighbourhood, and then extrapolate that linear relationship to the NNs of the unknown block to be predicted. This kind of operation makes neighbour embedding methods very similar to sparse prediction techniques based on MP methods. Actually, neighbour embedding methods can be interpreted as sparse image representations in which the selected dictionary atoms (*i.e.* the elements that participate in sparse solution) are given by k-NN patches. The investigation of two neighbour embedding methods for image prediction was performed in [38, 39], namely the Non-negative Matrix Factorisation (NMF) and Locally Linear Embedding (LLE) techniques. The main difference between these methods is related to the method used to compute the linear coefficients associated to the NN patches.

The NMF method consists in a low rank approximation of the input data, given by the product of two matrices of lower dimension whose elements are non-negative. One of these matrices typically contains basis functions (in its columns) that provide a good linear approximation of the input data. The non-negative constraint only allows additive linear combinations, which in some applications can be useful for physical interpretation of the results. In the case of image data, which is made of non-negative values, this constraint ensures a non-negative prediction result. Similarly to the sparse prediction problem given by equation (2.3), the NMF image prediction method is formulated as a constrained least-squares problem, which imposes non-negative coefficients. A sparsity constraint is also imposed, by using this method over a compacted matrix formed by k-NN patches. A detailed description of this algorithm can be found in [38].

The LLE has been designed to learn the underlying non-linear manifolds embedded in the high dimensional data. The method exploits the local linear characteristics of the high dimensional data, in order to generate a lower dimensional representation which preserves those characteristics. Applications of LLE to intra image prediction have shown better results than NMF method [38]. Given its importance for this thesis, the description of the LLE algorithm for image prediction as proposed in [38] is presented in what follows.

Image prediction based on LLE

The LLE method first searches a representation of the template $\mathbf{C}$ (refer to Figure 2.9) using a linear combination of the k-NN patches defined in the matrix $\mathbf{A}_c$, built from the causal search window $\mathbf{W}$, as previously defined in Subsection 2.5.1. Then, the same estimated coefficients are used to predict the block $\mathbf{P}$, by linearly combining the corresponding k-NN patches defined in matrix $\mathbf{A}_b$.

The k-NN method works like a sparsity constraint by choosing the k closest patches (represented by the columns of $\mathbf{A}_c$) to the template $\mathbf{C}$ (vector $\mathbf{b}_c$), in terms of Euclidean distance. The sub-matrix of $\mathbf{A}_c$ containing the selected k-NN patches is denoted by $\mathbf{A}_c^k$. In order to compute the linear weights, LLE problem solves a least-squares problem, which imposes a sum-to-one constraint on the linear weights. This constraint is part of LLE algorithm forcing the approximation of each data point to lie in the linear subspace spanned by its nearest neighbours. The enunciated problem can be written as

$$\min_{\mathbf{x}_k} \|\mathbf{b}_c - \mathbf{A}_c^k \mathbf{x}_k\|_2^2 \quad \text{subject to} \quad \sum_m \mathbf{x}_{k_m} = 1. \tag{2.7}$$

where $\mathbf{x}_k$ is the solution vector containing the k optimal linear coefficients. Note that, although sparsity constraint is not explicit in problem formulation, the selection of a limited number of columns of $\mathbf{A}_c$ by k-NN method imposes the sparse representation.

The solution to this problem can be obtained based on a covariance matrix, $\mathbf{D}_k$, computed for the k-NN templates of $\mathbf{A}_c^k$ in reference to the template $\mathbf{b}_c$, by solving the linear system $\mathbf{D}_k \mathbf{x}_k = \mathbf{1}$ (where $\mathbf{1}$ is the column vector of ones), and then rescaling the weights so that they sum to one. The predicted block is then computed using $\hat{\mathbf{b}}_p = \mathbf{A}_p \mathbf{x}_k$.

For an improved performance, the proposal of [38] suggests to vary the sparsity constraint, by testing several k-NNs, *e.g.* with $k = 1, \ldots, K$. The number of k-NNs can be either constant for the whole encoding process or explicitly signalled to the decoder. In the second case, the optimal number of NNs is selected by testing the block $\mathbf{P}$ prediction accuracy for each k value. The Algorithm 2.3 describes the presented LLE method for intra image prediction, assuming the explicit selection and signalling of the optimal sparsity value k^*.

Experiments performed in [38] using H.264/AVC standard demonstrate the advantage of the described LLE-based prediction method. The reported results show that LLE provides superior prediction quality and compression efficiency than directional prediction, TM algorithm, as well as other sparse prediction algorithms based on MP method.

In a more recent research work, modified variants of the neighbour embedding algorithms, denominated as correspondence map-aided neighbour embedding methods, were proposed [39]. The idea of these methods is to enable alternative procedures to select the k-NNs, which make use of auxiliary information transmitted to the decoder. These methods solve some limitations of implicit NNs selection, namely when the template and the block to be predicted are not correlated.

Algorithm 2.3: LLE-based algorithm for intra image prediction.

```
Input: $\mathbf{A}_c$ , $\mathbf{b}_c$ , $\mathbf{A}_p$ , $\mathbf{b}_p$ , $K$
Output: $\hat{\mathbf{b}}_p$ , $k^*$
initialisation: $k = 0$ , $\mathbf{A}_c^0 = [\,]$ , $\mathbf{A}_p^0 = [\,]$ , $\mathbf{A}_c'^0 = \mathbf{A}_c$
repeat
    $k = k + 1$
    $m_k = \arg\min_{m|a_{c_m} \in \mathbf{A}_c'^{k-1}} \{d_m\}$  where $d_m = \|\mathbf{b}_c - \mathbf{a}_{c_m}\|_2^2$
    $\mathbf{A}_c^k = \mathbf{A}_c^{k-1} \cup \{\mathbf{a}_{c_{m_k}}\}$ and $\mathbf{A}_p^k = \mathbf{A}_p^{k-1} \cup \{\mathbf{a}_{p_{m_k}}\}$
    $\mathbf{A}_c'^k \leftarrow \mathbf{A}_c \backslash \{\mathbf{A}_c^k\}$
    calculate covariance matrix $\mathbf{D}_k$ of $\mathbf{A}_c^k$ in reference to $\mathbf{b}_c$
    solve $\mathbf{D}_k \mathbf{x}_k = \mathbf{1}$ for $\mathbf{x}_k$
    $\mathbf{x}_k = \frac{\mathbf{x}_k}{\mathbf{x}_k^T \mathbf{1}}$
until $k = K$
$k^* = \arg \min_{k \in [1,K]} \|\mathbf{b}_p - \mathbf{A}_p^k \mathbf{x}_k\|_2^2$
$\mathbf{x}_{opt} = \mathbf{x}_{k^*}$
$\hat{\mathbf{b}}_p = \mathbf{A}_p^{k^*} \mathbf{x}_{opt}$
```

2.6 Conclusions

In this chapter we presented an overview of several prediction techniques proposed for image and video compression. Predictive techniques are of great importance in the design of current image and video coding algorithms, being the main focus of the investigation described in this thesis.

Section 2.2 presented the basis of predictive image coding, describing the legacy DPCM encoder and its relations with the current lossless and lossy image and video encoders. In Section 2.3, the state-of-the-art on image and video prediction, namely the directional intra prediction and motion-compensated prediction methods were described, as proposed in the recent HEVC standard. In this thesis, the directional intra prediction has been investigated to improve the MMP algorithm in Chapter 4 and for efficient depth map coding as proposed in Chapter 5.

Two important families of prediction techniques were also presented in this chapter, namely the linear prediction methods, based on least-squares optimisation described in Section 2.4, and the sparse representation-based prediction methods described in Section 2.5. Due to the importance of linear prediction and sparse representation, alternative techniques based on these principles have been investigated in this thesis for the compression of generic images. For instance, an extended version of the LLE-based prediction method, described in Subsection 2.5.4, is proposed in Chapter 6 for the HEVC encoder.

In the following chapter, we describe the main image and video coding algorithms used along this thesis, where the proposed prediction techniques were implemented and evaluated.

Chapter 3

Image and video coding algorithms

In this chapter, we present an overview on the main image and video coding algorithms used in this thesis, for both 2D and 3D video signals. These algorithms have been utilised as reference benchmarks for the experimental evaluation of the proposed prediction methods. Some of them have also been used as basis compression frameworks, where the proposed prediction methods were implemented and evaluated.

In Section 3.1 we introduce some basic concepts about video compression, in particular the hybrid architecture that characterises the coding algorithms described in this chapter. Section 3.2 describes the state-of-the-art algorithms for lossy compression of 2D video signals, which includes the recent transform-based High Efficiency Video Coding (HEVC) standard and the pattern matching-based Multidimensional Multiscale Parser (MMP) algorithm. Although HEVC was primarily designed for video compression, its performance for intra-frame coding has made it an unequivocal approach for still image coding, presenting state-of-the-art results. Thus, an experimental study of HEVC and MMP algorithms for still image and video coding is presented. Section 3.3 addresses the 3D video compression issues. First, an overview is given on the main 3D technologies, namely stereo, multiview, depth enhanced and holoscopic video formats. Then, the current algorithms and standards for 3D video compression are described. Some experimental results illustrating the 3D video standards coding performance are also presented.

3.1 Hybrid video compression

The ever growing consumption of multimedia contents, specifically still images and monoscopic video, associated to the increasing diversity of services and applications motivated the development of more efficient video coding standards able to cope with the increasing compression requirements. The main goal of the lossy image and video coding algorithms is to reduce the amount of information required to represent these signals, while minimising the loss of subjective quality.

The adopted hybrid video coding model has shown to be an effective solution, as it has been used in all video coding standards since the early days of digital video compression. As shown in previous chapter, the hybrid architecture is based on three main components: the prediction stage, the transform-based coding and entropy coding. The main difference among each generation of video coding standards is the design of the coding techniques that constitute each component of the hybrid architecture, which results in different compression efficiency and computational complexity.

The prediction stage typically comprises both intra and inter prediction tools. As previously explained in Chapter 2, intra prediction uses the previously encoded regions of the same image frame, while inter prediction exploits the temporal redundancy between different frames. Depending on the available prediction methods for each frame, video coding algorithms define different types of compressed frames, namely I, P and B frames. These frames types are used to define the Group Of Pictures (GOP) structure, *i.e.* a group of successive frames in a video sequence, which specifies the order in which I-, P- and B-frames are used.

The I-frame is an intra-predicted image frame, which only uses spatial prediction methods, does not depending on any other frame. I-frames are used as reference frames for temporal prediction of subsequent frames, through inter prediction methods. These frames are commonly positioned at certain instants along the video sequence, in order to provide random access function. This is possible because I-frames do not have any coding dependency from previous encoded frames and, consequently, the decoding process can be restarted. Other obvious use cases of I-frames are the first frame of a sequence and the compression of still images. In such cases, no temporal frame exists and only spatial redundancies can be exploited.

In addition to intra prediction methods, the P- and B-frame types may use inter prediction techniques to exploit temporal redundancies, that exist between neighbouring frames. For this purpose, most video encoders use Motion-Compensated Prediction (MCP) for inter prediction. As discussed in previous chapter, MCP basically consists on copying a reconstructed block from a previously encoded reference frame to the position of the current target block. Currently, in HEVC standard, the main difference between P and B frames is that the former can only estimate one block for MCP, while the latter may use bi-predictive coding by combining two motion compensation candidates.

Since the predicted signal barely matches the original signal, the residual signal (or prediction error) computed from the difference between the original and predicted signals must be determined and encoded. Typical video coding standards compress the residual signal using the transform coding and quantisation methods. The purpose of transform coding is to convert the residual signal, originally represented in spatial domain, into the frequency domain. Such a transformation decorrelates the residue signal and concentrates the signal energy in few coefficients, mostly at low frequencies. Quantisation is used to remove the least relevant coefficients and roughly approximate the remaining non-zero coefficients. This procedure intends to remove the irrelevant data, which is not perceptible by the HVS. Finally, the compressed representation of the residual signal generated after the quantiser output is entropy encoded. Entropy coding stage is used to further reduce the statistical redundancy of the symbols that represent the quantised residual.

In order to achieve good rate-distortion (RD) performance, image and video compression algorithms should use optimal coding modes at each situation. Formally, the goal of an encoder is to minimise the distortion D, subject to a maximum constraint R_c, on the number of used bits R (rate). This problem can be solved using Lagrangian optimisation, where the distortion term is weighted against the rate term [48]. The Lagrangian minimisation is an unconstrained problem given by:

$$\min J, \quad \text{where } J = D + \lambda R \,, \tag{3.1}$$

where J is the Lagrangian cost minimised for a specific value of the Lagrangian parameter λ. The λ value has a correspondence with the desired target bitrate R_c.

Since the relation between the coding modes and the Lagrangian cost J is unknown, typical video encoders evaluate the distortion and rate of each available coding mode, selecting the coding solution that provides the minimum Lagrangian cost J, for some predefined λ. This is an effective solution to achieve optimal usage of the available coding tools with highest RD performance. However, such a procedure involves an exhaustive search over all coding possibilities and existing parameters, which requires a lot of computational resources. To alleviate the computational complexity problem, practical encoder implementations often consider sub-optimal solutions, for instance, only a sub-set of coding modes or parameters are evaluated.

While the coding decisions are responsibility of the encoder, the decoder algorithm simply generates the reconstructed image or video using the optimal coding modes selected by the encoder, and signalled in the compressed bitstream. After parsing the bitstream and decoding the prediction symbols and residual information, the decoder generates the predicted signal. Then, the inverse quantisation and inverse transform are used to generate the decompressed residual signal. The final reconstructed signal is obtained by summing the residual to the intra or inter predicted signal. As the prediction methods depend on the previous processed information, the reconstructed signal is stored to be used as reference by intra and inter prediction methods. In the following sections, we describe some of the most important video coding implementations for this study.

3.2 Compression of 2D video

The H.265/HEVC [1, 2] is the most recent transform-based video coding standard developed by the Joint Collaborative Team on Video Coding (JCT-VC), which is a partnership of the two main standardisation organisations: the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). It provides a significantly improved compression performance relative to its predecessor H.264/AVC standard [5, 6], achieving an average bitrate reduction of 50% for the same subjective quality. HEVC addresses the main video coding challenges that appeared in the recent years, in particular the compression of high video resolutions beyond HD format, as well as the use of parallel processing architectures.

Other compression paradigms have been investigated and presented in literature as alternative to the dominant transform-based video coding standards. The MMP is one of these alternative algorithms, which belongs to the family of pattern matching compression schemes [7, 8]. MMP was initially proposed as a generic lossy data compression algorithm, but, as most improvements were focused on image coding applications, it rapidly became a successful image encoder. However, lossless versions of the MMP algorithm were also recently developed for image coding [49, 50]. Additionally, a wide range of signals have been encoded using the MMP paradigm, including natural images [7], text documents [8], compound documents [8], stereoscopic images [12, 51], depth maps [50, 52], as well as one-dimensional electrocardiogram signals [53].

Some recent works also propose the MMP paradigm for video compression using two different approaches. The first approach is based on the same architecture and coding techniques of H.264/AVC encoder, but the intra prediction and motion-compensation residues are encoded using the MMP paradigm, instead of the transform coding method [54, 55]. The second approach is an extension of MMP algorithm for three-dimensional signals, which processes the 2D video sequence as a volumetric signal based on 3D blocks, instead of using a frame-by-frame coding approach [56, 57].

3.2.1 H.265/HEVC standard

This section presents an overview of H.265/HEVC standard [1], focusing on the most relevant features of the encoding algorithm. A more detailed description of HEVC standard can be found in [2]. We also briefly present the main techniques introduced in HEVC extension for screen content coding [58].

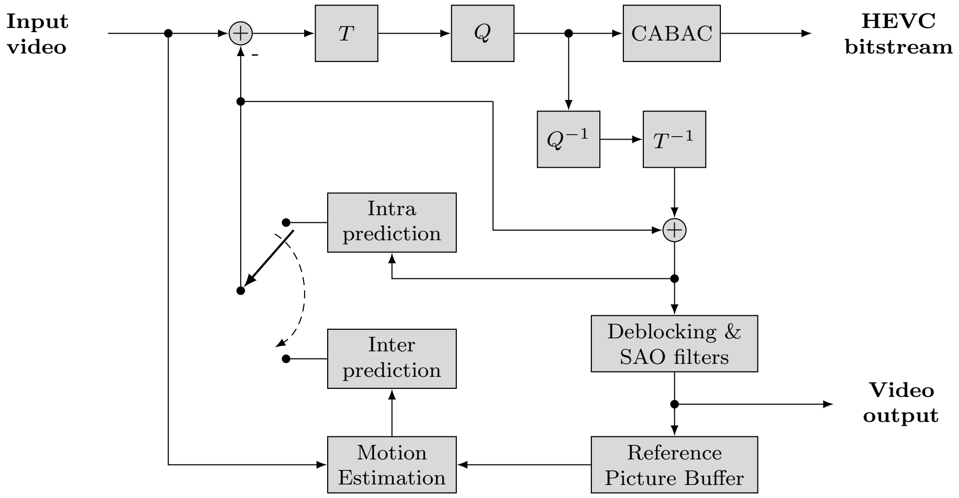


Figure 3.1: Architecture of HEVC algorithm.

Similarly to all previous video coding standards, HEVC only standardises the bitstream formulation and decoding process. This includes the bitstream structure, its syntax elements, the semantics of the bitstream elements and the process to generate the output images from the bitstream elements. Because the standard does not specify the encoder algorithm, the implementation of HEVC encoder is flexible enough to consider different optimisation levels, in respect to the computational complexity or compression quality. Thus, in order to be considered as an HEVC encoder, an encoding algorithm only needs to ensure that the produced bitstream complies the standard specifications.

Figure 3.1 presents the hybrid architecture of a typical HEVC encoder. In HEVC, each frame is partitioned into equally sized blocks which are processed independently. The encoder starts applying intra prediction methods and, for the case of P and B frames, it also tests inter prediction methods based on the motion estimation procedure. Then, the residual signal is computed, transformed by linear spatial transform, quantised and entropy coded. The resulting bits associated to the compressed residual, the chosen intra prediction modes, the motion information of inter prediction methods, as well as an auxiliary encoding information constitute the HEVC bitstream. In order to improve the overall quality of the reconstructed images and smooth the coding artifacts, HEVC also uses the Deblocking Filter (DBF) and the Sample Adaptive Offset (SAO) methods. During the encoding process, HEVC needs to replicate the decoder procedure, so that the same reconstructed causal samples in the current frame and previously encoded frames are available for intra and inter prediction methods. In the decoding loop, the compressed residual is delivered to the inverse quantisation and inverse transform modules of Figure 3.1. The reconstructed samples are generated by adding the decoded residual to the predicted signal and filtering with DBF and SAO methods. When one frame is completely encoded, the corresponding reconstruction, generated by the decoder output, is stored in the decoded picture buffer to be used as reference frame by the prediction methods of subsequent frames.

HEVC coding structures

Although HEVC follows the traditional hybrid coding architecture similarly to the previous video coding standards, it presents significant differences regarding encoding data structures. Instead of the macroblock structure of H.264/AVC, which uses 16×16 luma blocks and 8×8 chroma blocks, assuming YUV data with 4:2:0 sub-sampling, the HEVC algorithm defines the Coding Tree Unit (CTU), which is composed by the luma and chroma Coding Tree Blocks (CTBs) and associated syntax. The size of the luma CTB is configured by the encoder and signalled into the bitstream headers, allowing three possibilities: 16×16, 32×32 and 64×64 block sizes. The use of larger CTBs tends to provide better coding efficiency, specially for high-resolution video contents.

Each CTB represents a quadtree structure, with the partitions being denominated Coding Blocks (CBs). The Largest CB size (LCB) corresponds to the configured CTB size. The quadtree partitioning can be recursively applied until the smallest CB (SCB) is reached, which cannot be inferior to 8×8 pixels. Similarly to the LCB size, the SCB size should be configured by the encoder and signalled in the bitstream headers. A maximum number of four quadtree levels are possible, concretely when LCB is 64×64 pixels and SCB is 8×8 pixels. HEVC applies the same quadtree partitioning structure to luminance and chrominance CBs, except for the minimum size of chrominance blocks. The luma CB and corresponding chroma CBs constitute the Coding Unit (CU).

The CU is the root for two types of structures that can further partition the block, specifically the Prediction Unit (PU) and the Transform Unit (TU), which are formed by the luma and chroma Prediction Blocks (PBs) and Transform Blocks (TBs), respectively. The decision whether to use intra or inter prediction methods is made at the CU level. Thus, HEVC does not allow to use both intra and inter prediction methods in the same CU. However, different prediction modes of the same class (*i.e.* intra or inter) can be used within one CU, by partitioning it into multiple PUs.

When the CU is decided to be predicted using intra-frame methods, the size of the PB must be equal to the CB, except for the smallest available CB size. In the particular case of SCB, HEVC encoder may optionally partition it into four equally sized PBs, using one bit signalling flag. By partitioning the SCB into four PBs, different prediction modes can be used in the same CB, and smaller blocks can be used, such as 4×4 PBs when the SCB size is 8×8 pixels. The use of smaller

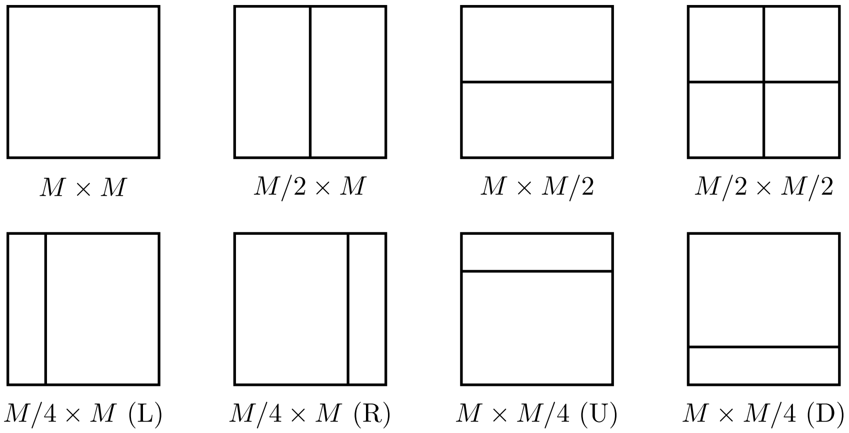


Figure 3.2: Partitioning modes of an inter-coded CB into PBs. Figure taken from [2].

blocks provides a higher prediction accuracy, namely for complex image regions. Although the PB corresponds to the image samples predicted by the same mode, the derivation of these samples, for the case of intra prediction modes, is done at TB level. The motivation for this procedure is to better reuse the reconstructed samples within the PB, specifically when the TBs (where residue coding is used) are smaller than the PB. An interesting non-normative research work which modifies HEVC standard to use non-square PUs for intra prediction is presented in [59].

For the case of inter-predicted CUs, HEVC defines a larger amount of partitioning modes, which may divide the luma and chroma CBs into one, two or four PBs. Figure 3.2 illustrates the available partitioning modes for inter-coded CUs. The $M \times M$ PB mode represents no partitioning at all, corresponding to a PB that matches the CB. The $M/2 \times M/2$ mode divides the CB into four equally sized PBs, but it only can be used on the SCB, when the minimum configured size is larger than 8×8 pixels. Note that, the use of the $M/2 \times M/2$ partitioning mode only makes sense at SCB, because otherwise it would overlap the existing quadtree partitioning for the CBs. The $M \times M$ and $M/2 \times M/2$ partitioning modes are the same as the ones available for intra prediction methods. The difference is that $M/2 \times M/2$ partition, in the case of intra prediction, can be used when SCB is set to 8×8 size. Because of this, only intra prediction can be used on 4×4 PBs. Regarding the remaining partitioning modes of Figure 3.2, the $M/2 \times M$ and $M \times M/2$ provide vertical and horizontal block partitioning, respectively. The lower four modes of Figure 3.2 are denominated as Asymmetric Motion Partitioning (AMP), since they divide the CB into two partitions with asymmetric sizes (one quarter and three quarters of the CB area). The AMP modes only are allowed when luma CB size is 16×16 pixels or larger.

In HEVC, the transform-based residue coding uses a quadtree structure with root at the CB. In such structure, the CB can be divided into four TB partitions, which in turn can be further partitioned, in a recursive fashion. The transform coding and quantisation are applied at the leaves of the resulting quadtree. The maximum depth of the residual quadtree, as well as the minimum and maximum TB sizes are initially defined at encoder configuration. The minimum luma TB size should not be inferior to 4×4 pixels and the maximum luma TB size should not be superior to 32×32 pixels. During the encoding procedure, a flag may be transmitted to indicate whether a residual quadtree partitioning occurs or not. When the minimum TB size is reached, or when CB is larger than the maximum possible size of TB (forcing block partitioning), the transmission of the flag is omitted. A particular feature of the residual quadtree is the possibility to use transform coding in one TB that spans across multiple inter-predicted PBs.

In addition to the block-based structures, HEVC defines the slice and tile structures for error resiliency and data parallel processing purposes. The slice is a sequence of CTUs processed in raster scan order, which belong to the same frame. One frame may contain one or more slices. Each slice can be decoded independently from other slices of the same picture, except for the effects of the in-loop filter near the slice boundaries. This implies, for instance, that intra-prediction methods do not use spatial neighbouring information across the slice boundaries. The main advantage of using slices is the ability to resynchronise the decoding process when transmission errors occur. The slices can be classified as I, P or B-slices.

The tile structure has been introduced in HEVC to enable the implementation of parallel processing techniques for both encoding and decoding processes. Tiles are rectangular regions of the image, that can be independently encoded and decoded, except for deblocking filter. The decoding of these regions may share some header information, namely when they are defined within the same slice. Another possible application of tiles is the random access in different spatial regions of one image.

The tiles provide a coarse solution for parallel processing without requiring any advanced synchronisation of threads. An alternative solution for a finer degree of parallelism, is to use the Wavefront Parallel Processing (WPP) technique. The WPP divides the slice into rows of CTUs and processes them in a synchronised way. The idea is to start processing one row of CTUs after two CTUs have been processed in the previous row. The advantage of WPP over tiles is the possibility to use prediction methods across WPP boundaries, which provides better coding efficiency.

Intra-frame prediction

As previously explained in Chapter 2, the intra-frame prediction plays an important role for reducing spatial redundancy of each frame. HEVC uses the Planar, DC and angular modes, which are defined at TB level with sizes that vary between 4×4 up to 32 × 32. A detailed description of these prediction modes has been previously presented in Subsection 2.3.1. Here, we describe some aspects related to the encoder implementation of intra prediction.

The encoder implementation proposed by MPEG, given by the reference HM software, uses the Rough Mode Decision (RMD) method to estimate the optimal intra prediction mode with reduced computational complexity [60]. Note that, this is a non-normative technique, as it is used at encoder side to speed up the intra RD optimisation process. The size of RMD candidates list depends on the PB size. For PBs larger than 8 × 8 pixels, RMD uses three candidates, while for PBs with 8 × 8 or 4 × 4 pixels, eight candidates are defined.

During the encoding procedure, the candidate mode list of RMD is built using a low complexity RD evaluation procedure, which avoids the compression of the residual signal. The RD cost function uses a distortion term based on the sum of the absolute coefficient values that result from the Hadamard transform of the prediction residual, and a rate term given by the number of bits used by HEVC to encode the prediction mode. The modes with lower RD cost are included in the candidate mode list. In addition to these modes, HEVC includes the MPMs, derived from the neighbouring PBs (see Subsection 2.3.1), into the candidate mode list. Once the candidate mode list is generated, the candidate modes are evaluated using full RD optimisation, where the residual signal is encoded by transform coding and quantisation, considering both the luma and chroma components, and using a simplified residual quadtree [61]. As a last step, the best mode obtained from the previous optimisation procedure is evaluated again using full RD optimisation, but without any restriction on the residual quadtree partitioning.

Inter-frame prediction

In HEVC, inter-frame prediction is carried out by the motion-compensated prediction, which has been previously described in Subsection 2.3.2. In order to perform motion estimation, HEVC stores the previously decoded pictures in the Decoded Picture Buffer (DPB). These pictures are identified by a Picture Order Count (POC) that is transmitted in each slice header. The POC is important, because the encoding order of the sequence frames can be different from the temporal picture order. This allows, for instance, to use future temporal pictures as reference pictures for MCP. Not all pictures of DPB are necessarily used as reference for MCP, and the set of reference pictures effectively used by MCP is configurable, being referred to as Reference Picture Set (RPS). Similarly to its predecessor, HEVC defines two lists of pictures in DPB, called reference picture list 0 and list 1. In the case of uni-prediction, MCP may use a picture from either of these lists, which should be signalled. For bi-prediction, one picture from each list is used.

The block matching-based motion estimation is one of the most computationally complex procedures in the HEVC encoder. The optimal solution for motion estimation is provided by the full-search algorithm, which tests all the possible candidate blocks of the search area and selects the best one. Despite its optimality, the full-search algorithm requires the highest computational complexity. However, as the motion estimation is a non-normative procedure, encoders may implement alternative search algorithms that provide sub-optimal solutions with reduced computational effort. Fast search algorithms, that avoid most of the search points, have been proposed in literature to alleviate the problem of computational complexity. An example of a fast search algorithm, used in the reference HM software, is the Test Zone Search (TZS) [62].

Transform and quantisation

Similarly to the previous standards, HEVC uses 2D transform and quantisation methods to encode the residual signal. The 2D transform is computed by applying the 1D transform in the horizontal and vertical directions. The core transform matrices are based on the Discrete Cosine Transform (DCT) basis functions and the used sizes include the 4×4, 8×8, 16×16 and 32×32, depending on the TB. For simplicity, only the 32×32 transform matrix is defined. Sub-sampled versions of this matrix are used to derive the remaining transform matrix sizes.

For the case of 4×4 luma residual blocks generated by intra prediction modes, HEVC uses an alternative integer transform based on the Discrete Sine Transform (DST). The motivation for DST is related to the statistics of the residual signal of intra predicted blocks, which tend to present higher residual amplitudes as the distance to the left and top block boundaries increases. Experiments have shown that DST provides 1% bitrate savings for the compression of intra predicted blocks, in comparison to the 4×4 DCT-based transform, without requiring a significantly superior computational complexity. DST is not used for other TB sizes, because the compression efficiency gains are irrelevant.

Regarding the quantisation method, HEVC follows a similar procedure as H.264/AVC standard, known as Uniform-Reconstruction Quantiser (URQ). The quantisation is the method responsible by the coding losses, given that it performs an irreversible many-to-one mapping of the transform coefficients. In general terms, URQ procedure consists of an integer division of the transform coefficients by a quantiser step. The value of the quantiser step depends on the Quantisation Parameter (QP) defined in the encoder configuration, which may vary from 0 to 51.

Entropy coding

In contrast to H.264/AVC standard, which allows two entropy coding methods, HEVC only uses the Context Adaptive Binary Arithmetic Coding (CABAC) [63]. CABAC is an arithmetic coding method that only uses binary symbols (bin), considering different probability models for each bin. The context model selection has an important weight in the entropy coding efficiency, being carefully designed in HEVC standard. For example, in addition to the spatial neighbouring information, which is used in H.264/AVC standard, the HEVC standard exploits the depth of the partitioned coding tree or residual transform tree in order to derive the context models for several syntax elements.

Regarding transform coefficient coding, CABAC uses a scanning method to organise the coefficients and transmits the position of the last non-zero transform coefficient, a significance map, the sign bits and the levels for the transform coefficients. Three coefficient scanning methods are available: the diagonal up-right, the horizontal and the vertical scan. The coefficient scanning is implicitly selected and always performed in 4×4 sub-blocks for all TB sizes. Details about the context models selection and the compression of each syntax element of HEVC can be found in [1].

In-loop filters

As previously referred, HEVC may use two filtering procedures over the reconstructed samples before writing them into the DPB, namely the DBF and the SAO filters. The main purpose of DBF is to reduce blocking artefacts caused by the block-based coding tools, operating similarly to the DBF of H.264/AVC standard. DBF is applied to the samples adjacent to the PU or TU boundaries, that are aligned on a 8×8 sample grid, for both luma and chroma components. Such grid-based restriction reduces the computational complexity and improves parallel-processing implementations. Three strength levels can be used by DBF depending on the coded block characteristics.

The SAO is a non-linear filter, that is adaptively applied to all samples of the image, after the DBF filter. In its operation, the SAO filter modifies the samples by adding an offset value, extracted from lookup tables transmitted by the encoder. For each CTB, the encoder decides whether to apply the SAO filter or not, and if used, one of the two available filter types is applied, namely the band offset or the edge offset. In the band offset mode the added offset value depends on the sample amplitude, while the edge offset uses the gradient to classify and derive the offset value.

HEVC screen content coding extensions

In the context of the increasing number of applications using non-camera-captured video content, such as computer screen sharing, wireless display or cloud gaming, the JCT-VC partnership initiated a new standardisation process on HEVC extensions for screen content coding (SCC), referred to as HEVC SCC [58, 64, 65]. HEVC SCC introduces new video coding tools within the basis HEVC technology, in order to improve its coding performance for the screen content type, which includes rendered graphics, text, animation, natural and mixed data. In the following, we review the main techniques available in the current draft of HEVC SCC standard (draft version 5) [65], namely the Intra Block Copy (IBC), Palette Coding, Adaptive Colour Transform (ACT) and Adaptive Motion Resolution (AMR).

The IBC method consists of BMA using the previously encoded region in the same frame as reference for the search procedure. While MCP uses BMA to estimate motion information based on the previously processed frames, the IBC tries to predict repetitive patterns that exist in the same frame. Such technique is motivated by the fact that screen content presents a different spatial correlation than natural content. For instance, video with text and graphics often presents repetitive patterns within the same frame, thus being efficiently represented by IBC. Similarly to MCP, the IBC method estimates a displacement vector that indicates the position of the predictor block relative to that of the current PU to be predicted. The residual signal is encoded using the existing transform-based methods of HEVC algorithm.

Due to the similarities between IBC and inter prediction methods, HEVC SCC unifies the design of these modes. As consequence, the IBC is reproduced by adding the current frame as reference picture for inter prediction modes. The whole reconstructed region of the current frame (within the same slice and tile), before the in-loop filters in the encoder, is used for block searching. However, in order to allow the Wavefront Parallel Processing, the top-right regions of the current block cannot be considered in the search window, as illustrated in Figure 3.3. IBC is executed at PU level, being able to use all the PU sizes available for inter prediction. The estimated vectors use full pixel accuracy.

The purpose of Palette Coding method is to efficiently represent image blocks that contain a small number of representative colour values. This kind of image blocks are more frequent in screen content than natural video. The text regions are a clear example where only the foreground text colour and background colour are present. The Palette Coding is used at CU level, instead of intra prediction and transform-based residue coding. In its procedure, the Palette Coding method classifies all CU pixels into a list of major colours (with higher frequency of occurrence in the CU). An index table, referred to as palette, maps each major colour of the CU

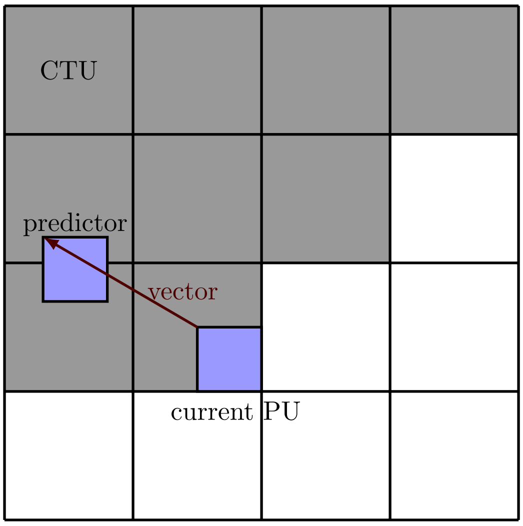


Figure 3.3: Search window used by IBC (represented by gray CTUs). Figure taken from [58].

to an index. All the CU pixels are converted into the respective indices, excluding some isolated pixels with colour values not belonging to the palette. For such pixels, an escape index is used. The indices are signalled to the decoder based on run-length coding, using either the left or top neighbour as predictor. For each escape pixel, its actual value is explicitly transmitted together with the escape index. Details about the Palette Coding procedure can be found in [58, 64].

The Adaptive Colour Transform is used at CU level to remove some redundancy that may exist between the components of RGB colour space, even after the encoder prediction step. For the case of screen content, the image signal is often encoded using the RGB colour space. It tends to provide more effective coding results, specially when very saturated colours exist. However, for some image regions, RGB components may present a high redundancy between them. For such cases, the ACT method converts each residual pixel of the CU into an alternative colour space that concentrates their energy. As the ACT is applied on the residual signal, it cannot be used with Palette Coding, because there is no residue coding.

In HEVC SCC, Adaptive Motion Resolution consists of a slice-level flag that indicates whether the current slice uses full-pixel motion vectors or not. When full-pixel precision is defined, there is no need to estimate fractional motion, neither to transmit the bits representing fractional precision. The main motivation for AMR is the fact that screen content is often generated with knowledge of the pixel positions. Consequently, the existing motion is often aligned with the image pixels. In these situations, the bitrate associated to fractional portion of motion vectors can be saved.

3.2.2 Multidimensional Multiscale Parser algorithm

Although the original MMP paradigm can be used to encode any N-dimensional signal, the research work of the last years on MMP algorithm has been focused on image and video compression applications. Currently, several proposals of MMP algorithms can be found for image and video coding [7, 8, 55]. Since the research work of this thesis is focused on the image coding implementation of MMP algorithm, the video versions of MMP are not discussed. Therefore, in this section we present an overview of the MMP-intra algorithm [7, 8], which has been proposed for generic image coding.

Apart from the residue coding module, which is based on the pattern-matching paradigm, the MMP-intra algorithm presents a hybrid coding architecture similar to other image coding standards, including the prediction stage, residue coding and entropy coding modules. The main idea of MMP paradigm is to approximate the prediction error using elements from an adaptive multiscale dictionary. By reusing the previously encoded patterns, MMP is able to learn the signal features and better encode the redundant information.

MMP begins dividing the input image into 16×16 non-overlapping blocks which are sequentially encoded. Each block may be recursively divided according to a flexible block partitioning rule [66]. Each partitioning occurs in either vertical or horizontal directions, producing two equally sized sub-blocks. By applying this partitioning rule down to 1×1 pixel, a total of 25 block sizes (or scales) are defined by all the possible combinations: $2^m \times 2^n$, for $m, n = 0, ..., 4$, as illustrated in Figure 3.4.

Predictive coding was introduced in MMP-intra encoder in order to improve its coding performance, especially for smooth images [7]. MMP uses a hierarchical prediction framework based on ten prediction modes, which include the Most Frequent Value (MFV) mode [67], eight directional modes inspired on the ones used in H.264/AVC encoder and the intra LSP mode [9]. The prediction step is applied on sub-blocks obtained through the MMP flexible partitioning scheme, having dimensions from 16×16 down to 4×4, as represented in Figure 3.4 by gray blocks. Due to the prediction step, the residual blocks tend to be homogeneous, with a probability distribution that is highly peaked and centred around zero. Experimental tests have shown that this peaked distribution favours the adaptation process of the MMP dictionary and increases the coding performance of the algorithm [7].

The residue generated by each prediction mode is approximated using the MMP coding paradigm. In this process, a given residue patch is recursively partitioned to find the optimal block representation. In order to decide whether a block should be partitioned or not, the rate-distortion cost $J = D + \lambda R$ is evaluated, where λ is the

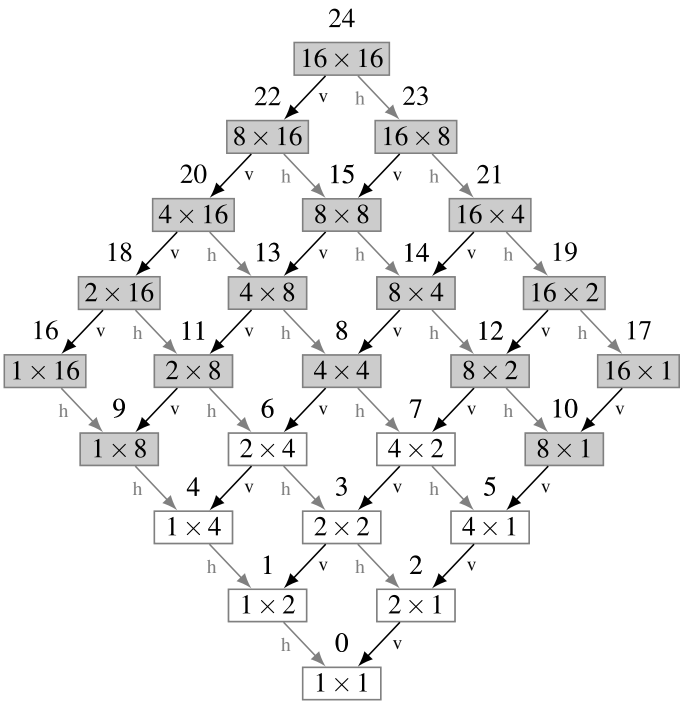
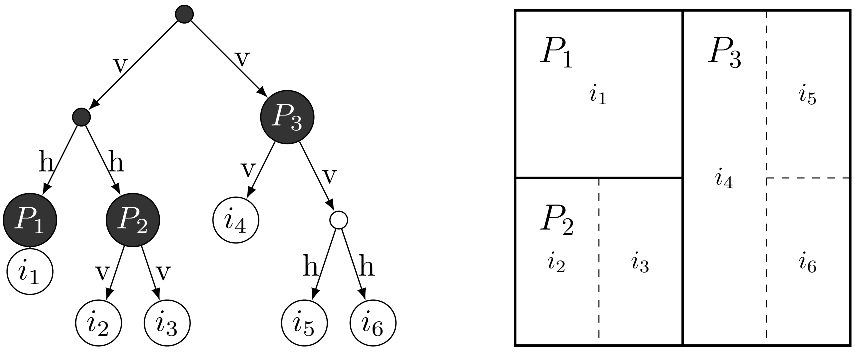


Figure 3.4: Possible block sizes (or scales) of MMP-intra algorithm.

Lagrangian multiplier that defines the target RD point, R is the rate required for the representation and D is the representation distortion given by the Sum of Squared Errors (SSE). When encoding a block X^l belonging to scale l, the dictionary pattern S_i^l that minimises the cost function J is chosen. Then, the block is divided into two equally sized sub-blocks, X_1^{l-1} and X_2^{l-1}. The patterns from dictionary scale $l-1$ that better represent the two sub-blocks are chosen, according to the Lagrangian cost function. If the sum of the Lagrangian costs of each sub-block is smaller than the cost of the parent block representation, then a block partitioning occurs. This procedure is recursively performed by evaluating both the horizontal and vertical partitioning directions until all the possible block sizes are tested, down to 1 × 1 pixel.

The MMP block partitioning is performed in two different stages: prediction and residue coding. Each 16×16 initial block can be divided into sub-blocks that are predicted using different prediction modes. This defines a segmentation tree where each node corresponds to a block partitioning and each leaf corresponds to the sub-block where the prediction mode is used and the residue pattern is generated. Figure 3.5 illustrates the MMP segmentation tree for one block, where the prediction tree is represented by dark circles. Horizontal and vertical block partitioning directions are represented on the tree, by "h" and "v".

Each leaf P_n of the prediction tree corresponds to a residue block, which can be further partitioned into smaller blocks, using the same rules as for the partitioning of prediction blocks. Thus, the final residue block can be approximated either by



(a) (b)

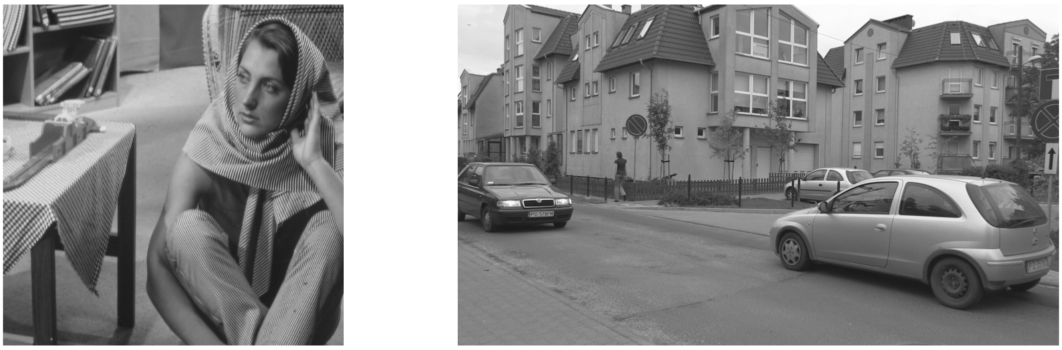
Figure 3.5: Example of an optimal MMP segmentation tree for some block (left) and corresponding partitioned block (right).

one dictionary element (*e.g.*, P_1 in Figure 3.5) or by the concatenation of several dictionary elements, from different scales (*e.g.*, P_2 and P_3, in Figure 3.5). Each used dictionary block is represented by its index, i_n. Note that, in Figure 3.5 solid lines represent prediction block partitioning, while dashed lines represent the residue sub-blocks.

The optimal MMP block partitioning is represented by a binary segmentation tree, that contains all the required information to generate the block approximation. The corresponding bitstream is constructed by scanning the tree from top to bottom, coding all nodes and leaves, using Adaptive Arithmetic Coding (AAC) [68]. Due to the flexible block partitioning, a binary flag needs to be transmitted at each node to signal whether the partitioning direction is horizontal or vertical. In order to identify each tree leaf, a binary flag is transmitted before the index of the dictionary pattern that approximates the residue block for the leaf.

The adaptation of the MMP dictionary is a key factor for its coding performance. During the encoding process, the dictionary patterns used to represent the residue blocks are concatenated and added to the dictionary. This procedure is strictly replicated in the decoder side, so that the dictionary is synchronised and does not need to be transmitted. The MMP dictionary is organised into multiple scales, *i.e.* block sizes. When an element is added to the dictionary, expanded and contracted versions of that element are computed and inserted into the corresponding dictionary scales. This procedure ensures that the new block will be available to encode future blocks, irrespective of their dimensions. As the MMP dictionary is updated after each coded block of size 16 × 16, it rapidly learns the image's features.

Several improvements have been proposed to MMP algorithm, such as the usage of geometric transformations over the concatenated elements to enrich dictionary updating, an efficient redundancy control scheme that limits the insertion of similar elements in dictionary, a norm-equalisation procedure to adapt the new elements to



(a) Barbara (512 × 512)

(b) Poznan Street, camera 4, frame 0 (1920 × 1088)

Figure 3.6: Used test images: Barbara (left) and Poznan Street, camera 4, frame 0 (right).

the residue statistical distribution, among others [7].

3.2.3 Experimental results

In this section we present a brief performance evaluation of the H.265/HEVC and MMP algorithms for generic image and video coding. The results of these methods are compared with the ones of the previous H.264/AVC standard. Since the MMP-based video coding implementations are out of scope of this thesis, we restricted the analysis of MMP performance for image coding scenario.

Two test images and two test sequences, with different resolutions, were selected for the experiments. The test images are the well known Barbara image, with 512×512 pixels, and the first frame of Poznan Street (camera 4), with 1920×1088 pixels. Figure 3.6 illustrates the luminance component of these test images. The video sequences were selected from the HEVC common test conditions [69], namely the Race Horses from class C (832×480), with 300 frames (30 fps) and the Basketball Drive from class B (1920×1080), with 500 frames (50 fps). This is a representative selection of natural image and video signals used along this thesis.

In order to evaluate the compression performance of the tested algorithms, we adopted the most widely used and known objective distortion metric, specifically the Peak Signal-to-Noise Ratio (PSNR), as a function of the compression ratio. The PSNR metric is usually measured in decibels (dB) and, for the case of the 8-bit pixel images, it is defined as:

$$PSNR_{dB} = 10 \log_{10} \frac{255^2}{\frac{1}{M.N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(I(m,n) - \hat{I}(m,n)\right)^2} = 10 \log_{10} \frac{255^2}{MSE(I, \hat{I})}, \tag{3.2}$$

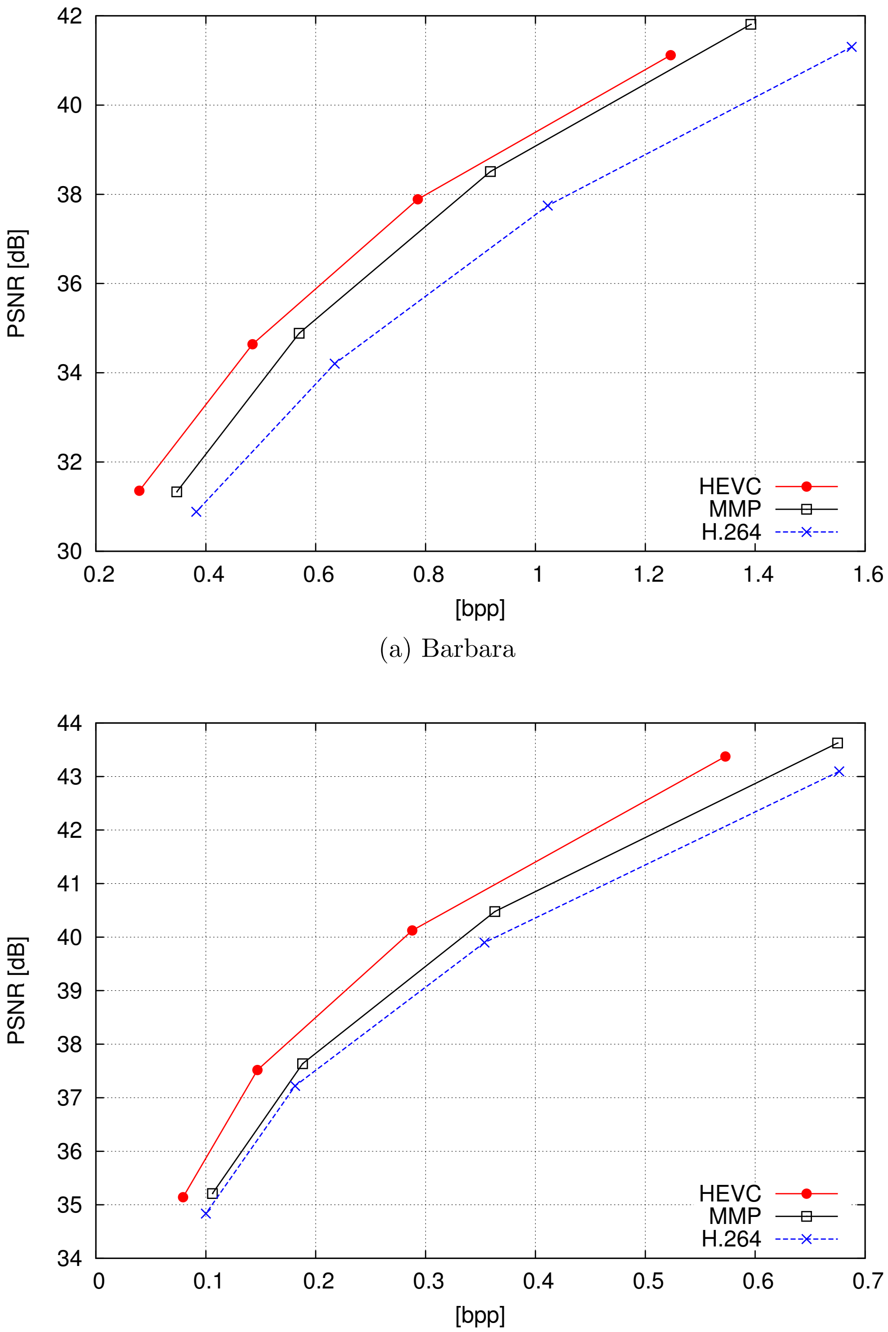
where M and N are the image dimensions in number of pixels, I is the original image and $\hat{I}$ is the decoded image, that presents the coding distortions. The 255 value in the numerator of the equation is the maximum value of the 8-bit pixel image. MSE is the Mean Squared Error function between two images. For the case of video sequences, the average PSNR of all frames is used.

The compression ratio of image signals is usually measured in terms of bits-per-pixel (bpp), *i.e.* the average number of bits used to represent each pixel of the image. For the case of 8-bit pixel gray images, the maximum bpp value is 8, which means no compression at all. Regarding video sequences, the bitrate in kilobits-per-second (kbps) is frequently used. The bitrate can be computed based on the average number of bits used per frame and the frame rate value, which is a characteristic of the video sequence.

Along this thesis, we will make extensive use of RD plots representing the PSNR as a function of the compression ratio or bitrate for evaluating the proposed coding tools. Another evaluation method used in this thesis is the Bjontegaard Delta PSNR (BDPSNR) or Bjontegaard Delta Rate (BDRATE) [70]. These metrics allow to compare two RD curves, commonly from different encoding algorithms, by means of a single scalar value that traduces the average PSNR improvement (BDPSNR) or the average percent bitrate saving (BDRATE), between the two coding methods associated to the compared RD curves.

In these experiments, we used the H.264/AVC encoder implementation provided by JM-18.5 reference software [71] and the HEVC encoder implementation provided by HM-14.0 reference software [72]. Regarding the image compression experiments, we used H.264/AVC with high profile for intra frame coding and the HEVC with main still picture profile. The QP values recommended in HEVC common test conditions document [69], namely: 37, 32, 27 and 22, were used for both transform-based coding standards. For the case of MMP algorithm, the Lambda parameter is used to specify the RD points. We selected the Lambda values 150, 50, 15 and 5, because they approximately comprehend the same range of PSNR values as the QP values used for transform-based standards. Note that, although these algorithms have been conceived for video compression, it has been shown that their performance for still image coding application is superior than the one of the traditional image encoders [22], such as JPEG [73] and JPEG2000 [74].

Figure 3.7 presents the RD performance of the HEVC, MMP and H.264/AVC for Barbara image and first frame of Poznan Street (camera 4), specifically the PSNR of luminance component as a function of the luminance compression ratio in bpp. For both images, the state-of-the-art HEVC standard presents the best RD performance. Regarding the MMP algorithm, these results confirm its superiority over H.264/AVC, as shown in literature [22]. However, one may observe that its RD performance is inferior than the one of the most recent transform-based HEVC



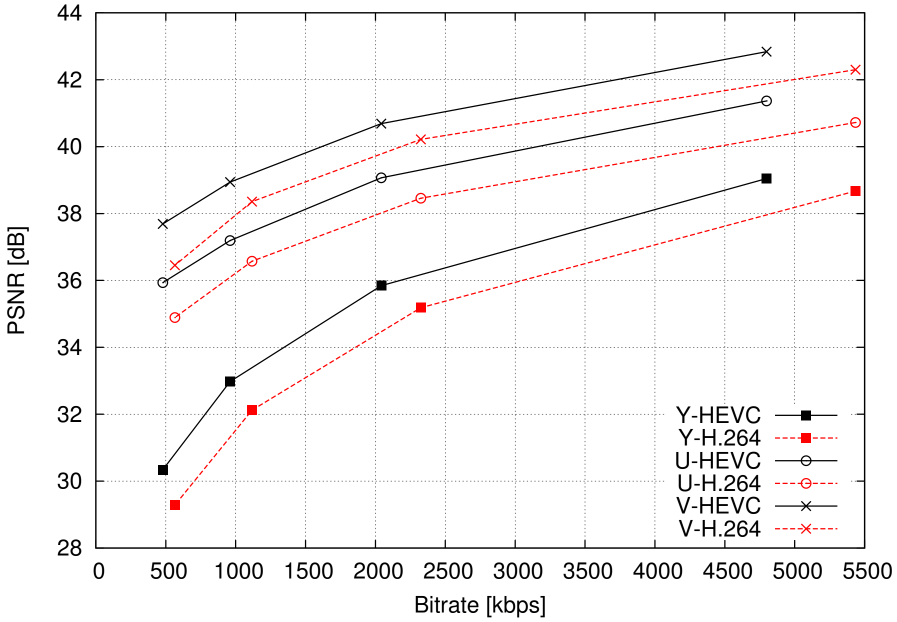
(a) Barbara

(b) Poznan Street, camera 4, frame 0

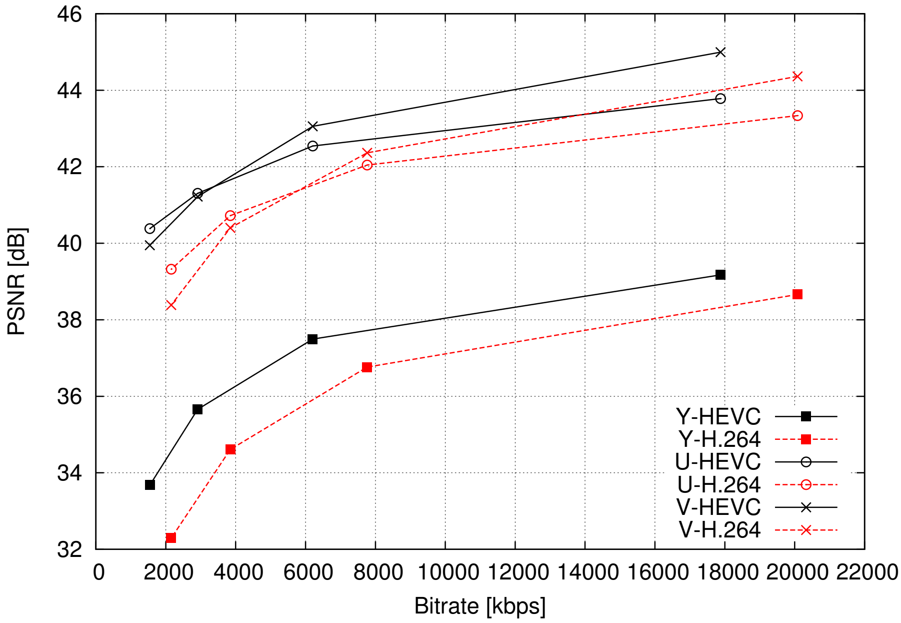
Figure 3.7: Rate-distortion performance of HEVC, MMP and H.264 for test images Barbara and Poznan Street, camera 4, frame 0.

standard. The disadvantage of MMP relative to HEVC standard constitutes one of the motivations for the first research topic addressed in this thesis, in Chapter 4. One may also observe that the gains of MMP algorithm relative to H.264/AVC are higher for Barbara image than for the Poznan Street frame. This is because Barbara image presents textured areas made of repeated patterns that are better predicted by the LSP mode used in MMP algorithm [9].

Regarding the experiments with test sequences, the HEVC random access main



(a) Race Horses (class C)



(b) Basketball Drive (class B)

Figure 3.8: Rate-distortion performance of HEVC and H.264 for test video sequences Race Horses (class C) and Basketball Drive (class B).

profile, with an intra period of 32 frames and the GOP size of 8 frames, has been adopted. Additional details of the used encoder configuration are available in the default random access configuration file that comes with HM-14.0 reference software. For H.264/AVC standard, we used the HM-like random access configuration, that comes with the JM-18.5 reference software, setting the intra period to 32. The used QP values were the same as the ones used for the image coding experiments.

Figure 3.8 presents the RD performance of HEVC versus H.264/AVC for Race

Horses and Basketball Drive test sequences, plotting the PSNR of luminance (Y) and chrominances (U and V) as a function of the total bitrate in kbps. These results show that HEVC algorithm presents a consistent advantage over the H.264/AVC standard, for both test sequences. The gains of HEVC are more expressive for the Full HD resolution Basketball Drive sequence, demonstrating the superiority of this standard for the compression of high resolution test sequences.

3.3 Compression of 3D video

In recent years there has been a growing interest in 3D video contents, mainly motivated by the recent advances on 3D display and transmission technologies, as well as by the increasing production of 3D contents. 3D video introduces the depth effect on the observed content, providing a more immersive and powerful viewing experience for the users. Due to its greater realism, 3D systems have been used in many applications. Beyond the cinema and entertainment markets, 3D technology is found in professional applications, such as medical surgery, robotics, education, among others.

In this section, we describe the main 3D technologies that have been used along this thesis, focusing on the compression methods proposed for the 3D representation formats. A brief overview of the most recent 3D-HEVC video coding standard, that is based on HEVC technology, is also presented. Furthermore, the performance of 3D-HEVC algorithm is illustrated and discussed. A more detailed description of 3D-HEVC standard can be found in [23, 75].

3.3.1 3D video systems

Two definitions have been used to classify 3D technologies, namely the 3D Television (3DTV) and the Free-viewpoint Television (FTV) video [76, 77]. The differentiation of these concepts has historical reasons, being more related to the activity focus, since the differences in the technology are not clear. The 3DTV video concept is commonly associated to the ability for offering depth perception of the observed scenario, while the FTV allows interactive changing of the viewpoint within some operation range, similarly to computer graphics systems [78]. The 3DTV and FTV video are not mutually exclusive, since a single 3D system may provide depth perception and free navigation.

A common element between the 3DTV and FTV technologies is the use of multiple views of the same scene, that need to be encoded and transmitted. Depending on the amount of used views and whether geometry information is used or not, several 3D scene representation formats can be defined. The choice of the 3D representation format has a large influence in the design of the 3D video system, specifically in the type of decoder, capture system, display technology, required bandwidth and system complexity. For example, geometry-based representations usually require the use of view synthesis (or rendering) algorithms on the decoder side [79, 80], which has strong implications on its complexity.

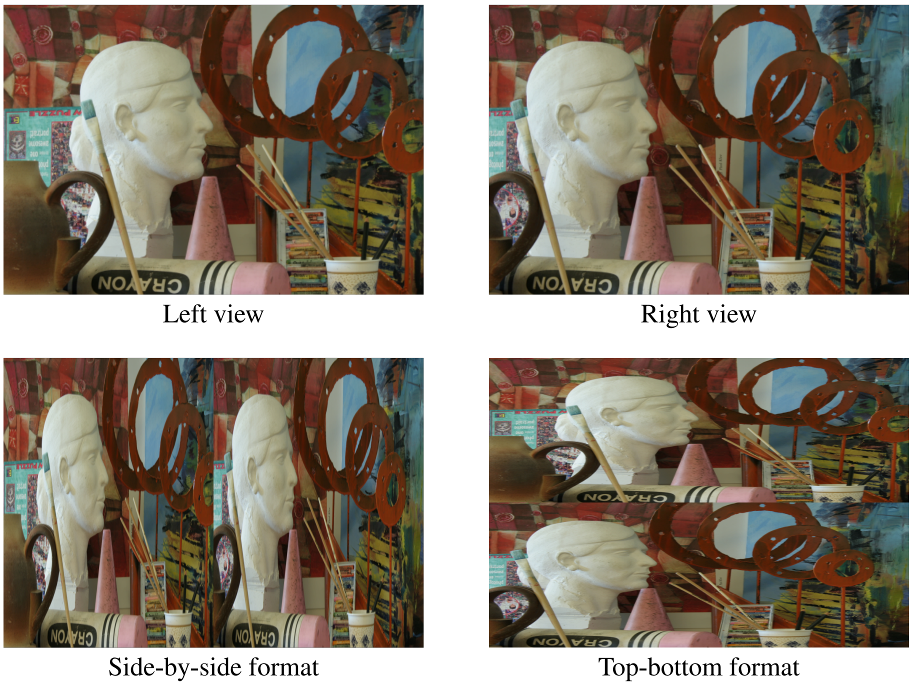
In the following, we review some of the most important 3D representation formats that have been used in this thesis, namely the stereo, the multiview, the multiview video+depth and the holoscopic video. The presented review for each 3D video format addresses the capturing, coding and displaying stages of the 3D data processing chain.

Stereo video

The stereo concept is related to the creation of depth illusion in the observed images. Such a functionality is provided by stereoscopic systems which exploit the binocular vision, specifically by separately presenting two slightly different images to the left and right eyes of the viewer. The human brain combines the two received images to give the perception of 3D depth in the observed scene. The stereo video is the conventional 3D representation format for stereoscopic systems [81]. This format simply uses two video views, which is the minimum number of views required to provide depth perception. Because of this, stereo video is also the simplest solution for 3D representation.

Typically, the acquisition of stereo video requires two capturing sensors installed in one or two independent video cameras. The two video views present significant similarities, since they represent the same scene captured from slightly different viewpoints. The distance between the left and right camera sensors (baseline distance), often corresponds to the average distance between the human eyes.

Each view of the stereo video can be separately encoded and transmitted, in a process known as simulcast coding. Although it provides backward compatibility with existing compression algorithms, this approach does not exploit redundant information present between views, requiring approximately double the rate of a single-view video. To improve the compression of stereo video, extensions to the state-of-the-art coding algorithms, such as H.264/AVC and HEVC, have been deployed. These stereo extensions introduce inter-view prediction methods that exploit the redundancy between views. Typically, the stereo video codecs independently encode the first view using the traditional single-view coding techniques, while the second view is dependently encoded using the same coding tools plus inter-view prediction methods that exploit similar information from the first view. To further improve the compression of stereo video, an uneven bitrate allocation between the two views can be also performed. This procedure is motivated by the binocular


Left view

Right view

Side-by-side format

Top-bottom format

Figure 3.9: Frame-compatible formats for stereo video: top images represent an example of a stereo pair; bottom images represent the side-by-side and top-bottom formats of the given stereo pair.

suppression theory [82], which states that the stereo vision quality is dominated by the quality of the higher fidelity view.

Another solution to represent the stereo video are the so called frame-compatible formats [83]. The purpose of these formats is to facilitate the introduction of stereo video in the existing infrastructure and transmission equipments. The common approaches multiplex the two stereo video views into a single frame, so that the conventional 2D video codecs can be used to process the stereo video. Examples of two frame-compatible formats, namely side-by-side and top-bottom ones, are illustrated in Figure 3.9. Other frame-compatible formats include row interleaved, column interleaved and checkerboard [83]. Alternatively, the stereo video views can be multiplexed in time, by interleaving right and left frames. Although frame-compatible formats are easily deployed, they may cause some losses in spatial and temporal resolution. Furthermore, the RD performance of frame-compatible formats is inferior to the one of the stereo video codecs explicitly designed to exploit inter-view dependencies.

Regarding the displaying technologies for stereo contents, there are many available solutions that map each stereo view into the respective eye. The most common solutions for stereoscopic displaying require the use of auxiliary glasses. Depending whether the displays interact or not with the glasses, these solutions may be classified as active or passive systems. An example of an active system is based on shutter

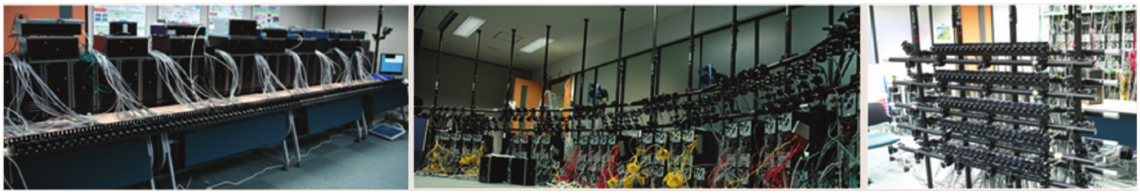


Figure 3.10: Three different arrangements for a 100-camera capturing system: linear (left), circular (centre) and 2D-array (right). Figure taken from [78].

glasses electronically controlled by the display. Passive systems can use polarised glasses, colour anaglyph glasses, chromadepth glasses, among others.

Alternative displays, that do not require the use of proper glasses, are known as autostereoscopic displays. Although, many autostereoscopic displays are based on stereo video, presenting only one observation point, most recent technologies use the multiview video format. The fact that the stereo video only captures one observation point of the scene is a drawback, because the viewer cannot observe different viewpoints of the scene, neither experience the motion parallax effect.

Multiview video

While the stereo video only uses two views, the multiview video format involves a larger number of views representing the same 3D scene. The use of multiple views is important to provide more interactive applications, that allow to change the observation point of the scene. Typically, the multiview video is captured using an array of cameras, that may be configured using different arrangements and densities. Figure 3.10 illustrates three possible camera arrangements, namely the linear, the circular and the 2D-array arrangements, for the 100-camera system proposed in [78].

The deployment of a multi-camera system is a challenging task, mainly due to the difficulties related to the synchronisation of the cameras and the handling of large amounts of data, specially in real time applications. However, for the case of static scenes, this capturing process can be highly simplified. Instead of using multiple cameras, a single camera can be used to record the same scene from several positions at different temporal instants, given that the lighting conditions do not change. Possible solutions include the use of robots to change the camera position or turn tables to rotate the scene.

Multiview video systems tend to produce huge amounts of data that need to be processed and transmitted. The processing stage usually involves the correction of some discrepancies and artifacts that exist between the captured views, namely by using algorithms for geometrical calibration, colour correction or to adjust lens distortion [84]. These systems often require the use of advanced computer clusters, mainly for the case of highly dense camera arrangements.

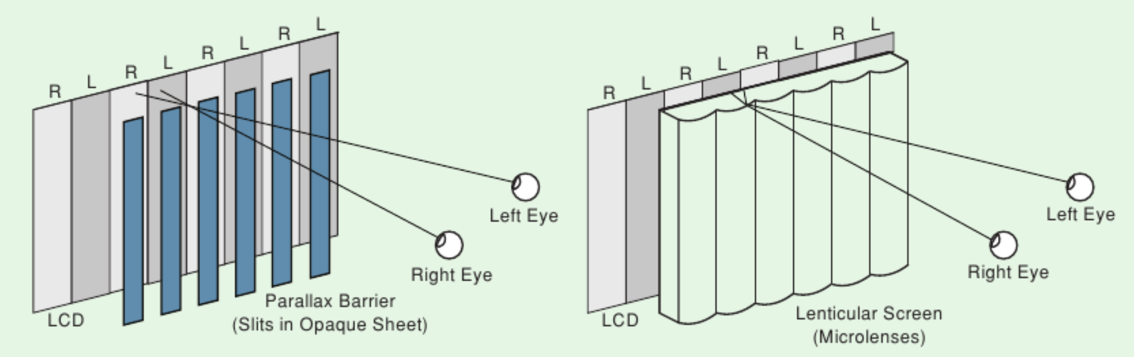


Figure 3.11: Two common autostereoscopic displaying technologies: parallax barrier (left) and lenticular screen (right). Figure taken from [88].

Due to the increased amount of data associated to the multiple views, it is crucial to use efficient compression algorithms for the multiview video format. The state-of-the-art coding approaches exploit the inter-view dependencies across several views, in addition to the spatial and temporal redundancies. The Multiview Video Coding (MVC) [5, 85, 86] is an amendment to the H.264/AVC standard for multiview video coding, that enables efficient inter-view prediction by adding decoded frames of other views to the reference picture list. A multiview extension for the HEVC standard has been also standardised with the designation of MV-HEVC [23, 87]. Although the multiview extensions of current state-of-the-art standards provide significant coding gains, the complex dependencies associated to the temporal and inter-view prediction limit the random access capabilities and increase the system delay, memory requirements and computational complexity of the encoder.

For displaying multiview video contents, autostereoscopic displays based on multiple views, typically eight or sixteen views, are used. These displays do not require the use of glasses, since they perform spatial multiplexing of distinct views into each eye using a light-directing mechanism. By using many views, the autostereoscopic displays may provide free navigation (within some limited spatial range) as well as motion parallax effect.

The autostereoscopic displays can be classified as active, when they track the head position of the user, or passive, when a large number of views is displayed simultaneously based on spatial multiplexing [88]. Two common technologies used by autostereoscopic displays are illustrated in Figure 3.11, namely the parallax barrier, based on slits in opaque sheet, and the lenticular screen, which uses thin cylindrical lenses. The main drawback of these displaying technologies is the blurring effect caused due to pixel crosstalk.

A common problem of the autostereoscopic displays is the visual fatigue and discomfort, caused by the accommodation-convergence conflict [89]. An alternative displaying technology which avoids this kind of problems and provides a real 3D

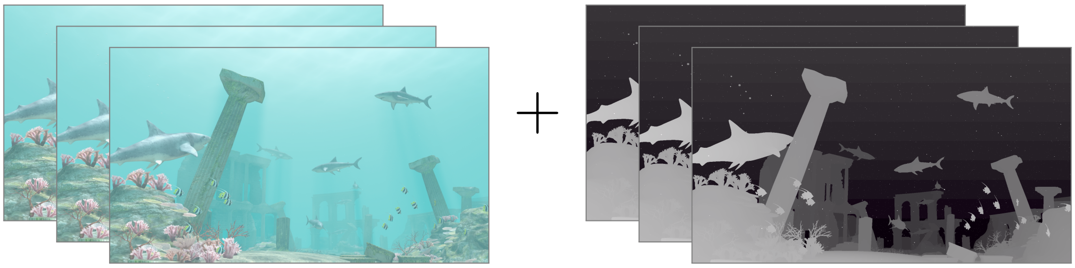


Figure 3.12: Example of MVD format using three texture video views (left) and three depth map views (right).

sensation is the holography. Although this technology is commonly associated to light field representation, it is possible to generate holograms from multiview video data as shown in [90].

Multiview video+depth

One of the main problems of the multiview video is the large number of views that need to be encoded and transmitted. Despite the significant RD gains provided by the multiview extensions for the state-of-the-art video coding standards, the total bitrate tends to increase linearly with the number of views. An alternative solution for 3D video representation makes use of the geometry information of the scene to reduce the number of required views [91, 92]. The idea is to transmit fewer texture views and render the missing ones, using the well-known Depth Image Based Rendering (DIBR) algorithm [79, 80]. This algorithm makes use of the geometry information, usually provided by the depth maps, to warp the few transmitted texture views to the desired virtual camera positions, and thus generate the virtual texture views. This kind of 3D video representation, that combines the multiview video with depth information, is known as Multiview Video+Depth format (MVD). Figure 3.12 illustrates an example of MVD data based on three texture video views (left) and three depth map views (right).

Three different methods can be used to obtain depth information, depending on the scenario and known data. For the case of 3D scenes whose geometry model is known, such as those used on 3D computer graphics applications, depth maps can be directly extracted, given the camera parameters. A solution for capturing depth data from the world scene consists in using active depth range cameras [93]. These cameras are able to measure the distance between each point of the scene and the camera by using appropriate techniques, such as time-of-flight, structured light or light coding. However, these cameras have some limitations, mainly because they tend to generate low resolution depth maps and the objects need to be next to the depth camera. Furthermore, the depth is captured at a slightly different position than the texture data, because both sensors cannot occupy the same physical position. Another solution to obtain depth information is based on stereo matching algorithms that estimate depth data from a pair of texture views [94]. The principle of these algorithms is to find matching points of the same objects between the two images, *i.e.* the disparity at each point, and then derive the depth information by triangulation. Although there many algorithms to perform depth estimation, these methods present several issues related to the mismatching errors that often occur at smooth regions and occluded areas. One solution to minimise some mismatching errors and reduce the computational complexity of depth estimation algorithms is to use horizontally aligned cameras or apply a rectification process that virtually rotates the textures views, so that the vertical component of disparity is zero and the search procedure only needs to be performed in the horizontal direction.

In order to efficiently compress the MVD data, appropriate methods that exploit similarities of both texture video and depth maps are required. Depth maps are piecewise-smooth signals which are made of smooth areas associated to several depth values separated by sharp edges at object boundaries. Although the state-of-the-art multiview video encoders, such as the MVC or the MV-HEVC, can be used to encode the multiple views of depth maps, as done with multiview video, these coding standards were not optimised to encode depth map data.

The use of transform-based coding standards for depth map compression tends to create coding artifacts, specifically edge smoothing and ringing effects. This kind of coding errors on depth maps usually reduces the performance of the rendering procedures, causing severe distortions in the synthesised views. In order to better compress depth maps and improve their performance for rendering applications, several coding algorithms have been proposed in the literature [23, 52, 95–97]. In the last years, the MPEG group has been working in the standardisation of a new coding solution for the MVD format.

Although the MVD format includes depth information, it provides a more efficient representation than the multiview video, due to the reduced number of texture views. Moreover, the depth maps usually convey less information than texture views, being encoded often using less than 20% of the texture video bitrate [98]. The main drawback of the MVD format is the computational complexity added to the decoder, that needs to synthesise the missing texture views from the few transmitted ones and depth maps.

Regarding the displaying side, the autostereoscopic displays or holographic systems can be used to present the MVD contents, as previously explained for the multiview video format, providing free-navigation and motion parallax effect. When compared to the multiview video format, MVD provides higher flexibility and easier adaptation to different 3D displays. This is because, in MVD most views need to be synthesised at the decoder side, which allows to generate any desired number of views. Moreover, other parameters, such as the baseline distance between the views, can be configured at the display side. These features of MVD format contribute to decouple the creation of 3D contents from the display technology.

Other related 3D representation formats that use geometry information based on depth maps are the video+depth and the layered depth video (LVD) [81, 99]. The video+depth format is a particular case of MVD format, which only uses one texture view and the respective depth map. This format is specified by ISO/IEC 23002-3, also referred to as MPEG-C Part 3 [100]. Since it only uses one texture view, this format presents backward compatibility with legacy 2D systems. It allows also to generate multiple views due to the depth information. However, this format has a limited rendering range, since it can not properly handle occlusions. In fact, the MVD format provides a higher level of interactivity and enables a larger navigation range, due to the larger number of provided texture views and depth maps. Regarding the LVD format, its principle relies on using multiple layers for different objects in one view, transmitting the colour and depth information for each layer as distinct views. The advantage of this format is the ability to represent colour and depth information of occluded areas, becoming less error prone than video+depth format in the presence of occlusions.

Holoscopic video

The 3D holoscopic imaging is a type of light field technique, which allows to record light intensity and direction information of a captured 3D object. This technique was firstly proposed in [101] with the designation of integral photography. Basically, a 3D holoscopic system comprises a regularly spaced 2D array of small spherical micro-lenses, known as a “fly’s eye” lens array, which can be used in both acquisition and display of the 3D holoscopic content, as shown in Figure 3.13.

In the acquisition side, a single aperture camera with a 2D array of micro-lenses is used to obtain the 3D holoscopic image, as shown in Figure 3.13a. Each micro-lens works as an individual small low resolution camera conveying a particular perspective of the 3D object at a slightly different angle. As a result, the planar intensity distribution representing the 3D holoscopic image comprises a two-dimensional (2D) array of micro-images, as illustrated in Figure 3.14. In the display side, the 3D holoscopic imaging allows 3D viewing experience without requiring glasses, presenting motion parallax and different views with no discomfort.

The compression of the holoscopic images illustrated in Figure 3.14, as well as holoscopic video sequences, can be done using a typical 2D video encoder. However, due to the small angular disparity between neighbouring micro-lenses, the holoscopic images present a significant spatial redundancy between neighbouring micro-images, that is not efficiently exploited by existing image and video coding standards. Several

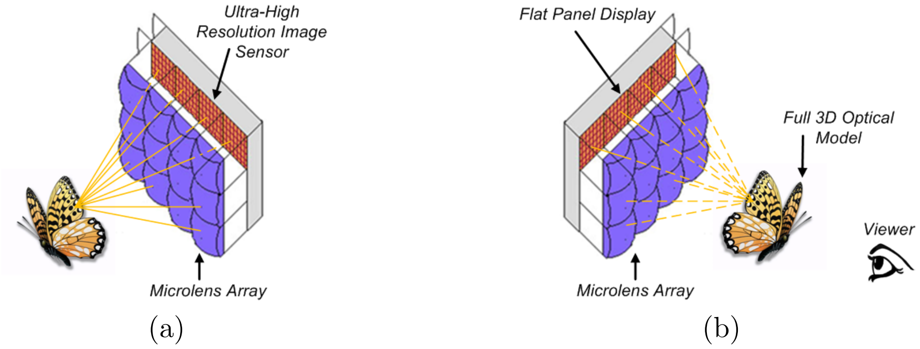


Figure 3.13: A 3D holoscopic imaging system [102]: (a) acquisition side; (b) display side.

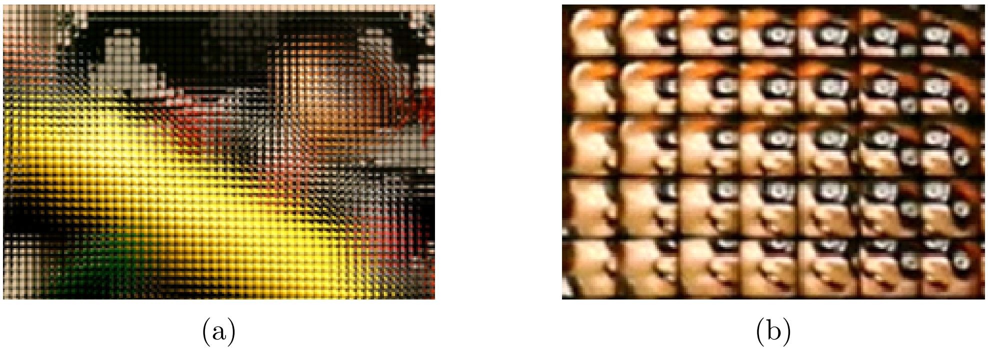


Figure 3.14: Holoscopic image captured with a 250 μm pitch micro-lens array: (a) full image with resolution of 1920×1088; (b) enlargement of 196×140 pixels showing the micro-images.

holoscopic image coding schemes have been proposed in the literature to better exploit the particular structure of these images.

In [103], a hybrid compression scheme, which combines two-dimensional Discrete Wavelet Transform (2D-DWT) and two-dimensional Discrete Cosine Transform (2D-DCT), is presented. In this scheme, the 2D-DWT is applied to each individual micro-image and, then, the 2D-DCT is applied on sets of 2D-DWT coefficients from neighbouring micro-images. The resulting 2D-DCT coefficients are quantised and entropy coded. Similarly in [104], the authors propose to decompose the 3D holoscopic image into several viewpoint images by extracting one pixel (from each micro-image) which had the same relative position inside the micro-image. A 3D-DWT is then applied to a stack of these viewpoint images. The lower frequency bands are assembled and compressed using a forward two-levels 2D-DWT followed by arithmetic encoding. In [105], the authors present a comparison between the performance of JPEG2000 Part 10 and H.264/AVC standards for encoding sets of micro-images and viewpoint images. These sets of micro-images (and also viewpoint images) are put together along the third dimension (or temporal dimension) to be encoded using the JPEG2000 Part 10 and H.264/AVC.

Another approach for efficient holoscopic image coding consists in using the existing video coding standards with an improved predictive framework. In this context, the Self-Similarity (SS) compensated prediction [106, 107] has been introduced into the H.264/AVC and HEVC encoders to efficiently handle the 3D holoscopic images. The SS works similarly to the IBC method introduced in HEVC SCC, which consists of BMA using an intra-frame search window. Such procedure allows SS to compensate the aforementioned non-local spatial correlation between the neighbouring micro-images.

3.3.2 3D video coding standards

As previously discussed, three-dimensional systems require the transmission of a larger amount of data than the traditional 2D video systems. In order to cope with the new requirements for 3D video compression, the Joint Collaborative Team on 3D video coding extension development (JCT-3V), formed by elements from MPEG (ISO/IECJTC1/SC29/WG11 3DV ad hoc group) and ITU-T (SG 16 WP 3), has been working on several 3D extensions for the current state-of-the-art HEVC standard. The MV-HEVC standard is the most straightforward HEVC-based solution proposed for the compression of multiview video contents. Similarly to the MVC (multiview extension of H.264/AVC), the MV-HEVC uses Disparity-Compensated Prediction (DCP) to exploit inter-view dependencies.

DCP is implemented by modifying the high-level syntax of the HEVC standard. In practice, the reference picture list of the current view, which originally contains the temporal frames for motion-compensated prediction, is augmented with the inclusion of already encoded frames from other views at the same time instance. In this way, the block-matching-based search procedure may adaptively find the optimal predictor, either from temporal or inter-view references, for each encoded block. The design of DCP provides a straightforward solution to adapt the existing HEVC technology for efficient representation of multiview video, given that no change is required at block-level tools.

In the scope of the MVD format, which is based on texture and depth map views, the standardisation process comes out with three solutions using different base coding technologies. MVC+D [108] is proposed as a simple solution for sending texture views along with depth maps, which does not introduce any change to the algorithm used in MVC. All changes are related to high level syntax elements providing a way to signal the presence of depth views. A more advanced approach which provides higher compression efficiency, being backward compatible with AVC (allowing a fast and easy adoption in the market), is referred to as 3D-AVC [109]. The recently finalised depth enhanced 3D video coding extension of the HEVC standard, known

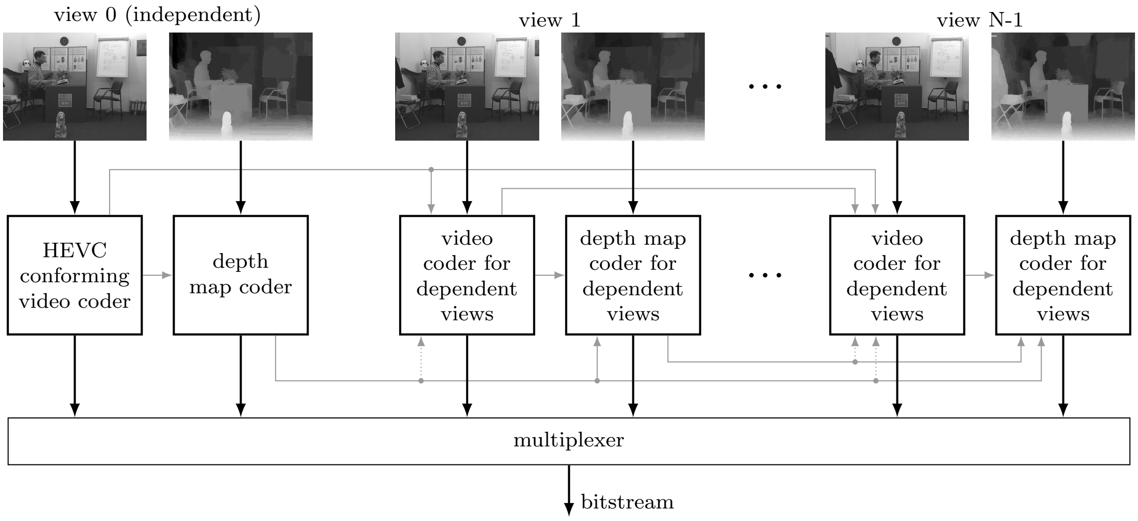

view 0 (independent)
view 1
view N-1
...
HEVC conforming video coder
depth map coder
video coder for dependent views
depth map coder for dependent views
...
video coder for dependent views
depth map coder for dependent views
multiplexer
bitstream

Figure 3.15: Main structure of 3D-HEVC encoder with inter-view and inter-component prediction represented by gray arrows. Figure taken from [23].

as 3D-HEVC [23, 75, 110, 111], is the state-of-the-art solution for compression of the MVD data. 3D-HEVC may also encode other 3D video formats, including the stereo video and the multiview video formats. In order to provide such format scalability, 3D-HEVC encodes each texture and depth map view based on HEVC technology. The dependencies between different views are exploited, in addition to the spatial and temporal ones encoded by HEVC basis algorithm. 3D-HEVC is also able to perform joint coding of the texture video and depth map components, allowing to exploit a new level of redundancy that arises with MVD format, namely the inter-component redundancy existing between the texture video and depth map signals.

Figure 3.15 illustrates the high-level structure of 3D-HEVC encoder, representing inter-view and inter-component prediction dependencies by gray arrows. The texture view 0 (independent view) is encoded using the reference HEVC standard, providing backward compatibility with 2D video systems, while other texture and depth map views (dependent views) use modified versions of HEVC which are able to exploit dependencies between views and between texture and depth components, as illustrated in Figure 3.15. These modified HEVC versions for dependent view coding include additional coding tools, which are described in what follows.

Dependent view coding tools

Similarly to MV-HEVC standard, the 3D-HEVC uses DCP to exploit inter-view dependencies for both texture video and depth map components. For an improved inter-view coding, 3D-HEVC also introduces modifications at the block-level tools, which allow to exploit the correlations of motion and residue information between multiple views. The most relevant tools include the Neighbouring Block-based Disparity Vector Derivation (NBDV), inter-view motion prediction, inter-view residual prediction and illumination compensation [75].

The main idea of the NBDV tool is to implicitly derive the disparity vector of the block, based on its spatial and temporal neighbourhood, in order to be used by other methods, namely the inter-view motion prediction and inter-view residual prediction. NBDV uses spatial candidates similar to the ones used in Advanced Motion Vector Prediction (AMVP) or merge modes of HEVC, plus a temporal neighbouring candidate. The first disparity vector found among the spatial and temporal candidates is returned by NBDV. When does not exist a disparity vector in the available candidates, a zero disparity vector is assumed. Note that, this approach does not require any additional signalling symbol in the bitstream.

Regarding inter-view motion prediction, its main purpose is to include additional candidates into the list of the merge mode (see Subsection 2.3.2), which can use up to six candidates instead of the five candidates defined in HEVC standard. Two possible additional candidates are the disparity vector (and associated reference picture index) derived by the NBDV method, and the motion vector corresponding to the block pointed out by the NBDV vector.

The inter-view residual prediction is used to exploit similarities between the motion-compensated residue generated by two distinct views. Its procedure consists in predicting the motion-compensated residual signal of the predicted block using an already encoded view, namely the motion-compensated residual signal associated to the block that is pointed out by the NBDV disparity vector and the corresponding reference picture index.

The illumination compensation is used to compensate luminance variations often observed between views, for instance when the cameras have calibration differences. For this reason, this tool is only applied on blocks predicted from inter-view reference pictures. The illumination compensation is achieved by using a simple linear model, whose coefficients are implicitly estimated by solving a least-squares problem, based on the reconstructed samples in the block neighbourhood.

Other tools used to improve the compression of dependent texture video views using information of previously encoded depth maps from other views, include the Depth-oriented Neighbouring Block-based Disparity Vector (DoNBDV), the View Synthesis Prediction (VSP) and the Depth-Based Block Partitioning (DBBP) [75].

Depth map coding tools

In order to better represent depth maps and preserve the sharp edges, 3D-HEVC proposes additional tools for depth map intra and inter coding, as well as for inter-component prediction. Moreover, some tools inherited from MV-HEVC are modified or disabled when compressing depth maps. The modified tools include the motion and disparity compensation methods. Instead of quarter-sample accuracy, the full-sample accuracy is used, because the eight-tap interpolation filters tend to create artifacts at sharp edges. Similarly, the estimated vectors are encoded using full-sample accuracy. The disabled tools in 3D-HEVC for depth map coding include the in-loop filters, specifically the DBF and the SAO. The inter-view motion and inter-view residual prediction tools presented above are also disabled for depth map coding. In order to exploit inter-component dependencies, 3D-HEVC proposes some modifications to the dependent coding techniques mentioned in Subsection 3.3.2 and includes new coding tools as well.

Regarding intra coding methods for depth maps, 3D-HEVC maintains the directional prediction modes and transform-based residue coding present in the reference HEVC standard. However, new tools that better preserve depth map discontinuities have been introduced [75], namely the Depth Modelling Modes (DMMs), segment-wise DC coding (SDC) and single depth intra mode.

DMM takes an important role for depth map representation in 3D-HEVC standard. It consists on new intra prediction modes that partition the depth block into two non-rectangular regions, which are approximated by constant values. The partitioning information and the mean value of both regions are the unique information required by this model. DMM may exploit some inter-component dependencies by deriving the depth block partitioning information from the co-located block in the corresponding texture video view. Similarly to directional modes, the residual signal calculated between the DMM model and the original depth map might be encoded using transform coding.

The SDC mode is a residual coding method alternative to the transform coding, which can only be employed on PUs of size $2N \times 2N$. For directional intra prediction, one segment is defined, while for DMM, two segments are considered. For each segment, SDC transmits the mean value of the residual signal using a procedure based on a Depth Lookup Table (DLT). DLT provides a residual DC value mapping that allows to reduce the residue bit depth, especially when original depth maps do not use the whole range of 256 depth values, e.g. quantised depth maps.

The single depth intra mode is another tool used for efficient representation of smooth areas in depth maps. It aims to approximate the depth block based on a single depth sample value that is derived from a sample candidate list. Specific sample positions in the current block neighbourhood are used to derive the sample candidate list. Whenever this mode is used no residual information is transmitted.

In relation to depth map inter coding, several techniques are introduced in 3D-HEVC. The Motion Parameter Inheritance (MPI) is used to predict depth maps based on the motion characteristics of the associated texture video view already encoded. The main idea behind this technique is to exploit similar motion characteristics between video and depth map signals. This is possible because the texture video and corresponding depth map signal capture the same scene from the same viewpoint at the same time instants. MPI adds a new candidate into the merge list of the depth block being encoded, which is derived from the motion information of the co-located texture block.

Inter predicted blocks in depth maps may also skip transform and quantisation methods. This is done using an inter-mode extension of the SDC method presented above, which approximates the residual signal using a constant DC value. Another source of redundancy resides in the block partitioning tree of both depth and texture blocks. 3D-HEVC exploits block partition similarities by predicting the depth map quadtree from the associated texture block quadtree for inter slices.

Since depth maps are not directly visualised, being instead used for view synthesis at the decoder side, the 3D-HEVC significantly improves the coding efficiency of depth maps using a modified version of the distortion metric used in Lagrangian cost function, known as view synthesis optimisation (VSO) [75]. The idea of VSO is to account the errors that encoded depth map blocks cause in the synthesised views, instead of considering the coding errors in the encoded depth map itself. The 3D-HEVC encoder test model allows to use two implementations of VSO: the Synthesised View Distortion Change (SVDC) and a model based synthesised distortion estimation. The latter avoids the rendering procedure, for a low complexity distortion calculation [75].

3.3.3 Experimental results

In this section we evaluate the RD performance of 3D-HEVC standard and compare it with the MV-HEVC standard, for the compression of MVD contents. The Shark test sequence, proposed in 3D-HEVC common test conditions (CTC) document [112], has been selected for the experiments. It includes three texture video views and the associated depth maps.

The multiview video coding with depth data scenario, based on the three view case, has been used in these experiments. As indicated by the common test conditions for 3D video core experiments [112], we used the encoder configuration files and rendering configuration provided with the 3D-HEVC reference software, specifically the HTM-13.2 [113]. In terms of temporal prediction structure, a GOP of 8, and intra period of 24, were used. The used texture QP values were: 40, 35, 30 and 25, and the corresponding QP values for depth maps were set to 45, 42, 39 and 34, as recommended by CTC.

In order to illustrate the importance of a modified distortion metric for depth map coding, a second configuration of 3D-HEVC replacing VSO method by the traditional SSE for depth map distortion (3D-HEVC_noVSO) was tested and evaluated. Although MV-HEVC was not designed for depth map coding it can be used to encode multiview depth data, considering depth maps as multiview video. In the performed tests, we encoded the Shark test sequence with MV-HEVC standard for comparison purposes.

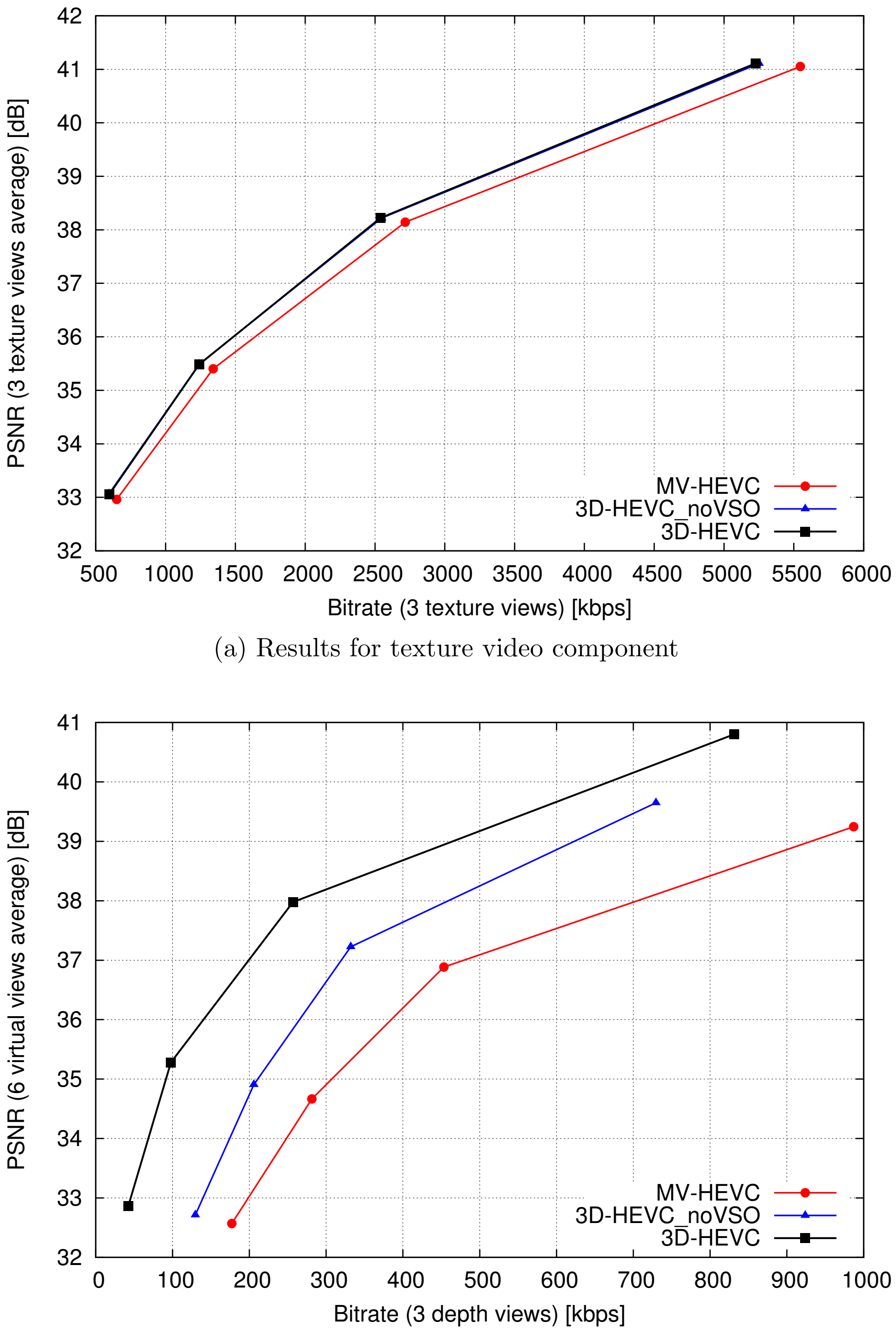
The evaluation procedure recommended by MPEG CTC has been followed. However, we independently plotted the results for the depth maps and texture views. For texture video, the average luminance PSNR of the three encoded views is presented as a function of the total bitrate used to compress the texture video views. For depth maps, the average PSNR of six virtual views, synthesised using the encoded depth maps and texture views, is presented as a function of the total bitrate used to compress the three depth map views.

Figure 3.16 presents the RD results for the three tested configurations: the default 3D-HEVC based on CTC, the 3D-HEVC without VSO and the MV-HEVC. From the RD results of texture video in Figure 3.16a, one may observe that 3D-HEVC is able to perform better in terms of compression efficiency than MV-HEVC standard. This is justified by the multiview block-level coding tools, which are only present in 3D-HEVC standard. Furthermore, these tools can make use of depth information from other views to better encode the texture video. Disabling the VSO method in 3D-HEVC does not result in a noticeable difference in the RD performance of the texture video, because this technique only affects the compression of depth maps.

Regarding the RD results for depth maps, shown in Figure 3.16b, they demonstrate the clear advantage of the 3D-HEVC encoder over the MV-HEVC standard for depth map coding. This advantage can be justified by the 3D-HEVC coding tools that were specifically targeted for depth map compression, such as the depth modelling modes, but also by the use of VSO method as distortion metric in the RD optimisation loop. The importance of VSO for depth map coding is clearly evidenced by the 3D-HEVC_noVSO curve of Figure 3.16b, which presents much lower performance than the full configuration of 3D-HEVC using VSO method.

It is also interesting to compare the bitrate used by the compressed texture video and depth map components. From Figure 3.16, it can be observed that depth maps can be encoded using approximately five times less bitrate (or 20%) than that used for texture, for high quality RD points. At lower quality RD points, the ratio between depth and texture bitrate can be as low as 10%.

The presented results clearly demonstrate the limitations of the existing transform-based video coding methods (such as MV-HEVC) and the benefits of using new coding tools, as the ones proposed in 3D-HEVC, for the compression of



(a) Results for texture video component

(b) Results for depth map component

Figure 3.16: Rate-distortion performance of shark test sequence using the 3D-HEVC, 3D-HEVC without VSO and the MV-HEVC configurations: (a) average PSNR of texture views versus bitrate of texture views (b) average PSNR of synthesised views versus bitrate of depth maps.

depth maps. These observations have encouraged the research work of Chapter 5, where we propose an efficient predictive coding framework for depth map compression.

3.4 Conclusions

In this chapter we described the main image and video coding algorithms used in this thesis, where the proposed prediction methods were implemented and evaluated. Section 3.1 described some important aspects about video compression, specifically the hybrid video coding architecture and the rate-distortion optimisation procedure used in current video encoders.

Section 3.2 presented an overview of the two main 2D video coding algorithms used along this thesis: HEVC and MMP algorithms. An experimental evaluation of these algorithms for image compression was performed, comparing them with the H.264/AVC standard. The results confirmed the superiority of both MMP and HEVC over the H.264/AVC algorithm. However, the MMP algorithm presented a worse rate-distortion performance than HEVC standard. These results motivated the first research topic of this thesis, that proposes an enhanced intra prediction framework for MMP algorithm (see Chapter 4). The HEVC standard was also evaluated for video compression. The presented results were consistent with the ones of image compression, showing that HEVC tends to perform better for high-resolution contents than low-resolution ones.

Section 3.3 presented a brief description of the main 3D video systems used in this thesis, focusing on the capturing, compression and displaying aspects of the 3D data processing chain. The described 3D video formats encompass the stereo video, multiview video, multiview video+depth and the holoscopic video. Regarding the 3D standardisation, the recent extensions of HEVC algorithm for 3D video coding, namely the MV-HEVC and the 3D-HEVC standards were briefly described. Some experimental tests comparing the 3D extensions of HEVC for the compression of multiview video+depth data demonstrated the advantage of 3D-HEVC over MV-HEVC standard. Such advantage is notorious specially for the depth map component, which is better compressed mainly due to the new coding tools introduced in 3D-HEVC for this type of content. These results emphasised the inefficiency of the original HEVC technology for depth map coding and demonstrated that significant improvements can be achieved using alternative coding methods for depth maps, being one of the motivations for the research work of Chapter 5.

Chapter 4

Efficient predictive coding using MMP

The experimental tests presented in Section 3.2.3, for the state-of-the-art image coding algorithms, shed some light in the limitations of the Multidimensional Multiscale Parser (MMP) algorithm in relation to the HEVC standard, mainly for the compression of high resolution images. This observation led us to the first research topic of this thesis, which aims to improve the intra predictive framework of the MMP algorithm.

In this chapter, we present several improvements to the prediction framework of MMP algorithm for efficient compression of image signals. Such improvements were implemented in the stereo image coding extension of MMP algorithm [114], which is briefly introduced in Section 4.1.

In Section 4.2, the proposed methods for the MMP algorithm are described. The directional prediction of MMP algorithm has been significantly improved, being extended with new prediction modes, based on the angular prediction modes used in HEVC. The use of a larger number of prediction modes tends to provide better prediction results and reduce the residual signal energy, consequently improving the overall RD performance of the pattern matching-based coding paradigm. Another significant enhancement to the MMP algorithm is the increased initial block size, which provides better prediction of high resolution images.

The experimental results of the proposed MMP algorithm are presented and discussed in Section 4.3, for both intra and stereo image coding. For the compression of stereo pairs, MMP independently encodes the left image based on intra coding methods, while the right image uses both intra and inter-view coding methods. The idea is to evaluate the influence of the proposed techniques in both the independently and dependently encoded images. Note that the dependent image exploits redundancies from the reference image that was previously encoded with the proposed intra prediction methods.

4.1 Introduction

Predictive coding methods have been already successfully combined with MMP algorithm as described in the literature [7, 115]. As presented in Subsection 3.2.2, the last version of MMP for intra image coding uses an intra predictive framework, which includes directional modes based on the ones of H.264/AVC standard, plus a least-squares prediction method [9]. However, the investigation of prediction methods for MMP algorithm has not been restricted to intra methods. Several inter prediction algorithms that exploit disparity information between stereo images [12, 114, 116], or dependencies between YUV colour components [10], have been also proposed in the last years for the MMP algorithm.

The original stereo version of the MMP algorithm, proposed in [114], is an extension to the intra-based MMP algorithm described in Subsection 3.2.2. Similarly to other stereo image encoders, MMP first encodes the reference image (usually assigned to the left view) using only intra coding methods, as described in Subsection 3.2.2. The dependent image (usually assigned to the right view) is then encoded using the same intra coding tools combined with additional disparity compensation methods that exploit the redundancy between images.

The MMP-based stereo image encoder in [114] uses up to three methods for disparity compensation. These methods include the well-known block-matching algorithm (BMA), for explicit disparity compensation, and two implicit methods based on template-matching (TM) [116] and least-squares prediction (LSP) [12]. The purpose of TM algorithm is to use the template area defined in the causal neighbourhood of the block in order to implicitly find the disparity vector, avoiding the transmission of the disparity vectors.

The LSP method for stereo image coding provides an efficient representation of disparity information, mainly when its representation in the block is more complex. In LSP approach, linear coefficients are adaptively estimated in a causal training window defined in the block neighbourhood. Such local training procedure is performed using several filter support shapes and the optimal one is explicitly signalled for each block to be predicted. More detailed information about this method has been provided in Section 2.4 and can be found in [12].

In this research work, we use a stereo implementation of MMP based on the one presented in [114], which enables the explicit BMA and the implicit LSP methods for disparity compensation. In comparison to the MMP version proposed in [114], BMA method has been improved by using better prediction of the disparity vectors and CABAC for entropy coding, while LSP received some improvements in its design, mostly related to the training procedure.

The combined usage of BMA and LSP methods allows MMP to achieve more efficient compression results for the right image. The LSP method enables disparity estimation without requiring the transmission of the disparity vectors, saving some bitrate. However, when the causal training area used by LSP is highly uncorrelated with the unknown block, MMP may select the explicit BMA and provide better prediction results, at the cost of a higher bitrate.

MMP evaluates the BMA and LSP methods in the rate-distortion loop, along with the intra prediction modes, for all available block sizes superior or equal to 4×4. The best prediction mode is selected according to a Lagrangian rate-distortion criterion. The MMP algorithm also employs a post-deblocking filter on both images of the stereo pair, based on the method proposed in [117].

4.2 Proposed contributions to MMP

The main contributions proposed in this work for MMP algorithm are related to the directional intra prediction method and the initial block size of MMP. The enhanced directional prediction method developed for the MMP algorithm is able to improve the compression of the left and right images (independent and dependent images, respectively), although its impact is more significant in the left image, where only intra prediction modes are allowed. The coding performance of the right image is also affected by the changes on the left image quality, because this image is used as reference by the disparity compensation methods.

The use of larger block sizes in the MMP algorithm is an efficient solution to increase the compression performance of high resolution images. In the proposed approach, the larger block size was not restricted for the intra prediction methods, but it was also enabled for the disparity compensation methods used to encode the right image. In the context of the increased initial block size proposed to the MMP algorithm, we also modified the block partitioning scheme, as described in the subsection that follows.

4.2.1 Flexible block partitioning with larger initial block size

In [7, 8, 114], MMP is used with a maximum block size of 16×16, as the H.264/AVC standard. However, with the advent of high resolution formats, the use of larger block sizes has shown to be beneficial, providing a better compression performance for high resolution images. Since MMP algorithm has been used for the compression of high resolution stereo pairs, we proposed to adapt it to use larger block sizes.

Initially, the MMP algorithm was adapted for the initial block size of 64×64. Keeping the MMP flexible block partitioning rule, the new initial block size enables a segmentation tree with 49 possible block sizes. However, an alternative block partitioning scheme was developed in this work.

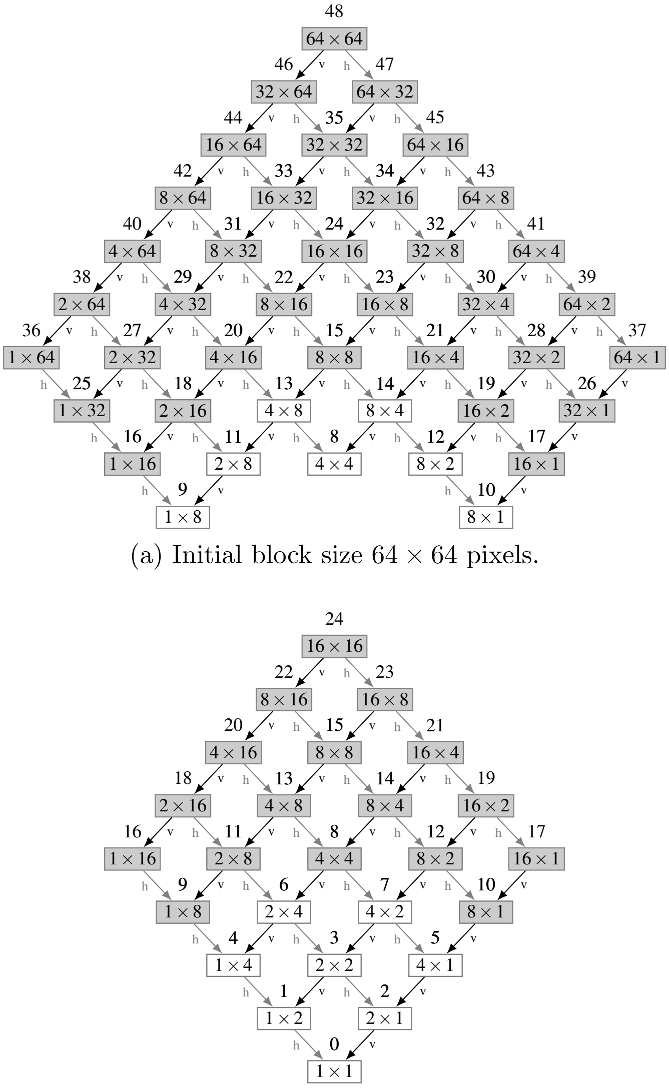
Empirical observations demonstrated that larger block scales improve MMP rate-distortion performance mainly at lower bitrates, while smaller block sizes continue to be frequently used at higher bitrates or in image regions with more complex structures or textures. In order to avoid the transmission of a large number of segmentation flags to signal the use of smaller block sizes, a bi-level approach for the initial block size is proposed. The idea of this approach is to enable the partitioning of the initial 64×64 block size into 16 square blocks with 16×16 size, which are independently optimised and encoded. MMP encoder decides the best initial block level by optimising both the coding of the 64×64 block size and the 16 smaller 16×16 blocks. One binary flag is transmitted to indicate the best approach.

Different segmentation trees were defined for the two initial block levels. While the 16×16 size uses the same segmentation tree as the one of the original MMP algorithm, a new tree was defined for the 64×64 size. Figure 4.1 illustrates the two possible block segmentation trees that can be used for MMP optimisation procedure. Note that the recursive partitioning of 64×64 blocks does not reach the smaller block sizes (below 4×4), because those block sizes are mainly used in the case wherein the initial block size is 16×16. This restriction allows to save some computational complexity for the initial block size of 64×64. While the partitioning trees of Figure 4.1 represent all the possible block sizes for residue coding, the use of prediction methods is restricted to a subset of these block sizes, being represented in Figure 4.1 by the gray blocks.

4.2.2 Improved intra prediction framework

Based on the recent developments of HEVC standard, an improved intra-prediction framework was proposed for the MMP algorithm. These improvements were highly motivated by the availability of larger block sizes, where new directional correlations may be exploited. Thus, the 8 directional modes used in the original MMP algorithm were updated to 33 directional modes, similar to the ones used in HEVC standard, as illustrated in Figure 4.2. Besides the directional modes, MMP uses the planar, DC and an intra-based least-squares prediction (LSP) mode [9], totalling 36 intra modes.

By using a more advanced intra prediction framework, the residue probability distribution of MMP algorithm tends to be narrower, leading to better encoding results. As demonstrated in [7], this happens because more uniform elements tend to be used at larger block sizes, favouring the adaptation of the MMP dictionary and its efficiency in representing the encoded blocks.



(a) Initial block size 64 × 64 pixels.

(b) Initial block size 16 × 16 pixels.

Figure 4.1: Block segmentation tree for the bi-level initial block size: (a) 64 × 64 and (b) 16 × 16.

In contrast to HEVC, which only uses square blocks for intra prediction, the MMP algorithm uses a wide range of block sizes, including ones corresponding to non-square blocks, as previously discussed. In order to employ directional prediction in MMP algorithm, we extended this method for non-square blocks and made some optimisations regarding the MMP algorithm. One of these optimisations is related to the available directional modes at each block size.

In some block sizes, *e.g.* very narrow blocks, not all the directional prediction modes are advantageous. For instance, two neighbouring modes may produce similar prediction results for some block sizes, despite having slightly different prediction directions. For this reason, we disabled some directional modes that seem to be irrelevant for certain block sizes. This enables a more efficient coding of the available prediction modes, and avoids unnecessary computations in determining similar

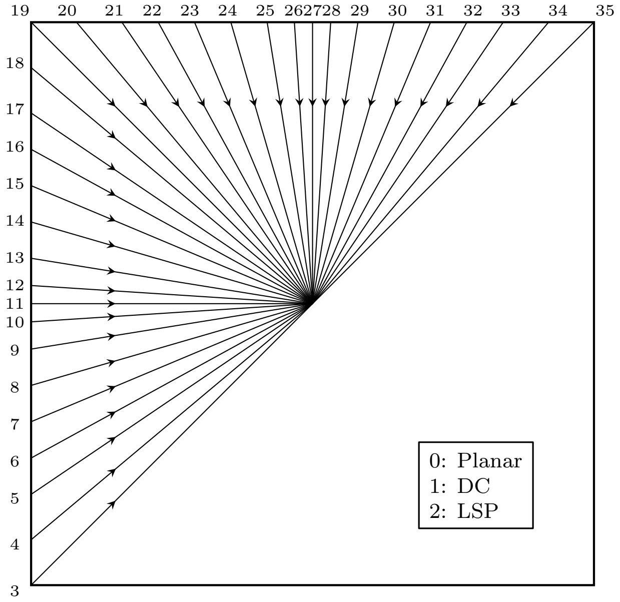


Figure 4.2: Set of intra prediction modes proposed for MMP.

Table 4.1: Prediction modes available for each block size.

Block size $h \times w$	**Used directional modes**
$h \geq 8$ and $w \geq 8$	all modes
$h \geq 8$ and $w = 4$	all except modes 4, 6, 8, 10 . . . 18
$h = 4$ and $w \geq 8$	all except modes 20, 22, 24, 26 . . . 34
$h = 4$ and $w = 4$	odd modes
$h < 4$ and $w \geq 4$	modes 3, 11, 19, 27 and 35
$h \geq 4$ and $w < 4$	modes 3, 11, 19, 27 and 35
$h < 4$ and $w < 4$	no mode

prediction blocks repeatedly. The proposed configuration of the available prediction modes for each block size present in MMP algorithm is shown in Table 4.1. Note that, $h \times w$ represents the block height versus block width, and the prediction mode indexes correspond to the ones used in Figure 4.2.

4.2.3 Coding of Prediction Modes

The proposed intra prediction framework uses 36 modes, which should be efficiently transmitted to the decoder. This can be done by exploiting eventual correlations that may occur between adjacent blocks of the image, using similar directional prediction modes. In this work, we propose a prediction process for the intra prediction modes in the MMP algorithm, in order to maximise the entropy coding performance of these modes.

The idea of the proposed scheme is to obtain 3 candidate modes, implicitly derived in both encoder and decoder, and to transmit the index associated to the candidate that matches the best prediction mode if it exists, otherwise the mode is explicitly transmitted. The candidate modes are derived from the causal reconstructed pixels associated to the left column and top row pixels adjacent to the unknown block. For each causal reconstructed pixel, the associated prediction mode is recorded. The proposed solution chooses the candidate mode based on the 3 most frequent prediction modes that were recorded from the causal reconstructed pixels. If the number of recorded modes is inferior to 3, the existing modes are used as candidates.

For mode encoding, a binary flag is transmitted indicating whether the prediction mode matches one of the candidates or not. Depending on the value of this flag, the following transmitted symbol is the candidate index (with 3 possible values) or actual prediction mode index (with 36 minus 3 possible values). The proposed prediction process provides an efficient mechanism for the compression of the prediction modes, being advantageous when the same mode is used on adjacent blocks.

In addition to the mode prediction process, we developed a way to further improve the entropy coding of prediction modes using the Adaptive Arithmetic Coding of the MMP algorithm [68]. Since some prediction modes may not be available, due to the previous presented restrictions in Table 4.1, the AAC of MMP only considers the available prediction modes in the initialisation process, using an uniform statistical distribution for these modes, and zeroing the probability of the unavailable modes.

During the compression process further modes can become temporarily unavailable, due to unavailable causal reconstructed pixels or due to an adaptive mode pruning method. We have improved the entropy coding of the chosen prediction mode by temporarily zeroing the probability distribution of those unavailable modes. This results in a higher coding efficiency because the total number of available modes is reduced and the probability of the transmitted mode is increased, requiring less bits to be represented. It is important to note that the availability of the causal reconstructed pixels can be checked on both the encoder and decoder, without requiring the transmission of any lateral information.

In order to further improve the efficiency of the MMP algorithm, we propose an adaptive mode pruning method that disables redundant modes based on the smoothness of the causal reconstructed pixels in neighbourhood of the unknown block. The motivation for this method comes from the fact that several directional modes produce exactly the same prediction results when the causal reconstructed pixels are constant, *i.e.* they have the same value.

We consider five causal neighbouring regions, namely the left-down, left, top-left, top and top-right regions, as previously illustrated in Figure 2.4 of Subsection 2.3.1. In the proposed method, we define a neighbouring region as constant whenever all

Table 4.2: Sets of prediction modes and associated neighbouring regions.

Set	Neighbouring regions	Prediction modes
1	top & left & top-left	modes 11 to 27, planar
2	left & left-down	modes 3 to 10
3	top & top-right	modes 28 to 35

of its pixels have the same value. Depending on the neighbouring regions declared as constant, several sets of directional modes can be disabled. This is particularly important in the MMP algorithm, mainly at lower bitrates, because homogeneous regions occur more frequently, as consequence of the rough block approximations generated by the uniform elements of the initial dictionary.

The proposed method defines 3 sets of prediction modes, as shown in Table 4.2, which were formed based on the neighbouring regions used in the propagation process of the directional modes. One set of prediction modes is disabled whenever all the neighbouring regions associated to that set are unavailable (*e.g.* block is adjacent to the image margins) or are constant (the pixels have all the same value).

This process is repeated on the decoder side, so that the statistical distribution of the modes is the same in both encoder and decoder sides, guaranteeing the success of the decoding process. Note that no additional information needs to be transmitted. Although this scheme introduces some computational overhead, due to the inspection of the smoothness of neighbouring pixels, it can also save some time on the encoder side, since modes associated to unavailable (or homogeneous) neighbouring regions do not need to be tested.

4.3 Experimental results

Experimental tests were performed in order to evaluate the RD performance of MMP algorithm using the proposed intra predictive framework and increased initial block size. In the first set of experiments, we evaluate the RD performance of the improved MMP algorithm for intra image coding, comparing it with the previous version of MMP [8] and the state-of-the-art transform-based image encoders. The idea is to evaluate the performance of the proposed coding techniques in the MMP algorithm.

A second experimental evaluation is presented for the stereo extension of the MMP algorithm using the proposed intra coding methods (referred to as MMP-stereo), being compared to the state-of-the-art stereo image coding algorithms. The used encoders were MVC, *i.e.* the multiview extension of H.264/AVC, using the stereo high profile (reference software JM-18.5) [5, 71, 85] and the MV-HEVC algorithm [23, 87] using the stereo configuration (reference software HTM-11.2) [112].

The used test data set comprises 11 stereo pairs, represented in Section A.1, which are generated using the first frame of two views selected from several multiview test sequences. Most of the stereo pairs come from multiview sequences proposed in the common test conditions for 3D video core experiments [112], namely the ones from Figure A.4 to Figure A.11. The stereo pairs from Figure A.1 to Figure A.6 have 1024 × 768 pixels, while the remaining ones have 1920 × 1088 pixels. Only the luminance component was considered in these experiments.

The MMP algorithm has been evaluated from low to high bitrates, running experiments with different input λ values (used in the MMP Lagrangian RD-cost function): 300, 75, 25 and 10. It is known that the overall encoding performance of a stereo pair can be optimised by playing with bitrate allocation between the left and right images, usually through the use of different QP or lambda values for each image. Since this research work is not focused on stereo coding aspects, efficient bitrate allocation for stereo coding was not addressed. Thus, the same lambda values were used for compressing both the left and right images of the stereo pair using MMP algorithm. Regarding the transform-based standards for stereo image coding (MVC and MV-HEVC), the same QP values were used for the two images of the stereo pair, specifically: 25, 30, 35 and 40.

4.3.1 Evaluation of the proposed intra coding methods for MMP

We evaluate the overall coding advantage of the proposed methods for intra coding by comparing the new improved MMP algorithm (using the proposed methods) with its previous version in [8]. In order to isolate the intra coding methods from the disparity compensation framework, only the results for the left image of the stereo pairs are evaluated in this subsection. Note that the left image is the first encoded one, thus the compression algorithms can only use intra coding methods for this image. We compare the intra coding performance of the MMP algorithm with the one of the MVC and MV-HEVC standards. Since only the left image is encoded using intra coding methods, these algorithms will be also referred to as MMP-intra, H.264/AVC and HEVC encoders in this subsection.

Table 4.3 presents the Bjontegaard Delta PSNR (BDPSNR) results [70] of the proposed MMP algorithm relative to the previous version of MMP [8], H.264/AVC and HEVC standards, for the compression of the left image of the tested stereo pairs. Positive values of BDPSNR correspond to the average PSNR gain of the proposed MMP algorithm relative to the compared method.

The results of Table 4.3 show that the proposed intra coding methods, specif-

Table 4.3: Bjontegaard Delta PSNR results of the proposed MMP algorithm relative to the previous MMP, H.264/AVC and HEVC encoders for the compression of the left image of the used test stereo pairs.

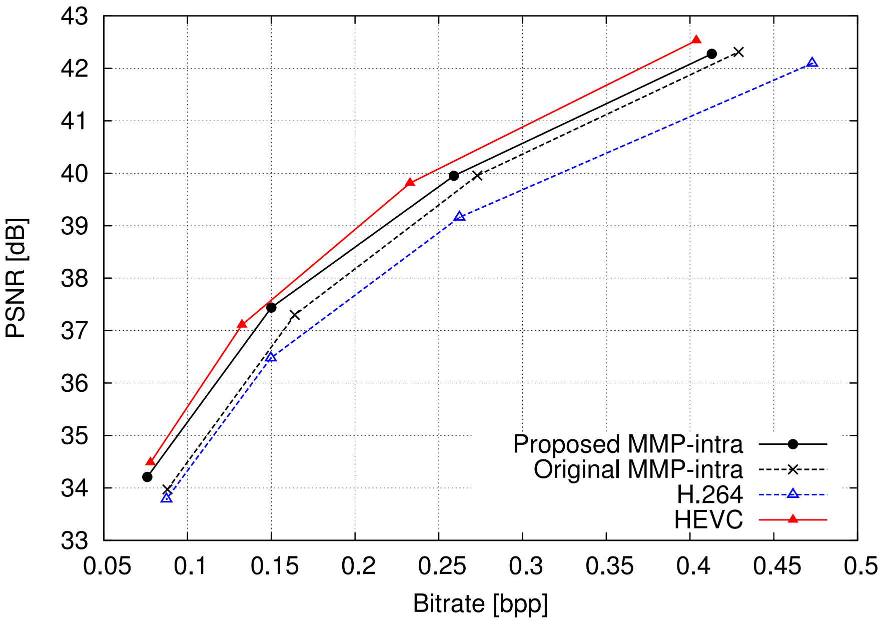
Image	**MMP**	**H.264**	**HEVC**
Ballet	0.855	1.263	0.127
Breakdancers	0.301	0.519	-0.054
Book Arrival	0.447	0.944	-0.133
Balloons	0.527	1.049	-0.334
Kendo	0.651	0.965	-0.492
Newspaper	0.484	0.931	-0.293
Poznan Street	0.484	0.952	-0.181
Poznan Hall2	0.754	0.919	0.063
GT Fly	0.491	1.179	0.322
Undo Dancer	0.299	0.978	0.209
Shark	0.430	1.019	-0.159
Average	**0.520**	**0.974**	**-0.083**

ically the advanced directional prediction and the new block partitioning scheme, improve the MMP performance for all images, presenting BDPSNR gains that range from, approximately, 0.3 dB up to 0.85 dB. When compared with H.264/AVC, the proposed MMP-intra algorithm presents BDPSNR gains close to 1 dB for most images.

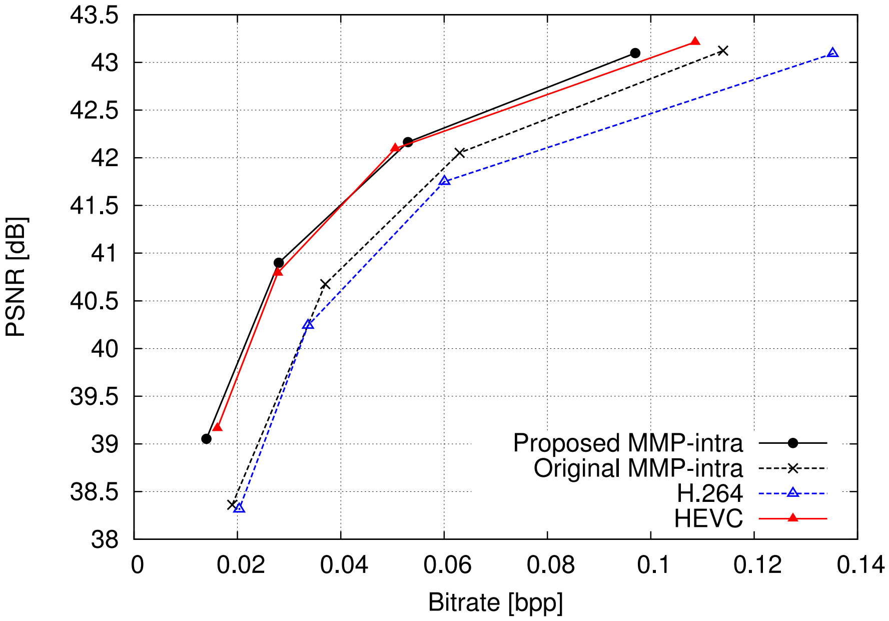
From the last column of Table 4.3, one may observe that the proposed MMP algorithm presents quite competitive results with the state-of-the-art HEVC standard. In the best case, MMP obtains a BDPSNR gain superior to 0.3 dB, specifically for the left image of GT Fly, while the worst case corresponds to a loss of almost 0.5 dB for the left image of Kendo stereo pair. The average results of MMP-intra are very close to the ones of HEVC, presenting a slightly lower performance, that traduces to a BDPSNR loss inferior to 0.1 dB.

It is interesting to observe that the proposed MMP outperforms the HEVC standard for some images, specially for the synthetic images GT Fly and Undo Dancer. These results support the fact that MMP tends to perform better than transform-based encoders for non-natural or synthetic images, as previously reported in literature, for text documents [8] and depth maps [52]. Although the results for Shark image, which is also synthetic, are not as good as the ones of the other synthetic images, the performance difference between MMP and HEVC for this image is small, as can be seen from Table 4.3.

To complement the previous discussed Bjontegaard results, Figure 4.3 presents the RD curves of two representative left images, namely the Newspaper and the Poznan Hall2, which illustrate the performance of the encoding methods from low to high bitrates. The results shown in Figure 4.3 reveal that the proposed MMP



(a) Newspaper



(b) Poznan Hall2

Figure 4.3: RD performance of MMP using the proposed intra coding methods (labelled as Proposed MMP-intra), previous version of MMP in [8] (labelled as Original MMP-intra), H.264/AVC and HEVC, for the compression of the left image of two selected stereo pairs.

algorithm tends to improve the original version of the MMP algorithm, mainly at lower bitrates. Such gains at lower bitrates can be justified by the increased usage of larger block sizes at these bitrates, providing more efficient representation using less bits. Furthermore, as the previous version of the MMP algorithm in [8] tends to provide a superior coding performance at higher bitrates, the improvements of the

Table 4.4: Bjontegaard Delta PSNR results of the proposed MMP-stereo algorithm relative to the MVC and MV-HEVC encoders for the compression of the used test stereo pairs.

Stereo pairs	**MVC**		**MV-HEVC**	
	Left	Right	Left	Right
Ballet	1.096	1.405	-0.093	0.099
Breakdancers	0.383	0.758	-0.162	-0.009
Book Arrival	0.764	1.255	-0.173	0.051
Balloons	0.947	1.734	-0.423	-0.144
Kendo	1.089	1.767	-0.415	-0.034
Newspaper	0.841	1.534	-0.263	-0.055
Poznan Street	0.612	1.283	-0.250	-0.021
Poznan Hall2	0.814	1.249	-0.017	0.129
GT Fly	1.121	1.269	0.283	0.266
Undo Dancer	0.922	1.045	0.227	0.237
Shark	0.927	1.210	-0.237	-0.026
Average	**0.8650**	**1.3189**	**-0.1385**	**0.0448**

proposed MMP algorithm at these bitrates are more restricted. Such superiority of the MMP algorithm from [8] at higher bitrates can be observed in Figure 4.3, by comparing it with the H.264/AVC standard.

Even though the proposed methods improve the MMP algorithm, results for the Newspaper image are still slightly inferior to the ones obtained for the HEVC standard. However, in some cases, such as Poznan Hall2, the proposed MMP algorithm stands out with state-of-the-art coding performance for most RD points.

4.3.2 Evaluation of the large block size MMP for stereo image coding

The experimental tests for stereo image coding were performed using the stereo version of the MMP algorithm (referred to as MMP-stereo) improved with the proposed prediction framework and increased block sizes, as well as the MVC and MV-HEVC encoders. In the case of MVC, the default configuration file *encoder_stereo.cfg* provided in JM-18.5 reference software was used. For MV-HEVC, the default configuration file for multiview video coding based on two views has been used. Both MV-HEVC and MVC configurations used a disparity search range equal to 96, and the *SearchMode* parameter, associated to the disparity compensation mode, was set to *Full search*, as done in MMP-stereo algorithm. In order to generate the Bjontegaard results and RD plots for stereo image coding experiments, the luminance PSNR of each image and the sum of the bitrate used to encode both images of the stereo pair were used.

The summary of the BDPSNR results for the proposed MMP-stereo, relative to the MVC and MV-HEVC standards, is presented in Table 4.4, for all the tested stereo images. Note that the BDPSNR values for the left image differ from the ones presented in Table 4.3, simply because the total bitrate (of both left and right images) was considered in the BDPSNR calculations.

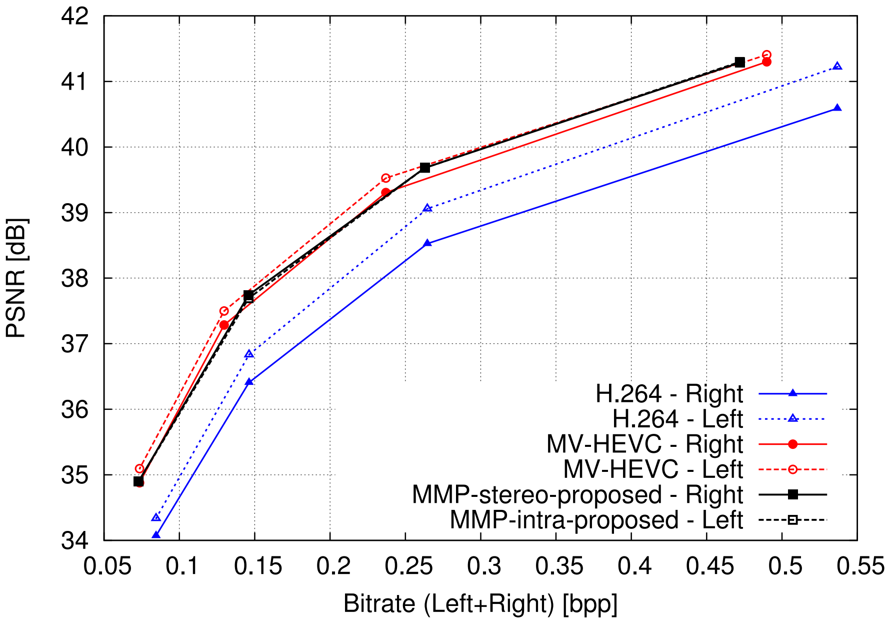
The results of Table 4.4 clearly show the advantage of the proposed MMP-stereo algorithm over the MVC standard, presenting relevant RD gains for all stereo pairs. The BDPSNR gains of MMP-stereo range approximately from 0.38 up to 1.12 dB for the left image, and from 0.76 up to 1.73 dB for the right image.

In regard to the MV-HEVC algorithm, Table 4.4 shows that MMP-stereo achieves very competitive results with the state-of-the-art encoder. Although it presents a slightly average loss of about 0.14 dB for the left image, the MMP-stereo is able to outperform MV-HEVC for the right image in some stereo pairs. These results demonstrate the capabilities of the proposed MMP algorithm for the compression of stereo pairs, being able to achieve state-of-the-art RD results, in particular for synthetic stereo pairs.

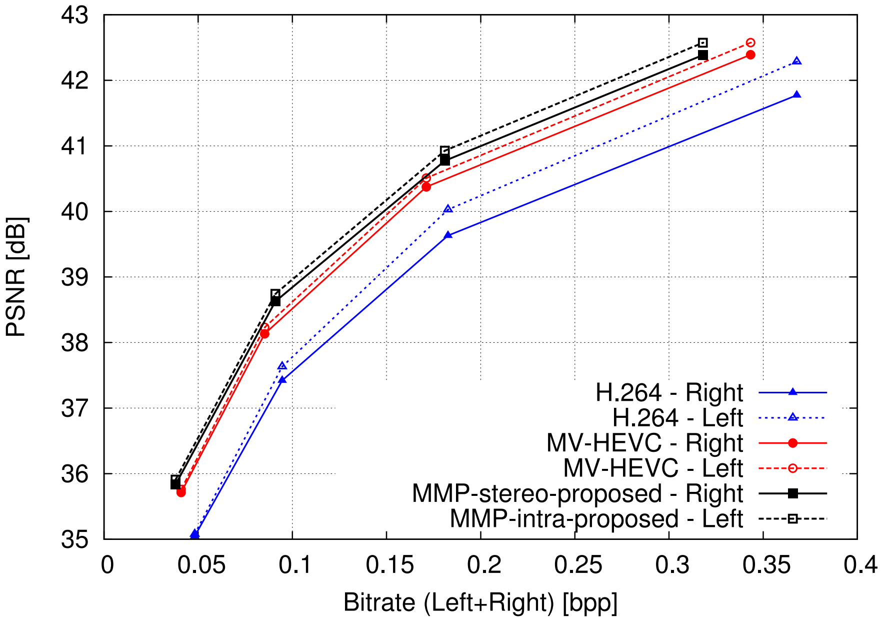
The RD curves presented in Figure 4.4 compare the MMP, MVC and MV-HEVC algorithms, for two selected stereo pairs, namely the *Book Arrival* and *GT Fly*. These results show that the performance of both the MMP-stereo and MV-HEVC algorithm is significantly higher than the one of MVC, for all bitrates. When compared to MV-HEVC standard, the MMP-stereo demonstrates a competitive RD performance for the Book Arrival stereo pair. The results for GT-Fly demonstrate that MMP-stereo can reach new compression benchmarks for high resolution stereo pairs, providing better performance than MV-HEVC standard from low to high bitrates.

The presented results for stereo image coding are in line with the ones previously presented for MMP-intra, demonstrating the advantage of the proposed intra prediction methods and increased block size for the MMP algorithm in both intra and stereo coding scenarios. These results show that the proposed MMP algorithm is worth pursuing, clearly outperforming the MVC standard, based on H.264/AVC technology, and presenting a competitive performance with the one of the state-of-the-art transform-based MV-HEVC for stereo image coding.

The major issue that prevents the practical use of MMP algorithm is the high computational complexity that characterises this kind of pattern-matching-based encoders. This problem was aggravated in this work by increasing the number of intra prediction modes and using a larger initial block size that significantly increased the block partitioning combinations. Due to these changes, MMP performs the pattern-matching-based coding of a larger number of residue blocks, which results in a considerably higher computational complexity. One solution to mitigate this



(a) Book Arrival



(b) GT Fly

Figure 4.4: RD performance of the proposed MMP-stereo, MVC and MV-HEVC for the compression of two selected stereo pairs.

problem is to use faster sub-optimal methods for selection of the prediction mode. The high recursive block partitioning scheme of MMP could also be combined with quad-tree block partitioning and use a restricted number of possible block sizes within each quadtree level.

4.4 Conclusion

In this chapter we have presented an improved intra prediction framework for the MMP algorithm using an increased number of block sizes. In Section 4.1 we briefly described the stereo implementation of MMP algorithm used in this research work as basis for the proposed compression framework. The weaknesses of the MMP algorithm for the compression of high resolution contents were the main motivation for this work.

Section 4.2 described the proposed techniques and improvements for the MMP algorithm. This includes the use of a larger initial block size and an improved intra prediction framework, which extends the number of available directional modes, based on the HEVC angular prediction modes. Several improvements related to the entropy coding of intra prediction modes and the block partitioning scheme were also presented.

The experimental results presented in Section 4.3 showed the importance of the improved intra prediction framework and increased block size for the MMP algorithm, for both intra and stereo image coding applications. The presented RD results demonstrated that the proposed MMP algorithm can be an efficient alternative to the transform-based coding standards for the compression of high resolution images, specially synthetic contents. When compared to the MVC standard, the proposed MMP-stereo encoder achieved an average BDPSNR result of 0.87 dB and 1.32 dB for the left and right images of a stereo pair, respectively. In relation to the MV-HEVC standard, the average BDPSNR results of MMP-stereo present a small loss of 0.14 dB for the left image and a slight gain of 0.04 dB for the right image.

Chapter 5

Compression of depth maps using predictive coding

The research on efficient algorithms for compression of depth map signals has been increasing in the last years, with the advent of the multiview video+depth (MVD) format for 3D video representation. As previously described in Subsection 3.3.2, the 3D-HEVC is the most recent standard for the compression of MVD contents, addressing the problem of depth map coding with new techniques specifically designed for this type of signals.

In this chapter we present an alternative depth map coding solution, which is mostly based on intra predictive techniques. The developed algorithm, referred to as Predictive Depth Coding (PDC), combines a flexible block partitioning scheme, based on the one of MMP algorithm, with an intra predictive framework based on directional prediction. To improve the representation of depth edges that cannot be efficiently predicted by the directional modes, we propose the constrained depth modelling mode, which uses explicit edge representation. For residue coding a simple and low complexity approach, based on constant and linear residue modelling, is proposed.

Section 5.1 presents an overview of the most important depth map intra coding techniques proposed in literature, including the state-of-the-art methods adopted in the recent 3D-HEVC standard for depth map coding. In Section 5.2, we present the first investigation work on predictive depth map coding, which resulted in a preliminary version of the proposed PDC encoder. Some experiments for the two-view image coding scenario are presented and discussed. The overall structure of the main proposed PDC encoder is presented in Section 5.3. A detailed description of the coding techniques used in PDC is given in Section 5.4 and Section 5.5 presents the encoder RD control mechanism used in PDC algorithm. In Section 5.6, we present and discuss the experimental results, demonstrating the advantage of PDC algorithm over the state-of-the-art 3D-HEVC standard for intra depth map coding.

5.1 Overview of intra techniques for depth map coding

Depth maps consist of grayscale images that capture the 3D structure of the scene, by representing the relative distance of the objects to the camera (with brighter pixels corresponding to closer points). In general, depth maps are characterised by approximately constant regions, separated by sharp edges at object boundaries. Due to these characteristics, current transform-based video coding standards, such as H.264/AVC or HEVC, present some issues for coding depth map signals. Most of these issues are caused by transform coding and quantisation that generate a limited number of coefficients in flat areas and introduce strong ringing artefacts, which blur sharp edges. As the depth maps are typically used to generate intermediate views, these artefacts highly affect rendering capabilities. These observations have motivated the investigation of alternative coding techniques for depth maps.

In the last years, alternative coding methods have been proposed for efficient depth map compression [23, 52, 95–97, 118–120]. The mesh-based depth map coding has been proposed in [97], by using an adaptive binary triangular tree (tritree). The main problem of this method is related to the placement of triangular patches on a regular grid, which usually causes the generation of a large number of small patches along the edges. The JPEG2000 standard has also been used for depth map coding by assigning a Region Of Interest (ROI) to each object in the depth map [118]. In spite of avoiding some of the artefacts of the transform-based algorithms, this method is not efficient when many objects exist in the scene. Also, the use of MMP algorithm for depth map coding was successfully demonstrated in [52]. Nevertheless, MMP requires a high computational complexity in both the encoder and decoder sides.

Another interesting method is the Platelet algorithm [95, 96] that approximates the blocks resulting from a quadtree segmentation by using different piecewise-linear modelling functions. Smooth blocks are approximated by using a constant or a linear modelling function. Blocks with depth discontinuities are modelled by a Wedgelet function, defined by two piecewise-constant functions, or by a Platelet function, defined by two piecewise-linear functions, both separated by a straight line. The optimal quadtree block sizes and modelling functions are chosen according to a cost function, that evaluates the rate and distortion. Bitrate savings of about 25% using Platelet algorithm for depth map coding have been reported in comparison to H.264/AVC intra coding [95].

Algorithms based on Wedgelet modelling have been also successfully used for efficient depth map representation [119, 121]. The use of Wedgelet block partitioning has been early investigated during HEVC standardisation, presenting bitrate reductions on depth map coding up to 11% [119, 121]. Currently, Wedgelets are used as part of the state-of-the-art depth intra coding techniques adopted in 3D-HEVC standard [23, 111]. A recent proposal on intra depth map coding proposes to replace the transform-based residue coding and directional intra prediction framework of 3D-HEVC standard by an advanced geometry-based intra prediction approach which includes plane fitting, Wedgelet modelling, inter-component prediction and constant offset residual coding [120]. Reported results showed a RD performance close to the one of the 3D-HEVC encoder. An interesting feature of this algorithm is the ability to provide a triangular mesh that represents the scene surface.

The 3D-HEVC algorithm is the most recent standard developed for the MVD compression, presenting the current state-of-the-art performance for depth map coding. Regarding the intra techniques for depth map coding, 3D-HEVC algorithm employs transform coding with directional intra prediction, as typically used for texture image coding [23]. However, in order to better represent depth maps, 3D-HEVC introduces new coding tools and disables some of the existing ones. The disabled tools include the in-loop filters, namely DBF and SAO, which were designed for natural image coding. These filters are useless for depth map signals, and add unnecessary computational complexity.

In order to better represent the sharp edges of depth maps, 3D-HEVC has three main additional intra coding tools [75, 110]: Depth Modelling Modes (DMM), segment-wise DC coding (SDC) and single depth intra mode. Depth Lookup Table (DLT) was also proposed to reduce the bit depth of residual signal, for depth maps with reduced depth range due to quantisation. The View Synthesis Optimisation (VSO) consists in using the distortion of synthesised views to directly evaluate the effect of the encoded depth block error in view synthesis during the RD optimisation process. The following subsections present a brief description of the most important depth intra coding methods used in 3D-HEVC algorithm, as for the reference software HTM-13.2 [75].

5.1.1 Directional intra prediction

Intra prediction is a key tool of standard image and video compression algorithms, being equally important for depth map coding. For an efficient prediction of the sharp edges and flat areas present in depth maps, 3D-HEVC maintains the same intra prediction modes, as the ones proposed to the HEVC standard, which were previously described in Subsection 2.3.1. This prediction framework includes 35 modes defined for square prediction unit sizes, from 4 × 4 up to 32 × 32.

HEVC directional block prediction is based on 33 different angular modes, numbered from 2 to 34, as illustrated in Figure 2.5 of Subsection 2.3.1. This represents a significant extension to the 8 directional modes used in H.264/AVC encoder, which is mainly motivated by the increased size of prediction units. To generate the predicted block, the reconstructed block boundary samples are projected into some directions, using bi-linear interpolation with 1/32 sample accuracy.

Alternatively, intra planar and DC prediction modes can be used to predict smooth areas, which are frequent in depth maps. Intra planar assumes an amplitude surface with vertical and horizontal slopes, derived from the block neighbourhood reference samples, while intra DC uses a flat surface with a constant value estimated from the block neighbourhood.

5.1.2 Depth Modelling Modes

Depth modelling modes (DMM) consist of new intra prediction modes for efficient edge representation [23, 122]. These modes are available together with the original HEVC intra directional prediction modes, providing an alternative approximation for depth maps. The residual difference between the depth modelling approximation and the original depth map is encoded using transform coding, as for ordinary intra prediction modes, or explicitly modelled by using constant approximation values. The main idea of the depth modelling modes is to divide the block into two disjointed regions, and approximate them by using constant values. Two types of partitions are defined, namely Contours and Wedgelets.

In Wedgelet partition, a straight line defined between two points located on different borders of the block is used to separate the block into two regions, P_1 and P_2. This type of partition is illustrated in Figure 5.1 (left) using the straight line defined between points S and E. At the encoder side, the best matching Wedgelet partition is searched using the original depth signal. The Wedgelet which presents the lowest distortion, between the original depth and the pre-defined Wedgelet pattern, is transmitted. More details about Wedgelet partition search and signalling can be found at [75].

The Contour partition mode differs from Wedgelet partition in the sense that it is not a geometry guided block division, but texture guided block segmentation. This is an inter-component-predicted mode, which uses co-located texture block to generate block partitioning (see Figure 5.1 - right), based on a thresholding method. The threshold is estimated from the average value of the four corner luma samples of the co-located texture block. Depending whether the samples of the texture block are above or below the estimated threshold, they are classified has being part of region P_1 or P_2, resulting in a block partitioned into two disjointed regions that represents the Contour partitioning. Note that, unlike Wedgelet partitioning, each region of Contour mode may contain multiple parts, as illustrated in Figure 5.1

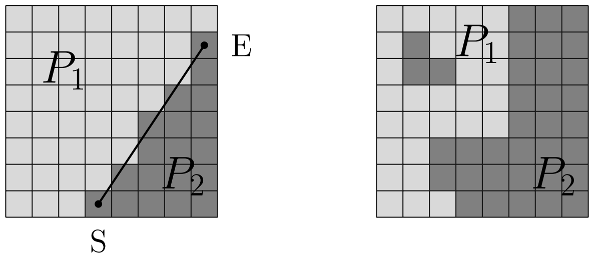


Figure 5.1: Example of Wedgelet (left) and Contour (right) block partitions.

(right) for region P_2. Because it uses the texture information to derive the partitioning pattern for the depth block, Contour mode provides an efficient solution to represent arbitrarily shaped partitions with no overhead.

Depth modelling modes also require the transmission of the depth model in each partition. Depth is modelled as a flat surface, characterised by a single Constant Partition Value (CPV). During encoding, CPVs are estimated as the mean depth value over each partition. For an even more efficient representation, estimated CPVs are predicted based on neighbouring reconstructed pixels. The difference between the estimated and predicted CPVs, denominated delta CPVs, can be transmitted using two different approaches. In the first one, delta CPVs are transformed, quantised and entropy coded, while the other approach signals CPVs using SDC and DLT methods, which are detailed below.

5.1.3 Depth Lookup Table

It has been observed that most depth maps were originally quantised, not using the full available depth range of 2^8 values. Depth Lookup Table [75, 123] was proposed to reduce depth residue bitrate using a restricted set of valid depth values. 3D-HEVC constructs the DLT based on the depth values used in each GOP of the input sequence, and transmits the DLT information to the decoder.

DLT algorithm uses some auxiliary mapping tables to map the valid depth values to index values and vice-versa. From the construction procedure, detailed in [75], three tables are derived, namely the DLT $D(.)$, the Index Lookup Table $I(.)$ and the Depth Mapping Table $M(.)$. In order to derive the residual index i_{resi} to be transmitted, the original depth, d_{orig}, and the predicted depth, d_{pred}, are converted to the respective indices and subtracted:

$$i_{\text{resi}} = I(d_{\text{orig}}) - I(d_{\text{pred}}). \tag{5.1}$$

At the decoder, the reconstructed mean depth value is firstly derived as

$$\hat{d}_{\text{orig}} = I^{-1}(I(d_{\text{pred}}) + i_{\text{resi}}), \tag{5.2}$$

and the mean residual signal is obtained using

$$\hat{d}_{\text{resi}} = \hat{d}_{\text{orig}} - d_{\text{pred}}. \tag{5.3}$$

The reconstructed samples $\hat{P}_{\text{x,y}}$ are computed by adding the mean residual value $\hat{d}_{\text{resi}}$ on each prediction sample $P_{\text{x,y}}$.

5.1.4 Segment-wise DC Coding

The Segment-wise DC Coding is an alternative residual coding method, which does not require transform and quantisation methods [75]. SDC can be applied to all depth intra prediction modes and it only can be used at Prediction Units (PUs) of size $2N \times 2N$. For directional intra prediction, one segment is defined, while for DMM two segments are defined. For each segment, SDC encodes the mean value of the residual signal using the DLT method.

5.1.5 Single Depth Intra Mode

The main motivation for single depth intra mode is the fact that most areas on depth maps are smooth and present similar depth values. The principle of this mode is to reconstruct the depth block by using a single depth sample value, which is obtained from a sample candidate list. Specific sample positions of current block neighbouring are used in a predefined order to derive the sample candidate list. No residue information is transmitted when this mode is used.

5.1.6 View Synthesis Optimisation

As pointed out previously, depth maps are not directly observed by the viewer. They carry on the structural information of the scene, being mainly used for view synthesis. In order to better encode depth maps for their purpose, the View Synthesis Optimisation (VSO) method is used in 3D-HEVC. The principle of VSO method is to measure the distortion of the synthesised views, instead of the Sum of Squared Errors (SSE) of the depth block [122]. This method is motivated by the fact that the coding errors in the depth maps lead to quite different distortions in the synthesised views. To measure the distortion of the synthesised view, two VSO metrics can be applied in the RD optimisation process.

The first VSO metric is known as Synthesised View Distortion Change (SVDC). SVDC estimates the change of distortion in the synthesised intermediate view caused by the change in the depth block due to the coding error. This process is defined as the distortion difference, ΔD, computed between two synthesised texture images, as illustrated in Figure 5.2.

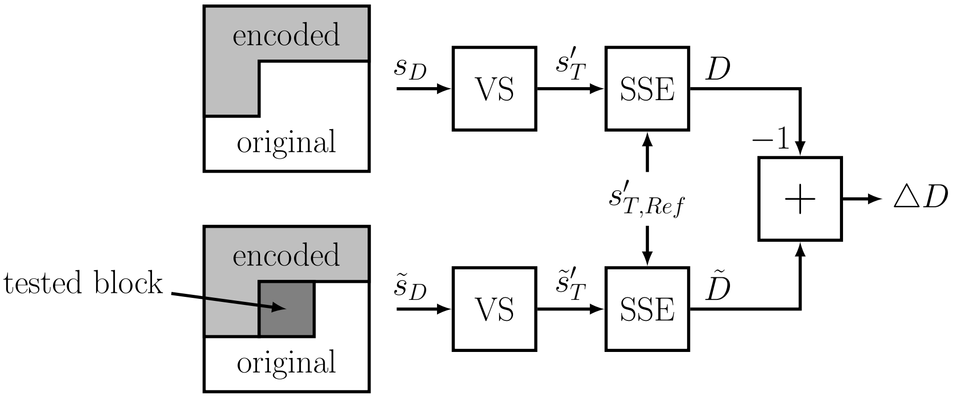


Figure 5.2: SVDC procedure for the distorted depth block identified in bottom depth map (labelled by tested block). Figure taken from [75].

The first synthesised view, s'_T, is generated through view synthesis (VS) procedure [79, 80] using a depth map that comprises the reconstructed depth values for the previously encoded blocks and the original depth values for the remaining depth map blocks. The second synthesised view, $\tilde{s}_T$, is generated using a similar depth map, that only differs in the samples of the block currently being encoded. Instead of using the original samples, the reconstructed samples produced by the coding mode under testing are used, as signalled in Figure 5.2 by the dark pattern. The SSE for each synthesised view is estimated against a reference synthesised texture view, $s'_{T,Ref}$, which is rendered using uncompressed depth and texture video data. The whole SVDC calculation can be represented by the following expression:

$$\Delta D = \tilde{D} - D = \sum_{(x,y)\in I} \left[\tilde{s}_T(x,y) - s'_{T,Ref}(x,y)\right]^2 - \sum_{(x,y)\in I} \left[s'_T(x,y) - s'_{T,Ref}(x,y)\right]^2 \tag{5.4}$$

where I represents the set of samples of the synthesised view. In order to avoid the rendering of the entire synthesised view for each tested block and save some computational complexity, 3D-HEVC uses an efficient implementation of SVDC that performs minimal rendering of the regions that are affected by the distorted depth block under test. Commonly, the SVDC implementation synthesises several intermediate views and the average view synthesis distortion is evaluated.

The second VSO metric used in 3D-HEVC estimates the synthesised view distortion without actually performing view rendering. It was designed to be fast, but still provides a reliable model for view synthesis distortion. Details about this metric are available at [75].

5.2 Investigation of efficient methods for depth map coding

Prior to the development of the main depth map coding algorithm proposed in this chapter, a preliminary investigation work on potential methods for depth map coding was initiated. This investigation has been initially motivated by the high performance results observed using MMP paradigm for depth map coding [52]. When this research was initiated, the standardisation process of 3D-HEVC encoder has not been started, yet. The Platelet algorithm [95, 96] was one of the most efficient approaches for depth map coding, being the main reference used to compare and evaluate the performance of the investigated algorithms.

In this research work, we started investigating the potential of the flexible block partitioning and intra predictive scheme of original MMP algorithm for depth map coding only, thus avoiding the high computational complexity associated to the residue coding method based on the pattern matching paradigm. Instead of using the pattern matching for residue coding as done in MMP, a simpler approach was investigated, based on linear block approximation combined with an efficient dictionary-based algorithm that reuses the previously encoded approximations.

The investigated algorithm resulted in an efficient codec for depth map coding, when compared with other algorithms proposed in literature. The directional prediction provides a good representation of sharp edges present in depth maps, while the piecewise-linear functions are mostly used to represent smooth regions. The use of piecewise-linear functions has been inspired on Platelet algorithm [95, 96]. However, the developed algorithm applies the piecewise-linear functions over the intra-predicted residual signal, instead of the original depth signal, as in [96].

Since depth maps are not directly observed by the users, the performance of depth map coding algorithms is often evaluated by analysing the quality of the synthesised virtual views using the compressed depth maps. Experimental tests demonstrated that the synthesised views generated using the depth maps encoded by the developed algorithm presented a higher PSNR quality than the ones synthesised using depth maps compressed by other contemporary algorithms, such as Platelet, MMP or H.264/AVC standard. When compared to the original MMP algorithm, which was used as the basis framework for this work, the preliminary proposed algorithm for depth map coding was able to achieve a superior RD performance with an inferior computational complexity.

The following subsections describe the main features of the preliminary version of the proposed depth map coding algorithm, namely the partitioning tree, the use of hierarchical intra prediction, the piecewise-linear residue approximation and rate-distortion optimisation procedure. Introductory experiments using the existing depth map coding algorithms, as well as the investigated method, under the two view depth map coding scenario are presented and discussed.

5.2.1 Flexible directional prediction using linear residue approximations

This algorithm first divides the input depth map into 32 × 32 pixel blocks. Each block can be divided either in horizontal or vertical direction, resulting in 32 × 16 or 16 × 32 blocks, which may in turn be further divided into 16 × 16 blocks. These blocks may be useful to approximate large smooth areas, commonly observed in depth maps. From 16 × 16 blocks, the same flexible partitioning scheme of original MMP algorithm is used. This results in three larger block sizes (32×32, 32×16 and 16×32) plus 25 block sizes with dimensions $2^m\times2^n$, for $m, n = 0, \ldots, 4$, obtained by recursive partitioning of the 16×16 blocks, either in horizontal or vertical directions.

The flexible dyadic partitioning has an important role in the representation of depth map edges, because very thin blocks may be used (*e.g.*, 8 × 1) to match the depth edges. For intra prediction, the developed algorithm uses nine modes, similar to those defined in H.264/AVC encoder. However, the neighbouring causal samples are not smoothed as done in the H.264/AVC.

Apart from the sharp edges, most regions of depth maps can be efficiently predicted, because they are constant (*e.g.*, areas inside an object) or vary smoothly (*e.g.*, on ground plane or walls). These depth map characteristics motivate the use of piecewise-linear fitting for residue coding. Thus, for a block S, of size $2^m\times2^n$, the linear approximation of each pixel $S(x,y)$, with block coordinates $x=[0,\ldots,2^m-1]$ and $y=[0,\ldots,2^n-1]$, is given by:

$$\hat{S}(x,y) = \alpha_0 + \alpha_1\tilde{x} + \alpha_2\tilde{y} \quad , \tag{5.5}$$

where $\tilde{x} = (x - 2^{m-1} + 1)$ and $\tilde{y} = (y - 2^{n-1} + 1)$. Note that $(\tilde{x},\tilde{y})$ is a displaced version of (x,y), with an approximately zero mean value. The $(\tilde{x},\tilde{y})$ values could be determined with exact zero mean by using fractional displacement values. However, fractional representation of $(\tilde{x},\tilde{y})$ is not favourable because the residue block values are integers. The advantage of using $(\tilde{x},\tilde{y})$ approximately centered around zero is that, in these conditions, the value of α_0 corresponds to the actual mean value of the target residue block. When the prediction step is successful, the values of the generated α_0 coefficients are highly correlated and centered around zero, which favours the entropy coding.

The linear model coefficients α_0, α_1 and α_2 are estimated in order to minimise the squared error between the original depth values $S(x,y)$ and the approximation

$\hat{S}(x,y)$. The approximation coefficients for each block are then entropy coded and transmitted to the decoder. In order to further reduce the overhead associated to the coefficient transmission, an improved residue coding scheme was developed, being presented in the following subsection, together with the adopted RD optimisation procedure.

5.2.2 Approximation model coding and RD optimisation

The developed depth map encoder takes the coding decisions that minimise a Lagrangian RD cost function. This cost is affected by many coding parameters, including the block partitioning decisions, the selected intra-prediction modes and the used piecewise-linear functions.

Typical image coding algorithms use a distortion metric based on the Sum of Squared Errors (SSE), in order to maximise the objective PSNR quality of the encoded image. This objective does not apply to depth maps because they are not directly observed by the users. Instead, the compressed depth maps should provide efficient synthesis results. In this work, we performed some experiences using alternative distortion metrics for RD optimisation, namely with the Sum of Absolute Errors (SAE), and analysed the results. The experimental tests revealed that using SAE as distortion metric provided more efficient PSNR results for the synthesised views than using the SSE-based distortion. For this reason, the developed encoder was setup to use SAE as distortion metric.

The linear model coefficients are quantised using a non uniform quantiser, that uses an adaptive quantisation step Q. The value of Q for each coefficient can be 1, 4, 8 or 13, depending on the interval in which the values of $|\alpha_0|$, $|\alpha_1.2^m/2|$ and $|\alpha_2.2^n/2|$ belong to: $[0,\ldots,10]$, $]10,\ldots,22]$, $]22,\ldots,86]$ and $]86,\ldots,255]$, respectively. Therefore, the quantised coefficients can be obtained by α_0/Q, $(\alpha_1.2^m)/(2Q)$ and $(\alpha_2.2^n)/(2Q)$.

In order to further improve the algorithm's compression efficiency, a dictionary-based method that reuses the previously encoded approximations was introduced. This method is inspired on the pattern matching paradigm used in the MMP algorithm. A dictionary is created for each block size and is initialised with an all-zero pixel block. During the encoding process, the dictionary is updated with the new encoded patterns.

Thus, each residue block can be approximated by using an explicit linear fitting model or by transmitting an index of the dictionary. The choice is made according to the solution that provides the minimum Lagrangian cost. For the case of an explicit linear model approximation, the J_{fit} cost function is given by:

$$J_{\text{fit}} = D_{\text{fit}} + \lambda \Big[R(\text{flag}_{\text{fit}}) + \sum_{j=0}^{2} R(\alpha_j) \Big], \tag{5.6}$$

Regarding the dictionary-based approach, the cost of each pattern of a dictionary with M elements, represented by index i, is computed as:

$$J_{\text{dic}}(i) = D_{\text{dic}}(i) + \lambda \big[R(\text{flag}_{\text{dic}}) + R(i) \big], \qquad i = 0, \ldots, M \tag{5.7}$$

In these equations, D_{fit} and D_{dic} are the distortions associated to each approximation, and $R(\text{flag}_{\text{fit}})$ and $R(\text{flag}_{\text{dic}})$ correspond to the bitrate of the flags used to indicate whether the residue is approximated by the linear function or by a dictionary pattern.

Each time a new linear approximation is explicitly transmitted, the encoded pattern is added to the dictionary, becoming available to approximate future residue blocks using less bitrate (dictionary index). This dictionary-based method provides an efficient way to reuse approximation patterns. It is important to note that the high computational complexity traditionally associated to the dictionary searching procedures has a reduced impact in this encoder. This fact results from the limited growth ratio of the dictionary in this encoder, which is much smaller than the one of the MMP algorithm, where many transformations are performed over each new element, to further enrich the dictionary [52].

The computational complexity of this method is mostly related to the exhaustive testing procedure used to find the optimal partitioning tree. In order to reduce the number of tested residual blocks, the proposed encoder selects the prediction mode solely based on the prediction error, avoiding the full RD optimisation of the residue block for all prediction modes.

5.2.3 Experimental results

The performance of the proposed depth map coding algorithm has been evaluated by analysing the quality of the synthesised texture views using the compressed depth maps. The first frame of six virtual views based on four test sequences commonly used in literature have been synthesised in these experiments: cameras 1 and 4 of *Ballet*[1] [124], cameras 1 and 4 of *Breakdancers*[1] [124], camera 9 of *Book Arrival* [125] and camera 40 of *Champagne Tower* [126]. In order to synthesise a virtual view associated to camera n, the compressed depth maps corresponding to cameras $n-1$ and $n+1$ (left and right cameras) and the uncompressed texture views (luminance

[1] http://research.microsoft.com/en-us/downloads/5e4675af-03f4-4b16-b3bc-a85c5bafb21d/

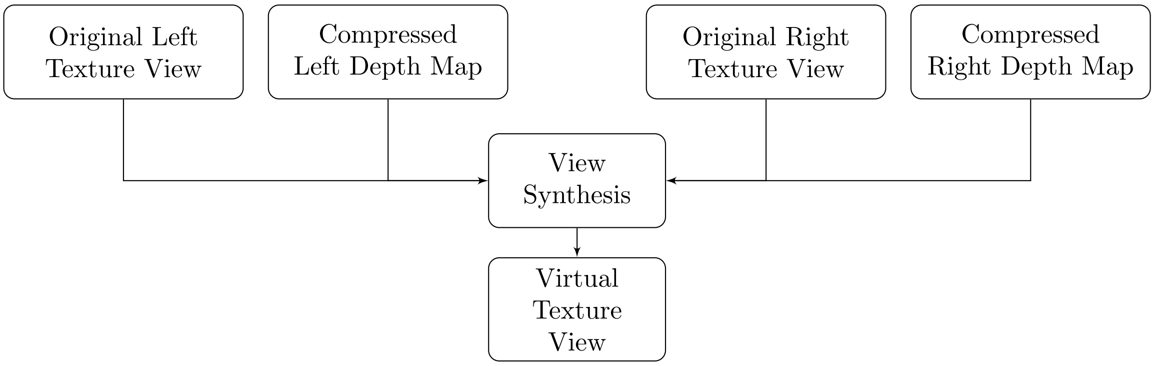

Original Left Texture View
Compressed Left Depth Map
Original Right Texture View
Compressed Right Depth Map
View Synthesis
Virtual Texture View

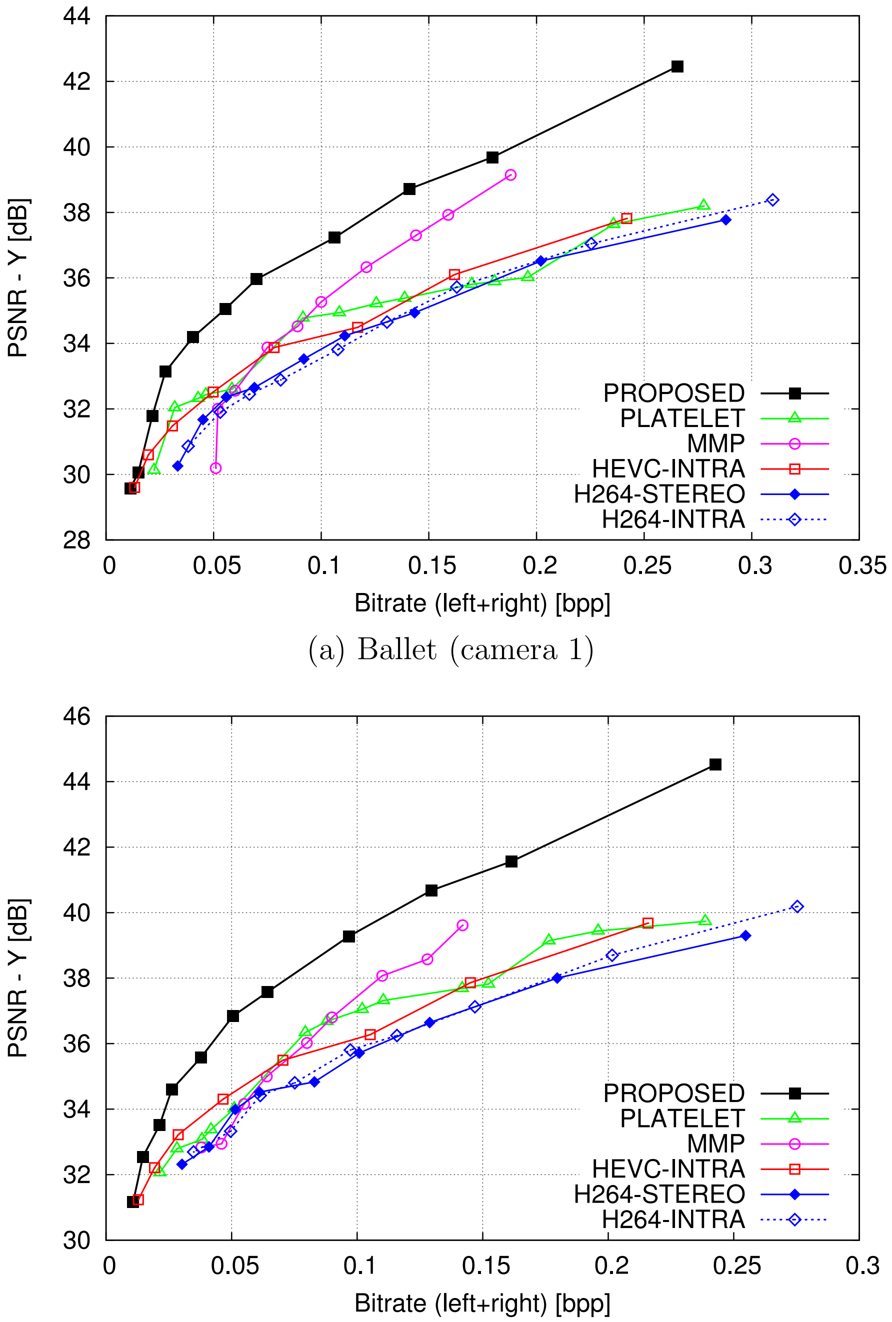
Figure 5.3: Evaluation framework for depth map coding using the two-view scenario.

signal) of the same cameras were used, as illustrated in Figure 5.3. The VSRS-3.5 software [127] was used to perform the view synthesis procedure based on DIBR method [80].

The method was compared against other algorithms with available source code or experimental results, namely the transform-based H.265/HEVC standard [1, 2] (reference software HM-9.1rc1 [72] using intra main configuration), the H.264/AVC standard (software JM-18.0 [71] using the intra and stereo high profiles) [5], the MMP algorithm [52] and the Platelet-based depth map encoder [95, 96]. The stereo high profile of H.264/AVC standard introduces a new level of dependency because it allows to exploit the inter-view redundancies between both encoded depth map views. Although the remaining algorithms do not exploit inter dependencies, this stereo configuration has been considered for the experimental comparison. For the case of Platelet algorithm, only the results for the sequences *Ballet* and *Breakdancers* were found available at [128]. It is important to note that the Platelet algorithm was the state-of-the-art approach when this research work was being developed.

Figures 5.4, 5.5 and 5.6 illustrate the rate-distortion performance of the proposed algorithm relative to the mentioned methods, for the first frame of the test set. The objective quality of the synthesised views is given by the PSNR of the luminance signal. In order to compute the PSNR, a reference synthesised view was firstly generated by using the uncompressed depth maps and texture views. The bitrate of the presented RD plots corresponds to the sum of the bitrate used by both, left and right, compressed depth maps.

These results show that the proposed method achieves the best performance for most bitrates and test sequences. When compared to the Platelet algorithm, for the available test sequences results, the developed algorithm presents a superior compression efficiency for all the bitrate values, gaining up to 4 dB for Ballet and up to 2 dB for Breakdancers, at higher bitrates. Similar observations can be drawn when the comparison is made with H.264/AVC standard, using intra or stereo high profiles, for all test sequences. Despite using the stereo high profile, depth maps



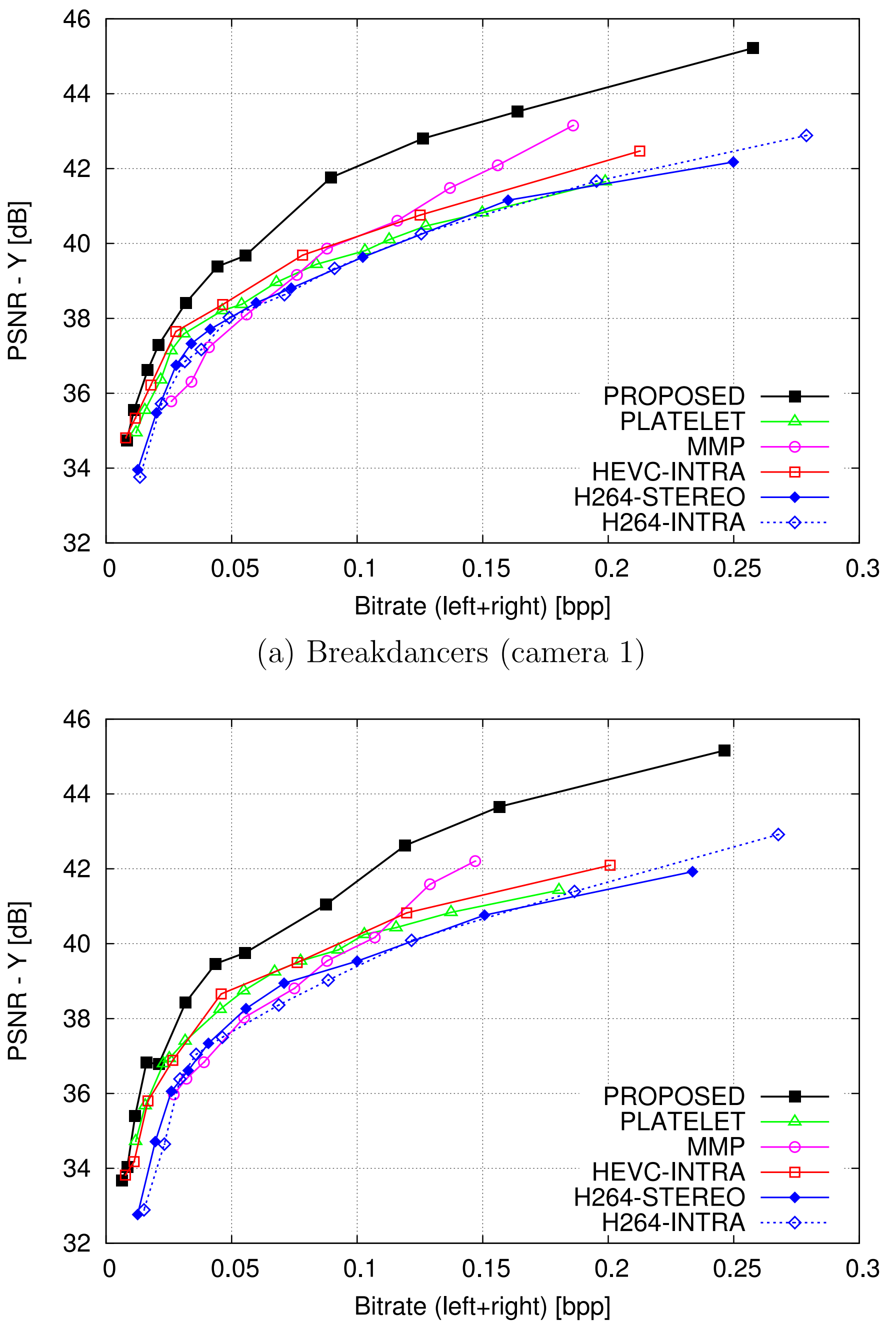
(a) Ballet (camera 1)

(b) Ballet (camera 4)

Figure 5.4: PSNR results for the synthesis of the first frame of Ballet sequence (cameras 1 and 4), as a function of the total bitrate used by left and right depth maps.

encoded by H.264/AVC are not accurate enough for view synthesis. These results can be justified by the kind of artefacts generated by the transform coding method.

The HEVC standard using intra main configuration for depth map coding presented interesting results at lower bitrates. Despite exhibiting results very close to the ones of the proposed method at lower bitrate, HEVC performance is still significantly inferior from medium to higher bitrate. The reasonable performance at

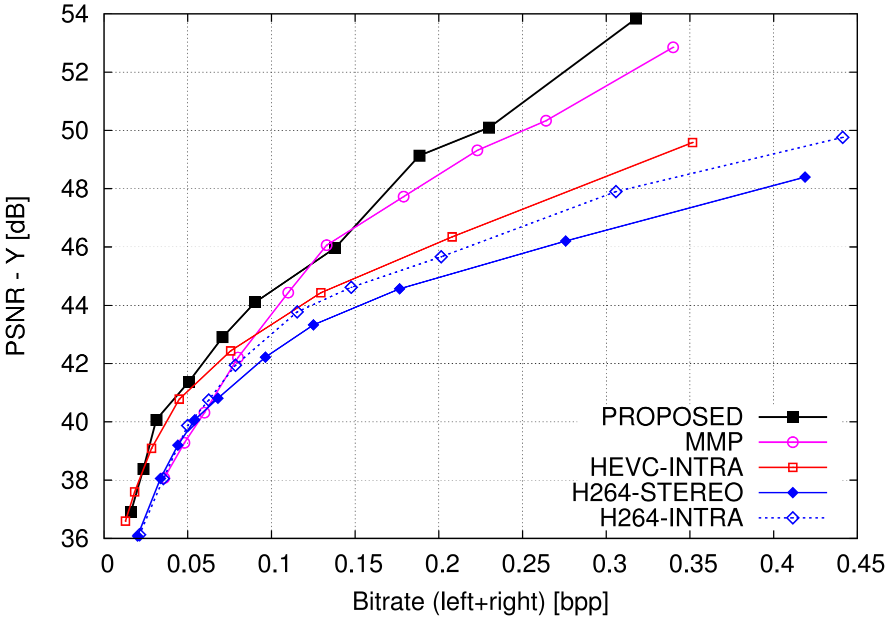


(a) Breakdancers (camera 1)

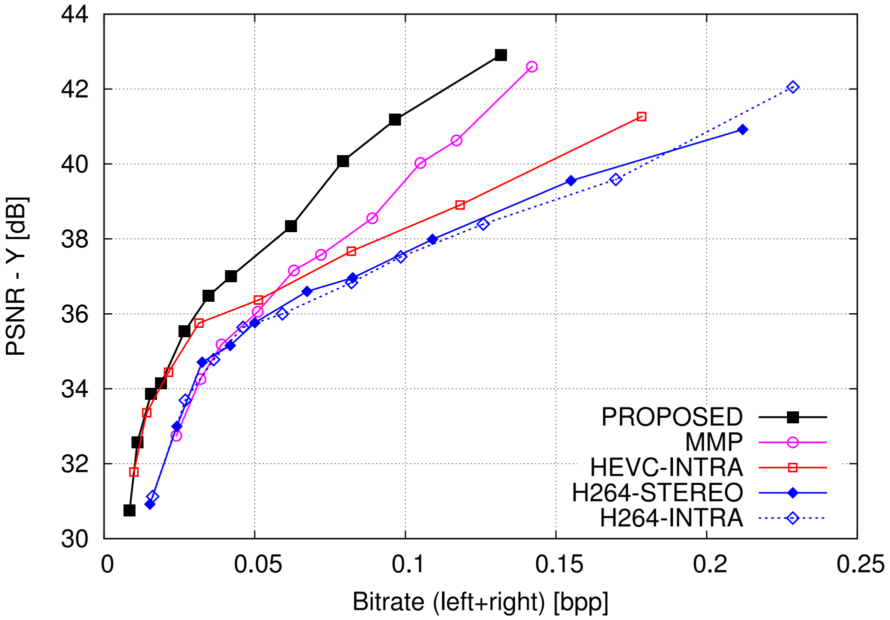
(b) Breakdancers (camera 4)

Figure 5.5: PSNR results for the synthesis of the first frame of Breakdancers sequence (cameras 1 and 4), as a function of the total bitrate used by left and right depth maps.

lower bitrate can be related to the maximum block size of 64 × 64 adopted in HEVC standard, which is superior to the maximum block size of 32 × 32 used in the proposed method. When compared to MMP, the proposed method also demonstrates to perform more efficiently for most RD points. It is interesting to observe that, at higher bitrates, MMP tends to perform better than the Platelet algorithm and the transform-based standards, as discussed in [52].



(a) Book Arrival (camera 9)



(b) Champagne Tower (camera 40)

Figure 5.6: PSNR results for the synthesis of the first frame of Book Arrival (camera 9) and Champagne Tower (camera 40) sequences, as a function of the total bitrate used by left and right depth maps.

In terms of computational complexity the proposed algorithm presents encoding times in the hundreds of seconds. Despite showing superior values than the encoding times of the transform-based algorithms, they are approximately 10 to 50 times less than the ones of MMP algorithm, which is the basis framework of the proposed method. The computational complexity of the proposed decoder is also much inferior than the one of MMP decoder, taking approximately one second to decode one frame.

5.3 Overview of Predictive Depth Coding

The investigation work described in the previous section resulted in a preliminary version of the proposed depth map coding algorithm, presenting an efficient coding performance when compared to other depth map coding algorithms presented in literature. The experiments demonstrated that directional prediction and flexible block partitioning, based on the MMP algorithm, is important for efficient representation of depth maps. The use of an alternative low complexity residue coding method, based on linear residue approximation, also revealed to be advantageous for depth map coding. These observations suggested further investigation on improved predictive methods that better exploit the flexible block partitioning scheme for edge representation and are able to reduce the encoder computational complexity.

With the advent of the 3D-HEVC proposal, new state-of-the-art results on depth map coding were reached, overcoming the existing coding approaches presented in literature, including the Platelet, the MMP, as well as the developed depth map coding algorithm, described in the previous section. In this context, based on the recent developments of 3D-HEVC standard, new coding techniques and improvements to the previous proposed algorithm were investigated, resulting in a new improved depth map coding algorithm. As the proposed algorithm is based on intra coding techniques, the coding of temporal, inter-view, and inter-component (between depth map and texture video) redundancies are not addressed in this work.

The block diagram of the novel proposed intra-based algorithm, denominated as Predictive Depth Coding (PDC), is presented in Figure 5.7. Similarly to most image/video coding schemes, PDC uses a hybrid coding approach based on intra

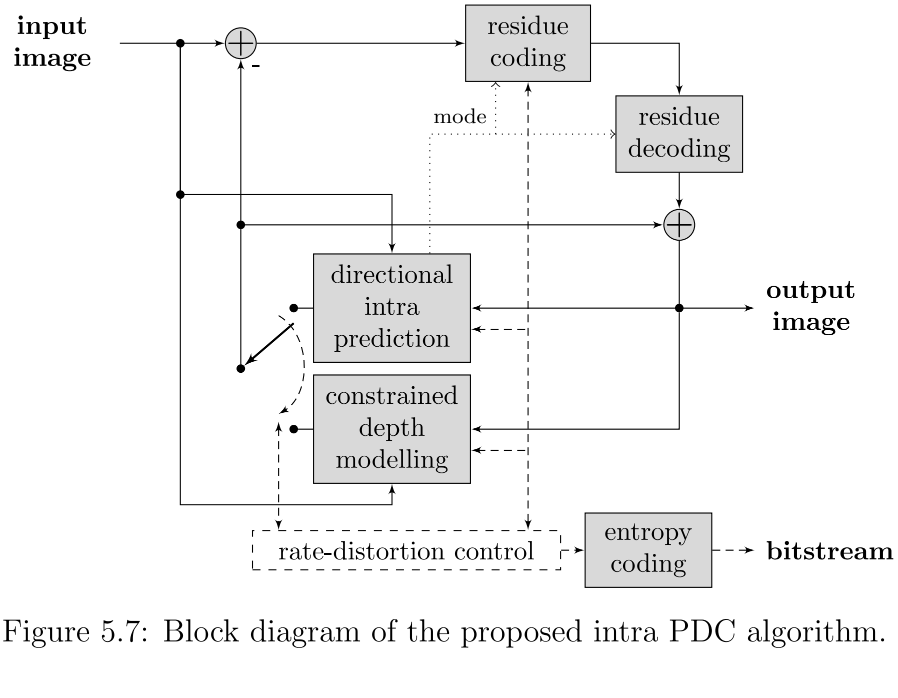


Figure 5.7: Block diagram of the proposed intra PDC algorithm.

prediction and residue coding. In the coding process, PDC firstly divides the input image into 64 × 64 pixel blocks. Each block is intra predicted based on the neighbouring reconstructed samples.

PDC intra prediction modes are the planar, DC and angular modes as proposed in the HEVC standard. Alternatively, the block may be encoded using a constrained depth modelling mode, designed for edges that are difficult to predict. This kind of edges is typically observed in the bottom-right region of the block, which cannot be predicted by directional intra prediction modes using left and top neighbouring block samples. The proposed constrained depth modelling mode allows to explicitly model those edges and approximate the surrounding smooth areas.

PDC encodes the residual information, given by the difference between the original and intra predicted signals, using a straightforward and efficient method based on linear fitting, which depends on the chosen intra prediction mode. This dependence on the prediction mode is indicated in Figure 5.7 by the dotted line. Like 3D-HEVC, PDC uses a depth lookup table to efficiently encode the residue signal values, mainly when depth maps present a very restricted depth range.

On the encoder side, most of the possible combinations of block partitioning and coding modes are examined. The best one is selected according to a Lagrangian RD cost. During the encoding process, each block is reconstructed like in the decoder. Reconstructed data is further used by the encoder to generate the prediction signal for the next blocks.

The encoded bitstream contains the flags used to signal the block partition, the DLT information and the symbols produced by the encoder blocks: directional intra prediction, constrained depth modelling mode and residue coding, as signalled by dashed lines in Figure 5.7. For entropy encoding, PDC employs the context adaptive m-ary arithmetic coding algorithm, based on the implementation of [68]. For each frame, the probability models are initialised with uniform distributions. A different context model, which depends on the block size, is used for most of the transmitted symbols.

5.4 Coding techniques of PDC algorithm

This section details the coding techniques comprising the PDC algorithm and highlights the main contributions of this work relative to the current existing techniques used in the 3D-HEVC encoder. The proposed techniques were designed to maximise the RD performance of the encoded depth maps and the quality of the synthesised views. However, in order for PDC to have a low computational complexity, some simplifications leading to sub-optimal decisions have been made. Thus, the proposed algorithm is a compromise between complexity and coding efficiency.

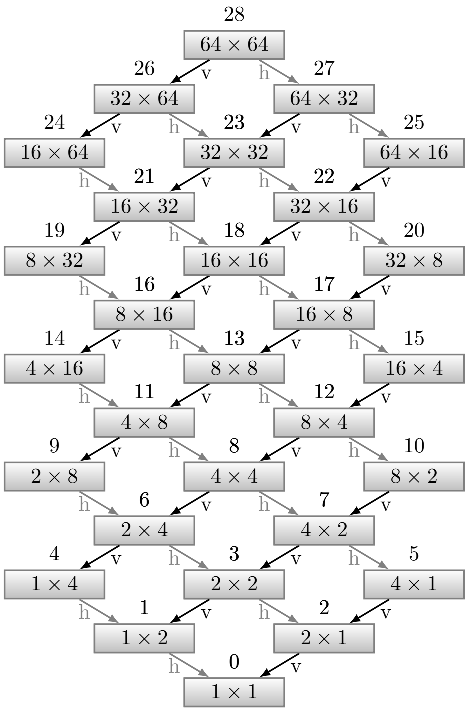


Figure 5.8: Possible block sizes in PDC and respective label numbers.

5.4.1 Flexible block partitioning

PDC divides the input depth map into fixed 64 × 64 pixel blocks. During the encoding process, each block can be further segmented through a flexible scheme based on the one of MMP algorithm, which recursively divides the block, either in the vertical or horizontal direction, down to the 1 × 1 size [66]. In this scheme, vertical partitioning is first applied and the left partition is processed before the right one. Then, the block is also partitioned in the horizontal direction and the top partition is processed before the bottom one. Each generated partition is recursively processed using the same partitioning scheme, until the smallest block size is reached. The possible block sizes are labelled from 0 up to 29, as illustrated in Figure 5.8. Note that, block sizes with a high ratio between horizontal and vertical dimensions (ratios larger than 4, *e.g.* 64 × 1) are not included in the proposed partitioning scheme because they significantly increase the encoder's computational complexity and have a small impact on the RD performance.

Despite the restriction to 29 possible block sizes, the complexity required to test all the possible block partitioning combinations in the encoder side still remains the main issue of the proposed flexible partitioning scheme. To mitigate this problem a quadtree block partitioning scheme was combined with flexible block partitioning. Three quadtree levels were defined at block sizes 16 × 16, 32 × 32 and 64 × 64. The four partitions, generated by each quadtree partitioning, are processed using a raster scan order. For each presented quadtree level, the flexible partitioning is used within

Table 5.1: Flexible segmentation restrictions per quadtree level.

Quadtree level	**Max. block area**	**Min. block area**
2 (64×64)	4096 (64×64)	256 (16×16)
1 (32×32)	1024 (32×32)	64 (8×8)
0 (16×16)	256 (16×16)	1 (1×1)

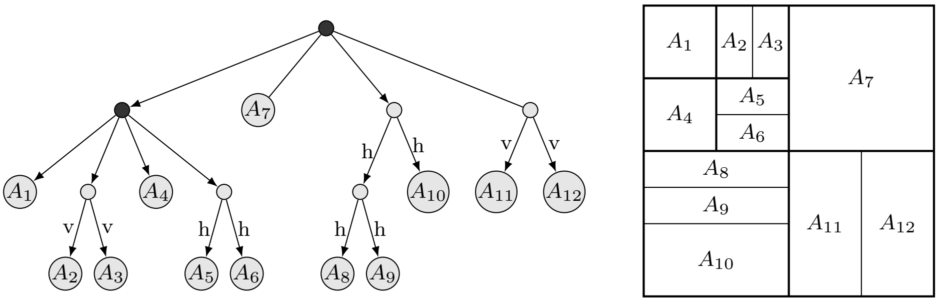


Figure 5.9: Example of an optimal block partitioning and corresponding segmentation tree in PDC algorithm using quadtree plus recursive flexible partitioning.

a restricted range of block sizes, which depends on the block area. Table 5.1 presents the proposed maximum and minimum block areas (and sizes) that are obtained by the flexible block partitioning scheme, for each available quadtree level.

In Figure 5.9, an example of an optimal segmentation tree (left) with the corresponding block partitioning scheme (right) is presented, based on quadtree plus recursive flexible partitioning. Darker nodes represent quadtree partition, while brighter nodes correspond to the flexible partitioning, being complemented with the “h” and “v” labels, for horizontal and vertical partitioning, respectively. Depth map coding using prediction and residue approximation is used at tree leaves, as labelled in Figure 5.9, by means of A_n, for $n = 1, .., 12$. In Figure 5.9, one can also observe the flexible partitioning within a quadtree level. There, sub-blocks as larger as 32×32 (represented by A_7) are used next to smaller non-square blocks, like 16×8 (represented by A_5). Note that some leaves could, eventually, be further segmented down to 1×1 size using flexible partitioning scheme.

5.4.2 Directional intra prediction framework

Combined with the flexible block partitioning scheme, the directional prediction is an effective technique in the PDC algorithm. The intra prediction framework is based on the one proposed to the current state-of-the-art HEVC standard [1]. It includes the intra planar, DC and 33 angular prediction modes, as described in Subsection 2.3.1. However, in PDC proposal some improvements were made for better prediction of depth map signals.

In order to keep the sharpness of edges and improve prediction results, the proposed framework does not smooth the reference neighbouring samples from which prediction is derived, as typically done in AVC and HEVC standards for prediction of natural image signals. Furthermore, PDC disables another filtering that HEVC uses over some predicted samples of DC, Angular 10 (horizontal) and Angular 26 (vertical) modes, specifically the samples of the first row and column of the predicted block. This is done in HEVC to smooth the transition between the reference neighbouring samples and the predicted block samples. In the case of depth maps, this filtering is not appropriate as it would blur the predicted depth edges.

This intra prediction model is able to produce a reliable depth map prediction and thus a small residual signal that can be encoded using a relatively low bitrate. The smooth and constant regions, which are very frequent in depth maps, are efficiently predicted by DC and planar prediction modes. In addition, angular modes combined with flexible block partitioning are able to represent the sharp edges which also characterise depth maps. The effectiveness of this method results from the piecewise edge approximation using a variety of rectangular sub-block sizes and prediction directions for each partition.

In order to further improve the performance of the proposed directional intra prediction for depth map coding using PDC algorithm, further solutions were investigated. As previously explained, PDC uses a great amount of block sizes which can be predicted based on 33 directional intra modes. Since some block sizes are very small or narrow, some directional intra prediction modes, namely adjacent directions, may be redundant and produce similar prediction patterns. For example, the 1 × 4 block size presents very few samples in the horizontal direction (1-wide pixel), which is not sufficient to project the left neighbouring reference samples into 17 clearly distinct prediction directions (Angular 2 up to Angular 18 modes). In this context, in order to avoid unnecessary calculations and to use less bits for directional intra prediction coding, a reduction in the set of prediction directions was investigated for the proposed algorithm according the used block size. Table 5.2 presents the proposed restricted direction set of angular prediction modes for each block size in the PDC algorithm. DC mode is enabled at all block sizes, while the planar is active for block sizes larger or equal to 2 × 2.

Besides this restriction, that is due to the large amount of small and narrow block sizes, the characteristics of depth maps may be exploited to further restrict the used directional intra prediction modes. Because depth maps present large smooth areas, multiple prediction directions may produce the same predicted samples. In order to exploit this prediction redundancy, an analysis of the reference samples in the neighbouring reconstructed blocks is performed and some prediction directions are disabled accordingly. This method provides both speed-up of the encoder and bitrate reduction, since a more limited set of directions is tested, and less bits are required

Table 5.2: Restricted direction set of available angular prediction modes per block size.

Block size $w \times h$	**Active angular modes**
$w \geq 16$ and $h \geq 16$	all modes
$w \geq 16$ and $h = 8$	all modes except 3, 5, 7, 9, 11, 13, 15, 17
$w = 8$ and $h \geq 16$	all modes except 19, 21, 23, 25, 27, 29, 31, 33
$w = 8$ and $h = 8$	even modes
$w = 8, 16$ and $h = 4$	even modes except 20, 24, 28, 32
$w = 4$ and $h = 8, 16$	even modes except 4, 8, 12, 16
$w = 8$ and $h = 2$	even modes except 20, 22, 24, 28, 30, 32
$w = 2$ and $h = 8$	even modes except 4, 6, 8, 12, 14, 16
$w = 4$ and $h = 4$	modes 2, 6, 10, 14, 18, 22, 26, 30, 34
$w < 4$ and $h < 4$	modes 2, 10, 18, 26, 34

Table 5.3: Groups of prediction modes defined according to the block neighbouring regions.

Group	**Neighbouring regions**	**Prediction modes**
1	top & left & top-left	modes 10 to 26, planar
2	left & down-left	modes 2 to 9
3	top & top-right	modes 27 to 34

to signal each directional mode.

The proposed method defines three groups of directional prediction modes, which may be disabled as a whole when the associated neighbouring reference samples are exactly constant. These groups of prediction modes and associated neighbouring regions are shown in Table 5.3. The group 1 contains all the directions that generate a prediction signal exclusively based on the top and left neighbourhood, including the top-left pixel. When these reference samples are constant, the associated modes of group 1 are disabled. DC mode can be chosen in place of the disabled modes of group 1, since it produces the same predicted samples. When the samples of the neighbouring left and down-left regions are constant, the modes of group 2 can be disabled. In this case, the Angular 10 mode (horizontal) is able to substitute these modes, producing the same results. Note that the Angular 10 mode, belongs to group 1, and it should be active if left and top neighbourhoods contain varying samples. Otherwise, if left-down, left, top-left and top neighbouring regions are constant, the DC mode should be able to replace all the modes of groups 1 and 2. Group 3 contains those modes that depend on top and top-right neighbouring regions, and can be replaced by Angular mode 26 (vertical).

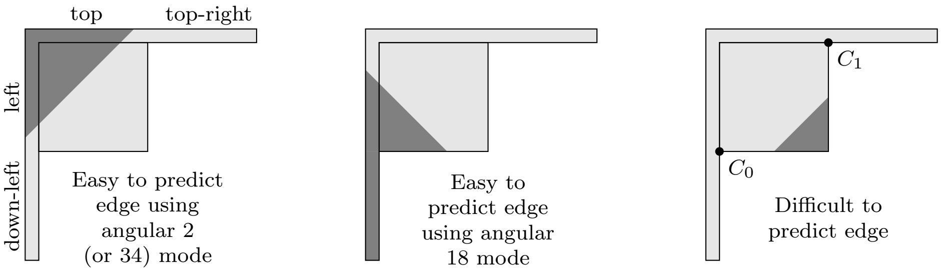

top
top-right
left
down-left
Easy to predict edge using angular 2 (or 34) mode
Easy to predict edge using angular 18 mode
C_1
C_0
Difficult to predict edge

Figure 5.10: Example of easy to predict edges (left and middle) and a difficult to predict edge (right).

5.4.3 Constrained Depth Modelling Mode

The Constrained Depth Modelling Mode (CDMM) is another important tool of the proposed PDC algorithm. The main idea behind this tool is to complement the intra directional prediction, by providing an alternative method that explicitly encodes depth edges in the bottom-right region of the block, that are hard to predict by directional intra prediction. This method is inspired on depth modelling modes used in 3D-HEVC, but several restrictions were applied to its design, in order to make it more efficient in the context of the PDC algorithm.

Intra prediction angular modes are able to represent most of the straight edges present in depth maps. However, some specific ones are difficult to predict. An example of a straight edge that is difficult to predict is illustrated in Figure 5.10. PDC intra prediction framework reasonably predicts straight edges coming from the left or top block neighbourhood. As can be observed, edges illustrated in the left and middle blocks of Figure 5.10 can be well represented by the proposed intra prediction framework, based on the left and top neighbouring reference samples. When an edge does not touch the left or top neighbouring samples, like the one shown in the right block of Figure 5.10, it is difficult to predict. In some cases, the top-right or down-left reference samples may provide the necessary information to predict this kind of edges. Unfortunately, these neighbourhoods are often unavailable. Furthermore, as illustrated in the right block of Figure 5.10, the visible edge in the block may not reach the top-right or left-down neighbouring region, if it does not maintain the straight shape outside of the predicting block.

In order to improve the representation of such difficult to predict edges, a depth modelling mode is proposed to represent edges below the block diagonal drawn between the down-left and top-right block corners, indicated in Figure 5.10 by the C_0 and C_1 points. The principle of the proposed CDMM consists in dividing the block into two partitions, which are approximated by constant values. The block partitioning should occur between two points of the right and bottom margins of the predicting block. As a second restriction to the proposed method, the line

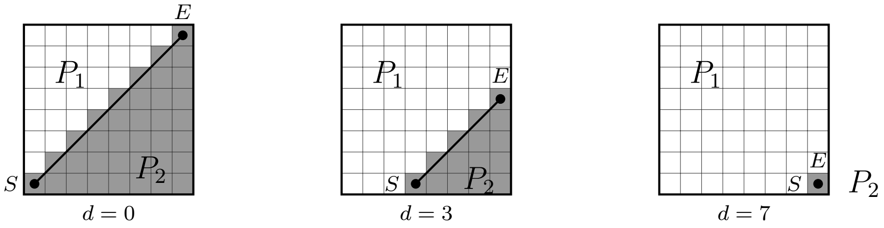


Figure 5.11: Block partition examples using the proposed CDMM.

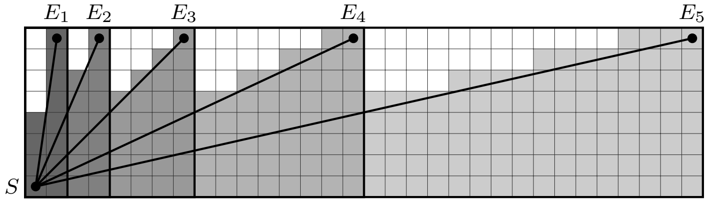


Figure 5.12: Different CDMM partition slopes provided by flexible partitioning.

drawn between the two chosen points should be parallel to the diagonal defined by the down-left and top-right block corners. Thus, it can be specified by just one parameter.

Figure 5.11 illustrates some partition possibilities of the proposed CDMM for an 8×8 square block. The imposed constraints highly simplify the signalling of the CDMM block partition, requiring a single value only, which is represented by the offset d in Figure 5.11. In this example, eight different partitions that vary between the minimum offset, $d = 0$, and the maximum offset, $d = 7$, can be employed. It can be observed in Figure 5.11 that block partitions are performed in the bottom-right half of the block and their slope is the same as the block down-to-top diagonal, satisfying the proposed constraints.

The restriction on the block partitioning slope is advantageous in terms of computational complexity because it avoids testing a lot of block partitions with different slopes. Furthermore, by using a unique partition slope depending on the block size, no bitstream overhead is required for its transmission. The main disadvantage of this partitioning restriction is the reduced flexibility to approximate depth map edges. However, the proposed PDC algorithm is able to alleviate this issue, by combining CDMM with the flexible partitioning scheme.

The large amount of block sizes generated through flexible partitioning provides up to five different CDMM partitioning slopes according to the possible down-to-top diagonals. Figure 5.12 illustrates these five CDMM line partition slopes generated from different block sizes available in the PDC algorithm. The blocks are overlapped and the available slopes are represented between the points S and E_n, for $n = 1, 2, 3, 4, 5$. The illustrated overlapped block sizes represent all the block width/height ratios available in PDC.

To generate the block partitions in the image sample domain, a simple formulation was derived. Let w and h define the block width and height, respectively, and $r = \max(w, h)/\min(w, h)$ define the ratio between block dimensions. The partition P_2 (see Figure 5.11) is defined by all the block samples (x, y) (with $x = 0, ..., w-1$ and $y = 0, ..., h-1$) that satisfy the following conditions:

$$\begin{cases} y + (x/r) \geq (h + d - 1), & \text{if } w > h \\ x + (y/r) \geq (w + d - 1), & \text{otherwise} \end{cases} \tag{5.8}$$

where $d = 0, ..., \min(w, h)$ is the offset variable used to change the partitioning line position, as illustrated in Figure 5.11.

The above description clearly highlights the importance of flexible partitioning scheme for the proposed CDMM. It allows CDMM to use different block partition possibilities, using a minimal overhead. Only the position of the CDMM block partition is signalled through offset d, while the partition slope is implicitly derived from the used block size.

CDMM block partitioning generates two partitions, whose depth values are approximated by using a constant value. For P_1 partition, the approximation coefficient is derived from the block neighbourhood, namely through the mean of the left and top neighbouring reconstructed samples. The constant approximation of P_2 partition is explicitly transmitted to the decoder. For that, the mean value of the original samples in P_2 is computed and the difference between the constant values P_1 and P_2 is encoded using the DLT technique (detailed in Subsection 5.1.3), as done for the residue generated by directional intra prediction. The residual information generated by the proposed approximations is bypassed, not requiring any extra bits.

Unlike directional intra prediction, which can use block sizes down to 1×1 pixel, CDMM is disabled for smaller block sizes, namely blocks with an area equal or less than 5 pixels (see Figure 5.8). This is so because at these levels the right-bottom block partition is not feasible, since the blocks are only one pixel wide in either dimension directions.

5.4.4 Residual signal coding

The residual signal is given by the difference between the original depth and predicted samples. In PDC, the flexible block partitioning scheme combined with the directional intra prediction and CDMM provides very efficient prediction, resulting in a highly peaked residue distribution centered at zero. For this reason, PDC does not use the DCT, but an alternative approach which often assumes null residue and uses linear modelling in the remaining cases. The simplicity of the proposed approach is also advantageous in terms of computational complexity. Figure 5.13

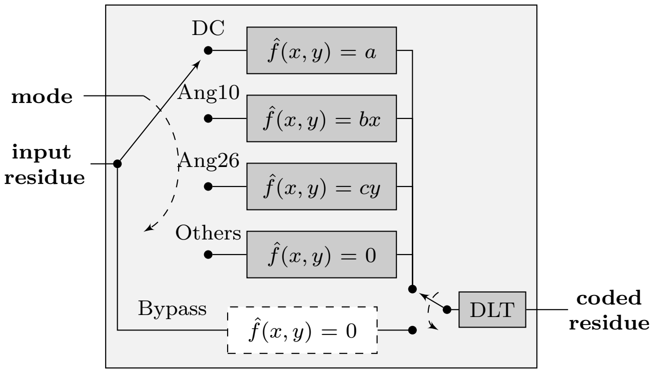


Figure 5.13: Detailed schematic of the PDC residue coding method.

illustrates the schematic of the proposed residue coding method. Four approximation models are available: constant, horizontal linear and vertical linear, as well as a special case of null residue. In this approach, the residue block size matches the prediction block size. Thus, the residue block is no further partitioned.

Linear residue approximation is mainly intended to encode a residue produced on smooth areas. Constant residue approximation is employed only when DC mode is used. On the other hand, linear fitting is used to approximate the residues generated by the horizontal (Angular 10) and vertical (Angular 26) directions. Note that, a row- or column-wise formulation is used without any additional offset, as formulated in Figure 5.13. The main motivation for Angular 10 and Angular 26 prediction modes is their ability to predict the offset component of depth regions similar to horizontal or vertical linear planes, respectively.

These modes (DC, Angular 10 and Angular 26) tend to be mostly used at smooth regions, where the residue often can be easily approximated by linear fitting. For the remaining planar and angular modes, PDC always encodes a null residue. Angular prediction modes are mostly intended for depth edge prediction - they usually choose angular modes whose direction matches the predicting edge. When the prediction direction does not match the edge, highly irregular residue patterns, which cannot be efficiently represented by linear modelling, tend to be generated. For this reason, in order to reduce the overhead, we disable residue coding on angular prediction modes. Note that, when depth map edges cannot be efficiently predicted by the existing modes, encoder RD control tends to further partition the block into smaller sub-blocks that can be better predicted and encoded.

The linear model coefficients used to approximate the residue signal (Figure 5.13) are computed using a straightforward method. For the DC mode, residue approximation is based on coefficient a, given by the mean value of the residue block. For the Angular 10 and Angular 26 modes, the mean of the rightmost column and the mean of the bottom row of the residue block are transmitted, respectively. Note that the transmitted mean values computed from the right column or bottom row do not correspond directly to the value of the linear coefficients b and c represented in Figure 5.13. However, they can be easily derived at the decoder from the received values.

Depending on the chosen prediction mode, which is known in both the encoder and decoder, one of the residue approximation models should be transmitted. However, for a more efficient RD coding, PDC allows to bypass residue approximation, through the use of a binary flag. The bypass mode is tested as an alternative residue coding mode that can be chosen, depending on the RD cost evaluation (see Figure 5.13). In practice, this flag provides a lower bitrate coding for the zero coefficient used in the approximation models.

PDC uses a simple residue quantisation scheme, by rounding the residue model coefficients in the form of mean values to the nearest integer. Note that in opposition to most codecs like 3D-HEVC, bitrate control is not performed by the quantisation step size, that controls only the precision of the transmission of the residue coefficients. As mentioned before, bitrate is controlled through the Lagrangian multiplier. When the intensity range of the input depth map is quantised, the DLT algorithm is used, as presented in Subsection 5.1.3.

5.4.5 Bitstream syntax and context modelling

The bitstream symbols encoded in PDC include the block partitioning flags, intra prediction modes, residue approximation coefficients and DLT. When the optimal partitioning tree and encoded depth data is determined for a 64 × 64 block, the symbols are organised as a string and represented in the bitstream. Context adaptive arithmetic coding is used to efficiently encode the symbols. For most symbols, an appropriate context model is derived and used. Probability models are initialised with uniform distributions and they are updated whenever a symbol is encoded. In order to guarantee that PDC can independently decode each frame, the arithmetic encoder probability models are reset with uniform distribution for each frame.

For quadtree partitioning, a binary flag is transmitted indicating whether the block is partitioned into 4 sub-blocks (q_1) or not (q_0). For flexible partitioning a ternary flag is used, indicating whether the block is partitioned, in horizontal (f_1), in vertical (f_2), or not further partitioned as it is a leaf of the partitioning tree (f_0). The partitioned block is encoded in the bitstream using the partitioning order as described in Subsection 5.4.1, for quadtree and flexible partitioning schemes.

Considering the example of Figure 5.9, presented in Subsection 5.4.1, the block would be encoded by the following strings of symbols:

$$
\begin{aligned}
&q_1\\
&q_1\ f_0\ A_1\ f_2\ f_0\ A_2\ f_0\ A_3\ f_0\ A_4\ f_1\ f_0\ A_5\ f_0\ A_6\\
&q_0\ f_0\ A_7\\
&q_0\ f_1\ f_1\ f_0\ A_8\ f_0\ A_9\ f_0\ A_{10}\\
&q_0\ f_2\ f_0\ A_{11}\ f_0\ A_{12}.
\end{aligned}
$$

The transmitted string would be given by the concatenation of these smaller strings, which are presented in a convenient way. The first string corresponds to the first quadtree node, while the following ones contain the symbols used to encode the four nodes (quadtree level 1) in a raster scan order, as a result of the first quadtree partitioning. The A_n strings mean that all symbols are used to encode the sub-block (by using directional intra prediction, CDMM and residue coding). They are signalled whenever a non-segmentation flag, f_0, is used. Note that at level 0 of quadtree (block size 16×16), the quadtree flag is omitted, since the block cannot be further partitioned. The same procedure is carried out when the flexible partitioning reaches block size 1×1.

PDC uses an appropriate context modelling, in order to efficiently exploit the fact that larger blocks are mostly used at low bitrate RD points and smaller blocks at high bitrates. For quadtree partitioning flags three independent probability models are used, depending on the quadtree level where the flag is transmitted. For the flexible partitioning scheme, context modelling depends on both, the block size (of flexible partitioning) and quadtree partition level, from where flexible partitioning was initiated. Thus, based on Table 5.1, 41 probability models (which correspond to $19 + 11 + 11$ block sizes used from quadtree levels 0, 1 and 2, respectively) are available to encode the flexible partitioning flags.

Intra prediction mode flags consist of 36 symbols that include planar, DC, directional modes and CDMM. This intra information is transmitted whenever the non-segment flag f_0 is used to signal a new encoded sub-block. For each block size, a different context associated to independent probability models, is used in AAC. The context modelling is important, since the number of available prediction modes depends on the block size. Furthermore, as explained in Subsection 5.4.2, the probability models can be adaptively adjusted depending on the characteristics of the neighbouring block samples.

In the case of CDMM, two additional symbols are transmitted, specifically, the partitioning offset symbol d and the mean value of the down-right block partition. In the AAC, an independent probability model associated to the block size is used to encode the partitioning offset d. Note that, this context is important because the range of values used in symbol d depends on the block size. To encode the mean value of the down-right block partition a fixed context is used for all block sizes.

When DC, Angular 10 or Angular 26 intra prediction modes are used, the residual information can be linearly approximated. In these cases, the binary flag, which indicates whether the linear approximation is applied or not, is encoded using an independent context per block size. If a linear approximation is used, the linear coefficient index derived from DLT is encoded using a fixed context for arithmetic coding. For the remaining directional intra prediction modes, a null residue is assumed and no symbol is transmitted.

In addition to the symbols referred above, PDC also transmits the DLT table used to encode the residual signal. Since the proposed PDC algorithm is designed for intra coding, DLT is computed and transmitted for each encoded depth map frame. Unlike the previous symbols, DLT does not use context-based coding, because it consumes a negligible amount of the bitrate. Thus, before to encode information for each frame, PDC writes 256 bits (one bit per intensity value) directly into the bitstream indicating which intensity values are available in the 8-bit depth intensity range (ranging from 0 and 255). Alternative solutions to encode DLT could be investigated, for example by calculating DLT and transmitting it only once for a group of pictures.

5.5 PDC encoder control

The encoder control plays an important role in the PDC rate-distortion performance, as well as in the computational complexity. During the encoding process, PDC seeks to minimise the following Lagrangian RD cost function:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}), \tag{5.9}$$

where $D(\mathcal{T})$ is the block distortion used to represent the partition tree $\mathcal{T}$, and $R(\mathcal{T})$ is its rate. λ is the Lagrangian multiplier used to control the target bitrate. The default distortion metric used in PDC is the SSE metric, typically used on standard image/video encoders. In this work we also implemented and evaluated the VSO method for distortion measure in the PDC algorithm, using the SVDC approach [75], as described in Subsection 5.1.6. The bitrate stands for the number of bits required to arithmetically encode the partition tree (Subsection 5.4.1), plus the symbols used to encode prediction mode and residue signal. Examples of these are the residue coefficient, prediction mode and CDMM related symbols.

PDC optimisation is performed independently for each 64 × 64 block, by recursively partitioning each sub-block down to the smaller block size and choosing the optimal block representation. This process involves the creation of a fully expanded partition tree, which is posteriorly pruned by evaluating the coding costs at the tree nodes. Whenever the parent's node cost is inferior to the sum of the children's node costs, the given node is not partitioned.

At each tree leaf the optimal representation is found by evaluating all intra prediction modes plus residue coding. In the case of CDMM, all the partitions given by each offset, $d = 0, ..., min(w, h)$, are examined (see Subsection 5.4.3) and the optimal one is selected. When using directional intra prediction, PDC tests all the available prediction modes (including planar and DC) at the current block size and encodes the residue for the case of DC, Angular 10 and Angular 26 modes. For these modes, the encoder also tests the bypass mode, which assumes null residue, and chooses the solution which minimises the Lagrangian cost function.

Most of the PDC encoder computational complexity is due to the directional intra prediction, CDMM mode and residue coding, because they are repeated thousands of times for the 29 types of block sizes. Although most of the techniques described in the previous sections were designed with the computational complexity in mind, an improved solution for the encoder RD control was proposed to further reduce the encoder complexity. This solution forces an early termination of block partitioning whenever the distortion is smaller than 10% of the Lagrangian cost ($D(\mathcal{B}) < 0.1J(\mathcal{B})$), for some sub-block $\mathcal{B}$ being encoded.

Other improvements were further implemented at PDC encoder control. For example, whenever the cost of the left (or top) child node is higher than the parent's node cost, PDC reduces the encoding time by aborting further partitioning of right (or down) nodes, since the overall cost of the child nodes is known to be higher. These improvements combined with the proposed coding techniques resulted in an efficient depth map coding algorithm with a reasonable computational complexity.

5.6 Experimental Results

This section presents the experimental results of the proposed intra depth map coding solution in comparison with the current state-of-the-art intra coding techniques present in 3D-HEVC algorithm. The experimental setup is based on the common test conditions (CTC) document for 3D video core experiments [112], with some modifications detailed in the following. Simulations were run under three-view configuration using recommended test sequences and view numbers, as presented in Table 5.4.

The three-view depth maps of each sequence were encoded using PDC algorithm and 3D-HEVC reference software version HTM-13.1 [113] for comparison purposes. Since PDC algorithm is designed for intra coding, the inter-view, temporal and

Table 5.4: Test sequences used for experiments.

Test Sequence	Resolution	No. Frames	Views
Balloons	1024 × 768	300	1-3-5
Kendo	1024 × 768	300	1-3-5
Newspaper1	1024 × 768	300	2-4-6
GT_Fly	1920 × 1088	250	9-5-1
Poznan_Hall2	1920 × 1088	200	7-6-5
Poznan_Street	1920 × 1088	250	5-4-3
Undo_Dancer	1920 × 1088	250	1-5-9
Shark	1920 × 1088	300	1-5-9

inter-component correlations are not exploited. In order to fairly compare the proposed algorithm with the current state-of-the-art depth intra coding techniques of 3D-HEVC, a reference HTM configuration (*RefHTM*) was created, based on the non-CTC all-intra encoder configuration provided by the HTM software. The inter-component prediction (which is present on HTM I-frames) was disabled in *RefHTM* configuration by turning off the Contour prediction mode. This *RefHTM* configuration makes 3D-HEVC operation similar to PDC, in the sense that only intra coding techniques are used. Also, to evaluate the RD performance and computational complexity of the original coding techniques used in PDC algorithm using the SSE distortion metric, the View Synthesis Optimisation (VSO) method was disabled on *RefHTM* configuration. Additional experimental tests for PDC algorithm using VSO method as distortion metric, based on SVDC approach [75], were also performed and compared with 3D-HEVC.

For all experiments, CTC recommended QP pairs (for texture and depth) were used with 3D-HEVC, namely (40,45), (35,42), (30,39) and (25,34). For PDC, the λ values 1200, 500, 250 and 75 were chosen as the ones that best match the bitrate produced by CTC recommended QPs for 3D-HEVC using *RefHTM* configuration. The PSNR metric is commonly used to evaluate the objective quality of the decoded video. However, as previously discussed, PSNR results are not very meaningful, since depth maps are not directly presented to the viewer, but rather used for view synthesis purposes. In these experiments we used the depth map evaluation method suggested by the experts of ISO/IEC and ITU-T JCT-3V group, which is described in the CTC document [112]. This evaluation methodology consists in assessing the quality of the generated virtual views, using the decoded depth data and original texture views versus the generated virtual views using the original uncompressed depth and the original texture views.

The quality of six synthesised views placed between the positions of the encoded depth maps has been measured by luminance PSNR. Note that such a methodology allows to assess depth map coding quality losses, excluding up to some degree the influence of the particular view synthesis algorithm used. Furthermore, by using always original texture data to generate virtual views, a more accurate evaluation of the encoded depth maps performance could be made, without interferences of coding artefacts present on decoded texture views. For the purpose of view synthesis, state-of-the-art view synthesis software for linear camera arrangement implemented in HTM software has been used [129].

5.6.1 Evaluation of PDC algorithm for intra coding

The first set of experimental results using the proposed PDC algorithm and 3D-HEVC *RefHTM* configuration, which use SSE as distortion metric for depth maps, are presented in Table 5.5. The evaluated virtual views are indicated by *vv1*, *vv2* and *vv3*, which are interpolated between the first and second depth map views, and by *vv4*, *vv5* and *vv6*, which are interpolated between the second and third depth map views. The PSNR of each virtual view as well as the average PSNR for all views (*avg-vv*) are presented for the used RD points (identified by *p1*,...,*p4*). The sum of the bitrate of the three depth map views (in kbits per second) and the average PSNR results of the virtual views were used to compute the Bjontegaard Delta Rate (BDRATE) results [70], which are shown in the last column of Table 5.5. BDRATE represents the average bitrate differences between RD curves for the same PSNR of virtual views, where negative values indicate actual bit-rate savings of PDC algorithm over 3D-HEVC.

The results presented in Table 5.5 clearly show the advantage of the proposed intra coding approach over the current state-of-the-art 3D-HEVC standard. The average bitrate savings of PDC algorithm for all the tested sequences is approximately 6%. The best result is observed for *Ghost Town Fly* sequence, where PDC saves 13% of bitrate in comparison to 3D-HEVC using *RefHTM* configuration. For the worst case, PDC presents approximately the same performance of 3D-HEVC, as can be observed for *Poznan Street* sequence. Note that PDC performance gains relative to 3D-HEVC are not constant, varying for different sequences. This is expected, since depth maps present distinct features that are differently exploited by PDC and 3D-HEVC algorithms.

Representative results for subjective evaluation of the PDC performance are presented in Figure 5.15. A detail of the first frame of the third virtual view (vv3) of Poznan Street sequence is presented. The virtual view was synthesised using reference uncompressed depth maps (left), 3D-HEVC encoded depth maps using QP 42 (middle) and PDC encoded depth maps using Lambda 500 (right), which correspond to the rate-distortion point 3 (p3) of Table 5.5. In the example of

Table 5.5: Rate-distortion results for depth map coding using PDC and 3D-HEVC (*RefHTM* configuration) algorithms.

Sequence	RD point	**PDC**								**3D-HEVC (*RefHTM* configuration)**								**BDRATE**
		All depths rate (kbps)	PSNR (dB) of virtual views (vv)							All depths rate (kbps)	PSNR (dB) of virtual views (vv)							
			avg-vv	vv1	vv2	vv3	vv4	vv5	vv6		avg-vv	vv1	vv2	vv3	vv4	vv5	vv6	
Balloons	p4	517,45	44,29	44,39	43,17	45,20	45,04	43,31	44,62	527,22	44,23	44,31	43,18	45,13	45,02	43,25	44,49	**-1,79%**
	p3	794,19	45,58	45,63	44,47	46,49	46,27	44,68	45,96	746,51	45,34	45,38	44,29	46,22	46,10	44,44	45,62	
	p2	1121,34	46,71	46,74	45,64	47,64	47,40	45,83	47,03	1048,90	46,45	46,46	45,41	47,35	47,16	45,54	46,75	
	p1	1989,73	48,92	48,82	47,92	49,92	49,53	48,09	49,24	1808,33	48,40	48,34	47,47	49,34	49,12	47,51	48,63	
Kendo	p4	456,44	40,73	43,87	41,84	41,72	40,08	38,35	38,53	462,36	40,54	43,75	41,60	41,47	39,92	38,04	38,45	**-6,32%**
	p3	683,47	41,88	45,20	43,26	42,98	41,04	39,25	39,53	646,36	41,48	44,78	42,75	42,56	40,66	38,84	39,31	
	p2	940,28	42,95	46,32	44,43	44,35	41,89	40,15	40,54	895,22	42,55	45,85	43,90	43,89	41,51	39,71	40,44	
	p1	1609,54	45,10	48,54	46,63	46,87	43,65	42,07	42,84	1507,68	44,59	47,86	46,02	46,30	43,10	41,56	42,71	
Newspaper	p4	666,30	39,20	38,72	37,82	39,86	40,43	38,35	40,03	678,00	39,15	38,70	37,80	39,85	40,37	38,29	39,89	**-2,70%**
	p3	1067,29	40,43	39,78	39,01	41,08	41,70	39,64	41,34	988,40	40,15	39,52	38,73	40,77	41,41	39,42	41,05	
	p2	1553,10	41,43	40,73	40,06	42,22	42,59	40,61	42,38	1429,92	41,13	40,47	39,77	41,78	42,29	40,40	42,05	
	p1	2805,18	43,25	42,32	41,95	44,13	44,38	42,44	44,27	2532,38	42,86	42,10	41,54	43,72	44,02	42,08	43,73	
GT Fly	p4	543,66	41,40	41,87	40,35	42,09	42,05	40,31	41,73	578,23	41,15	41,62	40,09	41,85	41,80	40,02	41,50	**-13,07%**
	p3	843,53	42,64	43,07	41,57	43,37	43,30	41,56	42,96	806,46	42,08	42,53	41,02	42,78	42,72	40,98	42,43	
	p2	1210,91	43,84	44,27	42,78	44,59	44,51	42,75	44,13	1160,24	43,21	43,64	42,17	43,93	43,86	42,12	43,55	
	p1	2211,44	46,19	46,56	45,15	46,99	46,90	45,11	46,41	2117,57	45,42	45,80	44,38	46,17	46,12	44,33	45,69	
Poznan Hall2	p4	204,67	46,27	46,87	44,76	46,44	47,37	45,29	46,86	203,51	46,12	46,81	44,75	46,13	47,09	45,22	46,69	**-7,31%**
	p3	273,79	47,85	48,49	46,41	47,91	49,04	46,97	48,29	264,36	47,32	47,97	46,00	47,33	48,37	46,48	47,80	
	p2	354,10	49,12	49,86	47,70	49,12	50,37	48,26	49,43	339,91	48,52	49,02	47,19	48,55	49,60	47,76	49,01	
	p1	571,05	51,48	52,20	49,98	51,40	52,88	50,61	51,79	538,12	50,67	51,14	49,25	50,63	51,87	49,96	51,19	
Poznan Street	p4	443,89	43,23	44,32	42,38	44,10	43,63	41,55	43,41	437,17	43,12	44,20	42,28	44,09	43,44	41,42	43,31	**0,07%**
	p3	678,13	44,33	45,43	43,53	45,18	44,70	42,68	44,50	625,95	44,14	45,15	43,28	45,01	44,54	42,51	44,34	
	p2	986,16	45,16	46,20	44,34	46,09	45,56	43,50	45,25	905,54	44,96	46,02	44,13	45,85	45,38	43,31	45,11	
	p1	1979,48	46,64	47,59	45,93	47,63	47,02	44,92	46,73	1772,76	46,34	47,38	45,59	47,24	46,70	44,69	46,46	
Dancer	p4	393,16	37,60	37,75	37,12	37,96	37,73	37,10	37,92	401,49	36,86	37,36	36,07	37,18	37,31	36,09	37,17	**-5,73%**
	p3	541,14	38,80	38,72	38,47	39,39	38,58	38,35	39,29	527,45	38,07	38,43	37,49	38,43	38,33	37,39	38,37	
	p2	697,54	39,90	39,57	39,69	40,53	39,55	39,59	40,46	676,57	39,48	39,55	38,97	40,04	39,51	38,87	39,91	
	p1	1077,78	41,96	41,42	41,92	42,82	41,36	41,73	42,49	1026,09	41,86	41,31	41,58	42,75	41,51	41,47	42,55	
Shark	p4	1135,87	41,71	42,01	40,86	42,21	42,15	40,92	42,10	1189,39	41,44	41,74	40,55	41,94	41,91	40,63	41,89	**-10,97%**
	p3	1893,69	43,30	43,56	42,44	43,83	43,76	42,51	43,71	1832,29	42,78	43,07	41,87	43,28	43,27	41,96	43,22	
	p2	2704,78	44,64	44,87	43,75	45,18	45,11	43,86	45,07	2729,00	44,24	44,52	43,30	44,78	44,75	43,39	44,70	
	p1	4728,10	47,09	47,24	46,16	47,72	47,53	46,28	47,59	4737,46	46,59	46,85	45,61	47,13	47,14	45,75	47,07	

Table 5.6: Encoding time results (in seconds per frame) for each RD point and used test sequence using PDC and 3D-HEVC (*RefHTM* configuration) algorithms.

Test sequences	PDC (seconds per frame)					3D-HEVC, only depth, (seconds per frame)					Ratio				
	p4	p3	p2	p1	**Average**	p4	p3	p2	p1	**Average**	p4	p3	p2	p1	**Average**
Balloons	3.03	3.59	4.06	4.83	**3.88**	8.69	9.03	8.90	9.31	**8.98**	0.35	0.40	0.46	0.52	**0.43**
Kendo	2.71	3.19	3.59	4.21	**3.43**	8.63	8.77	9.14	8.89	**8.86**	0.31	0.36	0.39	0.47	**0.39**
Newspaper	4.16	5.11	5.93	7.38	**5.65**	9.92	10.07	10.64	11.00	**10.41**	0.42	0.51	0.56	0.67	**0.54**
GT Fly	6.46	7.96	9.33	12.31	**9.02**	26.73	27.57	28.20	28.66	**27.79**	0.24	0.29	0.33	0.43	**0.32**
Poznan Hall2	2.63	3.31	3.90	4.91	**3.69**	19.83	20.67	21.68	22.53	**21.18**	0.13	0.16	0.18	0.22	**0.17**
Poznan Street	6.24	8.58	10.81	15.90	**10.38**	23.84	25.09	27.23	27.62	**25.95**	0.26	0.34	0.40	0.58	**0.39**
Dancer	4.40	5.45	6.29	8.05	**6.05**	24.24	24.97	25.75	26.39	**25.34**	0.18	0.22	0.24	0.31	**0.24**
Shark	6.53	8.20	9.61	12.21	**9.14**	21.80	22.24	24.96	25.40	**23.60**	0.30	0.37	0.39	0.48	**0.38**
Average	**4.52**	**5.67**	**6.69**	**8.73**	6.40	**17.96**	**18.55**	**19.56**	**19.98**	19.01	**0.27**	**0.33**	**0.37**	**0.46**	0.36

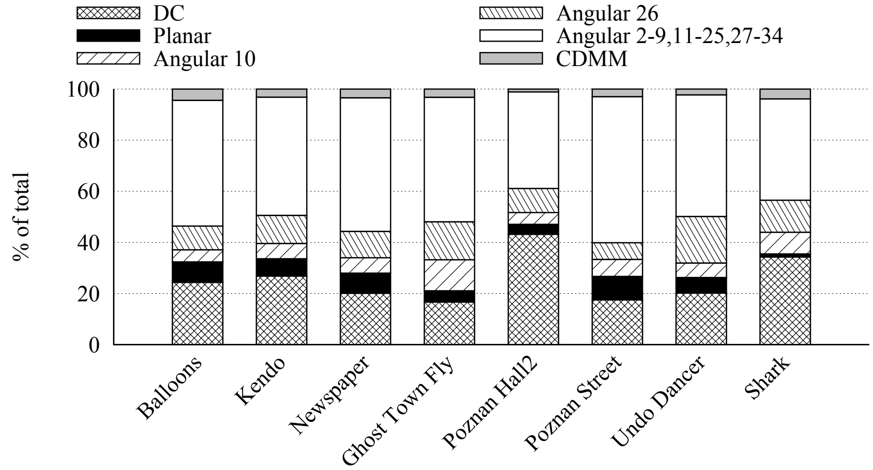
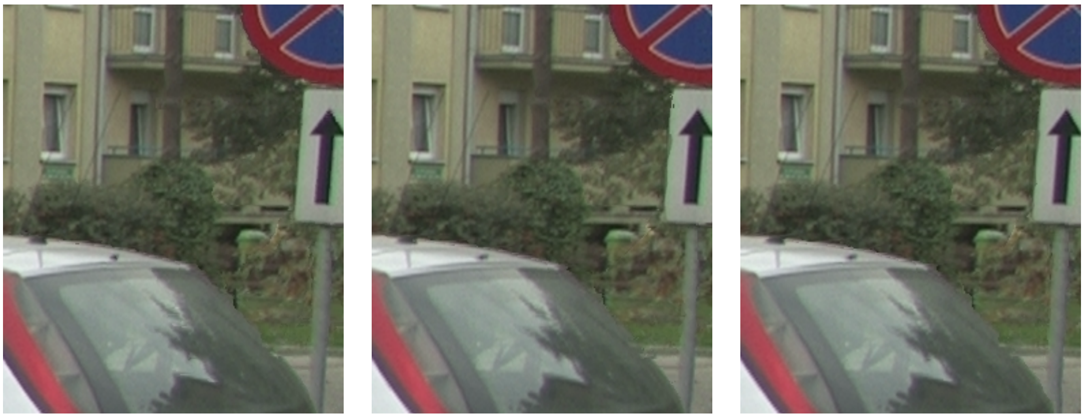


Figure 5.14: Average prediction mode usage in PDC algorithm for all tested lambdas and encoded views of each test sequence.



(a) Reference (b) 3D-HEVC (c) PDC

Figure 5.15: Detail of the third virtual view (vv3) of Poznan Street (frame 0) synthesised using a) reference uncompressed depth maps, b) 3D-HEVC (*RefHTM*) encoded depth maps and c) PDC encoded depth maps, for rate-distortion point 3 (p3).

Figure 5.15, one may observe some rendering distortions in the reference virtual view synthesised using the uncompressed depth maps. This is mainly due to the acquisition noise of depth maps. In these cases, rendering distortions are inevitable using both compressed and uncompressed depth maps. From the results using the

3D-HEVC and PDC encoded depth maps, one may observe that PDC presents a similar subjective quality. Its main advantage is the bitrate savings achieved for similar quality as presented in Table 5.5.

The computational complexity results for PDC and 3D-HEVC using *RefHTM* configuration are presented in Table 5.6. The average number of seconds used to encode each depth map frame is shown for each rate point of each recommended test sequence. These time values were obtained using a 2.00 GHz Intel Xeon E5-2620 CPU running GNU/Linux (Ubuntu 14.04) and give us a rough idea of the method's computational complexity. Table 5.6 shows that PDC presents a lower computational complexity, with most encoding times being inferior to 50% of the ones of 3D-HEVC. One may observe that PDC presents higher variation of encoding times between different sequences and λ values. This can be justified by the use of a condition for early termination of block partitioning in PDC, which mainly reduces encoding times at lower bitrates or when sequences present large smooth regions.

In Figure 5.14, we represent some statistical results, namely the distribution of chosen intra prediction modes in PDC algorithm, using all lambdas for each sequence. These results show that DC mode presents a high usage rate, which is mainly due to the large smooth regions present in depth maps. DC mode is also chosen more often than Planar mode. This is because constant residue coding is only available for DC prediction, forcing its usage when encoding of prediction error reduces RD cost. Horizontal and vertical modes also present a high usage rate in comparison to other angular modes. These two modes are also used in many smooth areas because linear residue approximation is available to encode the prediction error of these modes. Despite presenting a low usage rate, CDMM plays an important role in achieved results. Experiments using PDC without CDMM showed an average BDRATE performance of -2.5% relative to 3D-HEVC using *RefHTM* configuration (in contrast to the previously presented gain of -6%), being inferior to 3D-HEVC for some sequences.

5.6.2 Evaluation of PDC algorithm using VSO metric

In this section we evaluate the PDC algorithm using the VSO method as distortion metric, based on the SVDC approach [75]. In order to compare with 3D-HEVC, the *RefHTM+VSO* configuration was created by enabling VSO method. The VSO string (of HTM configuration file) was also modified to use original texture views instead of encoded ones. As the evaluation methodology uses the original texture views and encoded depth maps for view synthesis, the VSO method must also use the original texture views, for optimal coding performance. PDC λ values were adjusted to approximately match RD distortion curves of 3D-HEVC, resulting in

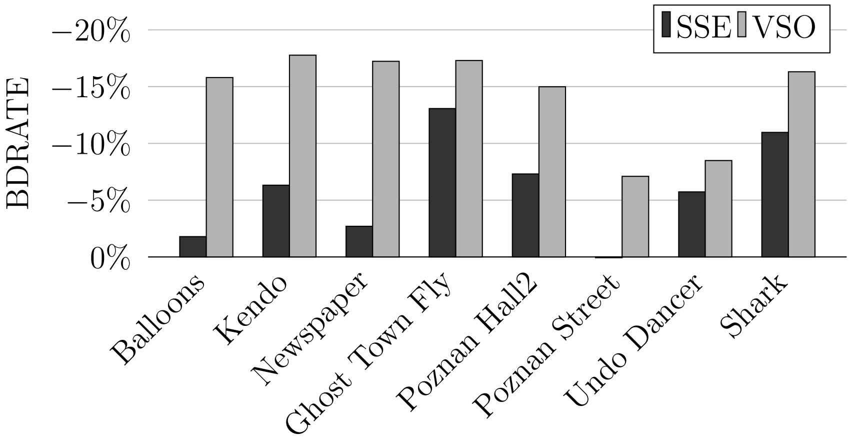


Figure 5.16: Average BDRATE values of PDC algorithm using SSE and VSO distortion metrics relative to 3D-HEVC using *RefHTM* and *RefHTM+VSO* configurations, respectively.

the values 500, 100, 25 and 5.

Average BDRATE results for *PDC+VSO* using 3D-HEVC *RefHTM+VSO* configuration as reference are presented in Figure 5.16 for each test sequence (bright bars). For comparison purposes, we plotted results of previous tests based on SSE distortion metric (dark bars), specifically the values of the last column of Table 5.5. These results clearly show that PDC rate-distortion performance highly benefits from VSO method, achieving an average BDRATE gain of -14.3% when compared to 3D-HEVC using *RefHTM+VSO* configuration.

For illustration of RD performance of the discussed configurations at different bitrates, Figure 5.17 represents several RD curves using PDC and 3D-HEVC algorithms to encode depth maps of *Shark* sequence. These RD curves correspond to the average PSNR values of virtual views generated using encoded maps in function of the total bitrate used to encode the three depth map views. The two lower RD curves compare PDC with 3D-HEVC *RefHTM* configuration, using SSE as distortion metric. The advantage of PDC is clear, matching the BDRATE gain of -10.98% represented in Table 5.5.

The previously discussed results using VSO are represented by *PDC+VSO* and *RefHTM+VSO* curves of Figure 5.17. We can observe that PDC tends to achieve superior RD performance at higher bitrates when using VSO method. This tendency can be related to the highly flexible block partitioning scheme used in PDC, since block partitions are more frequent at higher bitrates. We also compare PDC with *RefHTM+VSO+Contour* configuration which enables inter-component prediction on 3D-HEVC intra frame coding. Although the proposed PDC algorithm does not exploit redundancy from texture frames, we can observe that *PDC+VSO* still presents a superior RD performance when compared to *RefHTM+VSO+Contour*

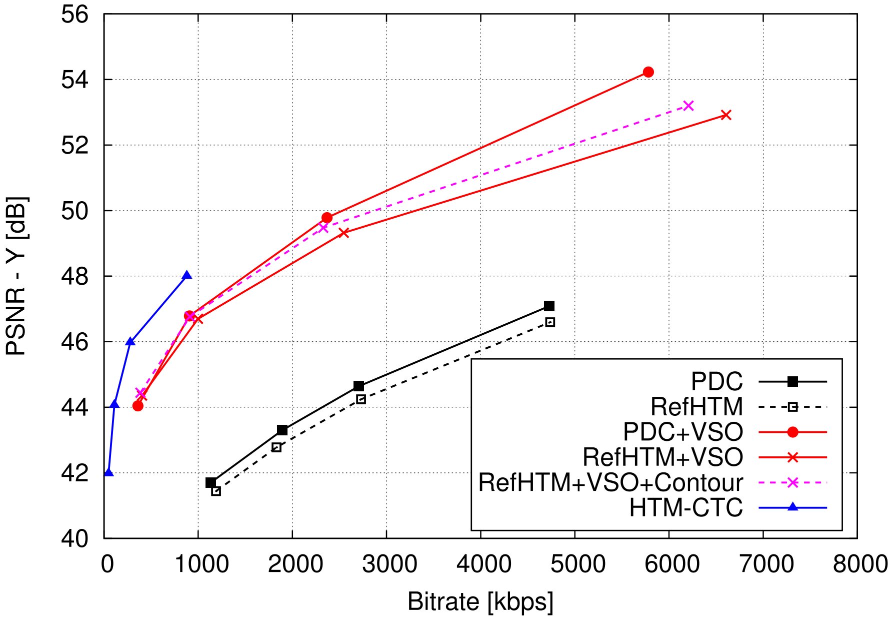


Figure 5.17: Average RD performance of shark sequence using PDC and 3D-HEVC algorithms under various configurations.

configuration for most RD points of *Shark* sequence. Experiments for all sequences revealed an average BDRATE gain of -11% for *PDC+VSO* in relation to *RefHTM+VSO+Contour* configuration.

As reference, we also present the RD results of the CTC configuration (*HTM-CTC*), which enables temporal and inter-view prediction in *RefHTM+VSO+Contour* configuration. Figure 5.17 shows that the use of temporal and inter-view predictions on HTM significantly improves its performance, achieving the highest RD performance on depth map coding. This performance difference can be justified by the large amount of redundancy existing between temporal and inter-view frames.

5.6.3 Evaluation of PDC algorithm combined with 3D-HEVC standard

In this subsection we present some experiments that evaluate the potential of PDC algorithm to improve the 3D-HEVC standard. The main idea of these experiments is to investigate the use of PDC algorithm to encode the depth map I-frames of 3D-HEVC, while the compression of P- and B-frames is kept unchanged.

The 3D-HEVC coding configuration proposed in CTC, only uses I-frames in the first view of depth map data, with an intra period of 24 frames. For the case of the MPEG test sequences, which have between 200 and 300 frames, this corresponds to a number of I-frames between 9 and 13 for each three-view sequence. In our experiments, we first selected the depth map I-frames and then encoded them using PDC algorithm with VSO enabled. The bitrate of the encoded frames is approximately the same as the one provided by 3D-HEVC standard. However, eventual differences between the 3D-HEVC I-frames and the PDC ones were properly accounted in the final results. Once the I-frames were encoded, we compressed the MVD contents using a modified version of 3D-HEVC which replaced the depth map I-frames by the PDC ones, during the encoding procedure. The reconstructed I-frames were stored in proper buffer structures of 3D-HEVC standard in order to be used as reference frames by inter coding methods.

As previously discussed, since PDC only uses intra coding tools, the inter-component dependencies cannot be exploited for the compression of I-frames, as done in original 3D-HEVC standard using the Contour mode. This is a missing feature of PDC algorithm, which has been designed as standalone algorithm for depth map coding and does not have access to the encoded texture video views for exploiting inter-component dependencies.

The performance of 3D-HEVC using the PDC encoded I-frames, was compared with two different encoder configurations of HTM software, referred to as *RefCTC* and *RefCTC_withoutContour*. The difference between these HTM configurations is that *RefCTC_withoutContour* disables inter-component prediction in depth map I-frames, by turning off the Contour mode. The idea of *RefCTC_withoutContour* configuration is to make a fair comparison with the modified 3D-HEVC encoder, which cannot exploit inter-component dependencies for I-frames.

The used coding configurations are based on the reference CTC configuration file present in HTM-13.1 reference software, with a slight change on VSO configuration string. This change forces the VSO to use the original texture video views in the synthesis procedure performed during the encoder optimisation loop, even when the reconstructed texture video is available. This was done because the PDC algorithm only can use the original texture video views for VSO process. Thus, this configuration change keeps VSO performing similarly in both 3D-HEVC and PDC algorithms.

Figure 5.18 presents the BDRATE results for depth map coding considering the average PSNR quality of the six synthesised views, and the sum of bitrate used to encode the three depth map views. One may observe that, even without exploiting inter-component dependencies at depth map I-frames, the 3D-HEVC using PDC algorithm presents a superior RD performance than the state-of-the-art 3D-HEVC using *RefCTC* configuration, with average bitrate savings of 1.85%. However, these performance gains are not very expressive, because there are few I-frames in the CTC configuration. Bitrate savings can be observed for all test sequences, except for the Undo Dancer sequence. The results of this particular sequence can be justified by the fact that the Contour method, enabled in *RefCTC* configuration, has an significant

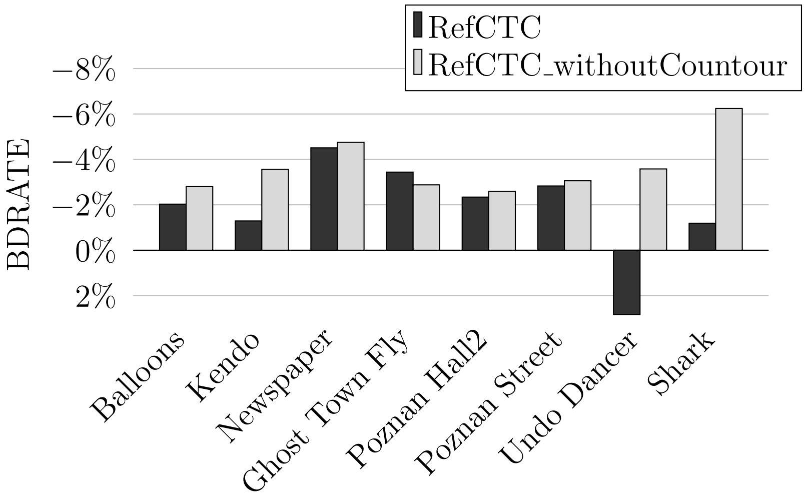


Figure 5.18: Average BDRATE results of the modified 3D-HEVC (using the PDC encoded I-frames) relative to the 3D-HEVC *RefCTC* and *RefCTC_withoutContour* configurations.

influence in the RD performance.

The comparison with *RefCTC_withoutContour* shows that the presented solution is consistently superior to 3D-HEVC algorithm when Contour mode is disabled in both encoders, presenting an average bitrate saving of 3.68%. These results suggest that implementation of inter-component prediction in PDC algorithm may be worth considering, in order to improve its overall coding performance relative to the state-of-the-art 3D-HEVC *RefCTC* configuration.

5.7 Conclusions

In this chapter we presented the PDC algorithm, a complete intra-based encoder for efficient depth map coding, intended for 3D video based applications. This work was mainly motivated by the flexible intra prediction framework used in MMP algorithm, which has presented interesting results for depth map coding. In Section 5.1 we described the main algorithms proposed in literature for the compression of depth maps, in particular the intra depth map coding methods used in the recent 3D-HEVC standard. Section 5.2 presented the first research work of this chapter, which resulted in a preliminary version of the proposed algorithm for depth map coding. We have shown that combining the flexible block partitioning scheme with directional prediction and linear residue fitting provides an efficient representation of depth maps when compared with other methods, such as Platelet algorithm. Despite being based on the MMP, the developed algorithm provided significant reduction of the encoding times, mostly due to the lower computational complexity of the proposed residue coding method.

The architecture of the PDC algorithm developed in this work was described in Section 5.3 and its techniques were detailed in Section 5.4. Unlike the state-of-the-art 3D-HEVC algorithm, the presented approach uses alternative coding techniques which do not rely on the transform-coding paradigm. PDC algorithm exploits an improved flexible block partitioning scheme with an intra predictive framework that uses directional prediction and a novel constrained depth modelling mode. For some blocks, the residual signal can be encoded using a simple linear-fitting method.

The experimental results presented in Section 5.6 demonstrated the superior RD performance of the PDC algorithm when compared to the state-of-the-art 3D-HEVC standard for depth map coding. The intra coding methods of PDC algorithm presented a reduced computational complexity, saving about 25% of the encoding time in relation to 3D-HEVC standard. The use of PDC algorithm to encode the I-frames in 3D-HEVC standard demonstrated the potential of the proposed intra coding solution to improve the current state-of-the-art depth map encoder.

Chapter 6

Sparse representation methods for image prediction

In this chapter, we investigate an alternative class of prediction methods for intra image coding based on sparse representation. The main sparse representation techniques proposed in literature for image prediction were previously described in Section 2.5. These methods included the straightforward Template-Matching (TM) algorithm and more elaborated approaches based on iterative greedy algorithms, such as the Matching Pursuit (MP) [37], and dimensionality reduction methods, in particular the Locally Linear Embedding (LLE) based prediction method.

These algorithms aim to approximate the unknown block as a linear combination of few patches extracted from an adaptive dictionary, that is formed by texture patches derived from the previously encoded area. Due to their features, sparse-representation methods tend to perform better in the presence of complex textures with repeated patterns. The sparse characteristic of these methods imposes a reduced number of linearly combined patches, providing a representative model that performs more efficiently for prediction purposes.

As most of these methods have been proposed for the H.264/AVC algorithm, we started investigating dimensionality reduction methods for the most recent compression standard. In Section 6.1, we evaluate the prediction capabilities of LLE method in HEVC standard, in particular for 3D holoscopic image coding. The presented results demonstrate the advantage of the LLE-based prediction over other existing approaches to exploit the high redundant structure of 3D holoscopic images, which consist of a regularly spaced array of micro-images.

An improved sparse linear prediction method is also presented in Section 6.2. The proposed method can be seen as a generalisation of existing linear prediction and sparse representation prediction methods. Unlike most LSP-based algorithms presented in literature, which use a fixed filter context limited to a small set of closer neighbouring pixels, we propose a linear prediction method that may use a sparse filter context defined in larger causal area. Sparsity restrictions are imposed to the proposed linear prediction method by limiting the number of non-null coefficients in the filter context, using a similar procedure to the one of the LLE-based prediction method. In Section 6.3, the proposed sparse linear prediction method is evaluated within HEVC standard. Experimental tests demonstrate the advantage of the developed method relative to other prediction solutions, namely LLE, TM and LSP.

6.1 3D holoscopic image coding using LLE-based prediction

The main purpose of LLE-based prediction method is to estimate the unknown block by using a linear combination of k-Nearest Neighbour (k-NN) patches, determined in a causal reconstructed area of the image. As described in Subsection 2.5.4, the LLE method uses an implicit approach that avoids the transmission of the linear predictors [38]. In this approach, LLE-based prediction learns the optimal linear combination of the k-NN patches that best approximate a template patch defined in the causal neighbourhood of the unknown block and uses the same estimated model to predict the unknown block.

The ability of LLE-based prediction method to exploit the information associated to the causal area is important for the compression of image contents that present high spatial redundancy. Its advantage for the compression of natural images has been shown in [38], where it was used within H.264/AVC standard.The major gains were achieved for images with repeated patterns and complex textures.

In this context, we investigated the potential of LLE-based prediction method to improve the RD performance of the recent HEVC standard for the compression of 3D holoscopic images. These signals are formed by a 2D array of micro-images which are characterised by strong similarities between them. Due to this fact, 3D holoscopic images present a high spatial redundancy that can be exploited by the LLE-based prediction as well as other related methods.

The current state-of-the-art solution proposed in literature for the compression of 3D holoscopic images is based on a modified version of HEVC standard that includes an intra block-matching method, known as Self-Similarity (SS) compensated prediction. Similarly to the LLE-based prediction method, the SS uses information from the causal reconstructed neighbourhood to predict the unknown block. However, instead of a linear combination of causal patches, the SS only uses one patch and its position is explicitly transmitted to the decoder, using a vector.

In this section we describe the proposed 3D holoscopic image encoder, specifically the incorporation of the LLE-based prediction in HEVC standard. Some experimental tests that illustrate the advantage of the developed algorithm over the previous proposed approach based on SS method are presented and discussed.

6.1.1 Proposed HEVC encoder using LLE-based prediction

As explained in Subsection 3.3.1, the 3D holoscopic images present a significant spatial redundancy between the micro-images. This kind of non-local spatial correlation (separated at least by one micro-image) can be exploited by the LLE-based intra prediction method, which approximates the unknown block using a linear combination of causal patches that may come from the previously encoded micro-images. Unlike the LLE-based prediction method, the directional prediction modes cannot remove the redundancy between micro-images, because only the neighbouring pixels of the first row and first column along the block margins are used to generate the predicted block. Therefore, one expects that the proposed prediction framework based on LLE method is able to improve the overall prediction accuracy of HEVC for the compression of 3D holoscopic content.

As proposed in [38], the number of used causal patches linearly combined in LLE method may vary. These patches are related to the k number defined for the k-NN method, which allows to define different levels of sparsity. Due to the importance of this method for 3D holoscopic image prediction, we used an advanced solution that tests LLE method using several values of k, selecting and explicitly signalling the one that results in the optimal prediction solution, *i.e.* that produces the lowest prediction error.

In this proposal, the LLE-based prediction method was implemented in HEVC prediction framework by replacing some intra directional prediction modes. The idea is to enable explicit signalling of the optimal number, k_{opt}, of NN predictors used by LLE method, without changing the bitstream syntax of HEVC. In the proposed LLE-based prediction method, the k-NN procedure is tested for eight sparsity levels, specifically for $k = 1, \dots, 8$. Thus, eight directional prediction modes, represented in Figure 6.1 by the dashed lines and bold numbers, were replaced by LLE modes. The replaced modes are uniformly spaced to avoid prejudicing any direction in particular, corresponding to the prediction mode number given by $m = 4k - 1$. Although the whole set of replaced modes could be shifted, for instance starting from mode 4, there is no special motivation for this adjustment, because its influence in RD performance is minimal.

The proposed solution may reduce the performance of directional prediction framework, because less angular modes are available. However, as will be shown in the experiments, the LLE-based prediction method is more important for 3D

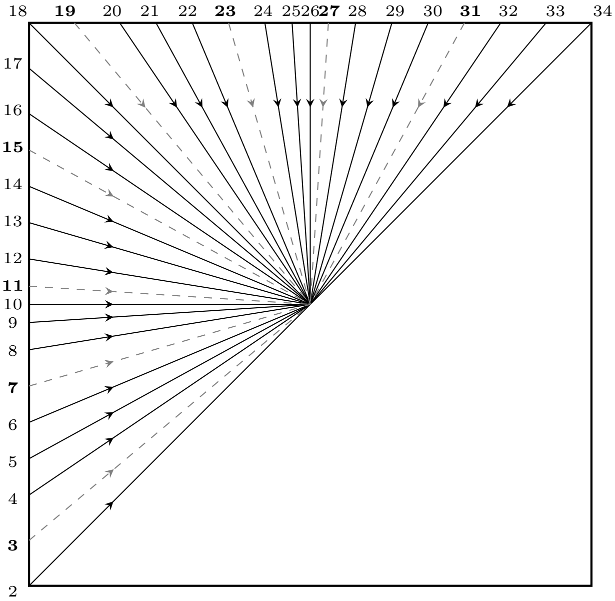


Figure 6.1: Set of HEVC prediction modes replaced by the LLE-based prediction modes, represented by dashed arrows and bold numbers.

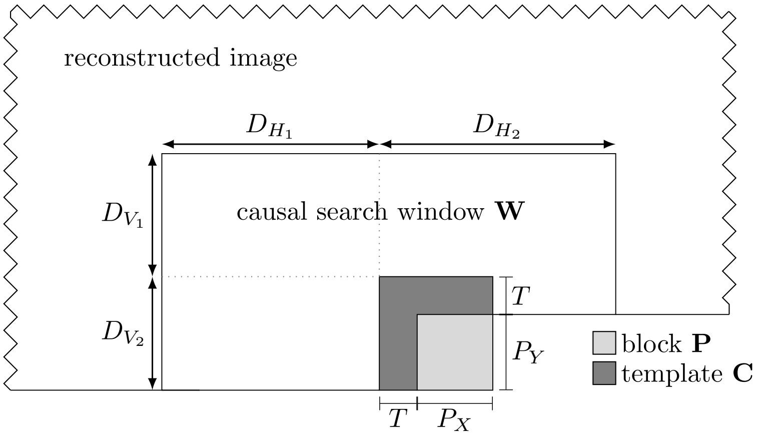


Figure 6.2: Search window used by the proposed LLE-based intra prediction.

holoscopic image coding, compensating the absence of these modes. Furthermore, since there is no need to add new signalling symbols to the HEVC bitstream, this solution simplifies the incorporation of LLE-based prediction into HEVC framework. Note that we only need to transmit the optimal number of NN predictors. The position of the NN predictors and the associated linear coefficients are implicitly derived in both the encoder and decoder sides, based on the template patch defined in the causal reconstructed area.

In order to exploit the redundant information across different micro-images, the size of the causal search window, where the k-NNs are selected, was defined large enough to include at least one micro-image, which for the used test images can reach at most 75×75 pixels. Thus, the size of the causal search window $\mathbf{W}$, represented in Figure 6.2, was set to: $D_{V_1} = 128$, $D_{V_2} = P_Y + T$, $D_{H_1} = 128$ and $D_{H_2} = 64 + P_X + T$.

The size of the template patch used for k-NN search procedure, depends on the unknown block size and dimension T, which has been set to 4. The proposed template size provides a good compromise between using as much neighbouring samples as possible and minimising the inclusion of neighbouring samples that are not correlated with the unknown block.

The optimal block partition and prediction modes are chosen based on the HEVC rate-distortion optimisation procedure. In this procedure, the RMD and MPM methods of original HEVC encoder are used in the same way, although they were originally designed for the directional prediction framework. The proposed mode is also available for chroma prediction, through the luma derived mode, which uses the same mode of luma to predict the corresponding chroma block.

6.1.2 Experimental Results

The performance of the proposed LLE-based prediction framework for 3D holoscopic image coding using HEVC technology (referred to as HEVC+LLE-KNN) was evaluated against the original HEVC standard as well as against the block-matching-based SS compensated prediction presented in [107] (referred to as HEVC+SS). The reference software HM-13.0 [72] of HEVC standard was used as benchmark in the experimental evaluation and also as basis framework to implement the proposed prediction scheme.

Additional experiments were performed using two particular cases of LLE method, based on a fixed number of predictors in k-NN method: the HEVC+LLE-8NN that refers to HEVC enhanced by LLE mode using 8-NN; and the HEVC+LLE-1NN that uses only 1-NN (and corresponding coefficient weight equal to 1), being equivalent to the template matching algorithm. To incorporate these two non-adaptive LLE approaches in HEVC standard, only one intra directional mode (Angular 3) was replaced. The objective of these experiments is to show the advantage of the adaptive LLE-based prediction method relative to the TM algorithm as well as to the non-adaptive approach based on 8-NNs.

Figures A.20 to A.22 of Section A.3 illustrate the four 3D holoscopic test images used in these experiments, namely the frame 0 of Plane and Toy (1920 × 1088), Demichelis Cut (2880 × 1620), Demichelis Spark (2880 × 1620) and Laura (7240 × 5432). These test images were encoded based on the common test conditions defined in [69], using the HEVC intra main configuration, and the recommended QPs: 22, 27, 32 and 37.

Figures 6.3 to 6.6 present the RD results for the tested 3D holoscopic images in terms of luminance PSNR and compression ratio given in bits-per-pixel (bpp). These results show that the proposed coding scheme using LLE-based intra predic-

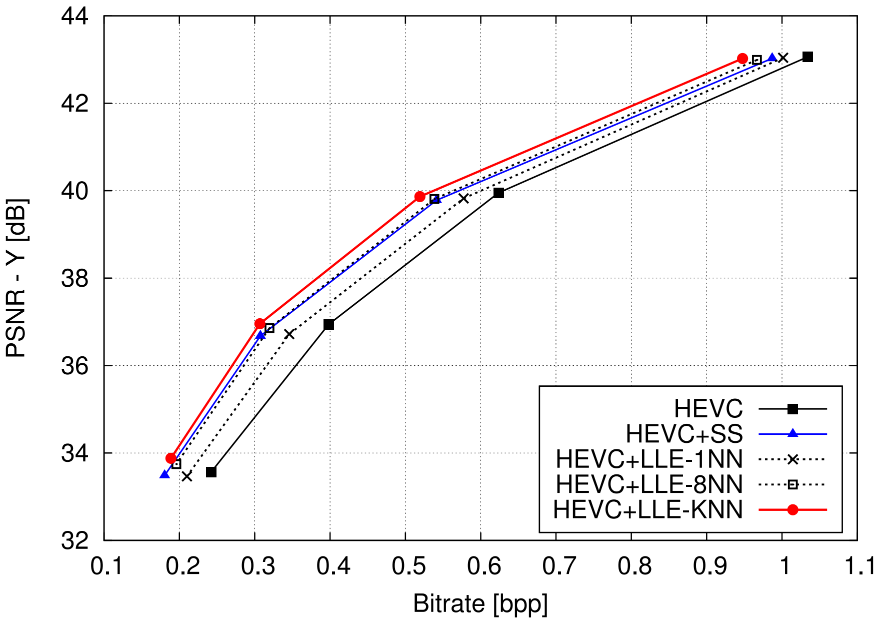


Figure 6.3: Rate-distortion results for Plane and Toy (frame 0) holoscopic image.

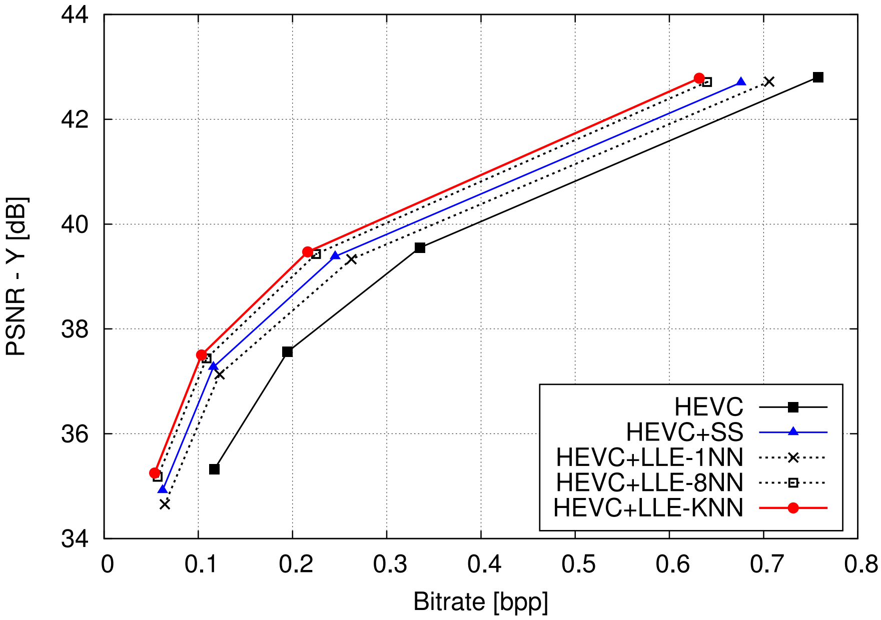


Figure 6.4: Rate-distortion results for Demichelis Cut (frame 0) holoscopic image.

tion method (HEVC+LLE-KNN) is clearly superior than the other tested methods. The lower coding gains are observed for HEVC+LLE-1NN (or template-matching approach), which uses a single predictor. The use of eight predictors in LLE method (HEVC+LLE-8NN) provides a superior RD performance than a single predictor. However, none of these non-adaptive schemes outperformed the adaptive implementation of LLE method (HEVC+LLE-KNN). It can be also observed that HEVC+SS provides significant gains over HEVC, but it is not as efficient as the HEVC+LLE-8NN and the HEVC+LLE-KNN approaches.

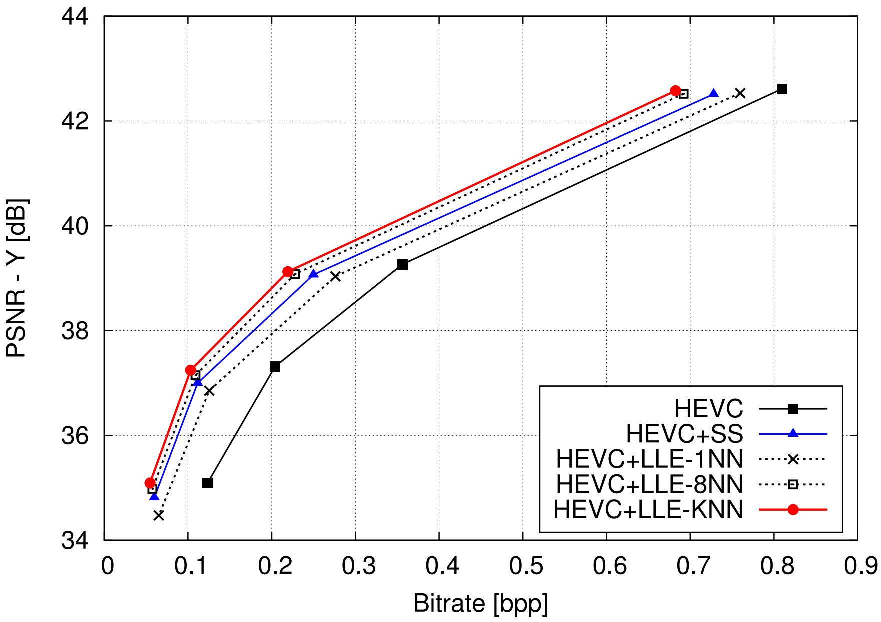


Figure 6.5: Rate-distortion results for Demichelis Spark (frame 0) holoscopic image.

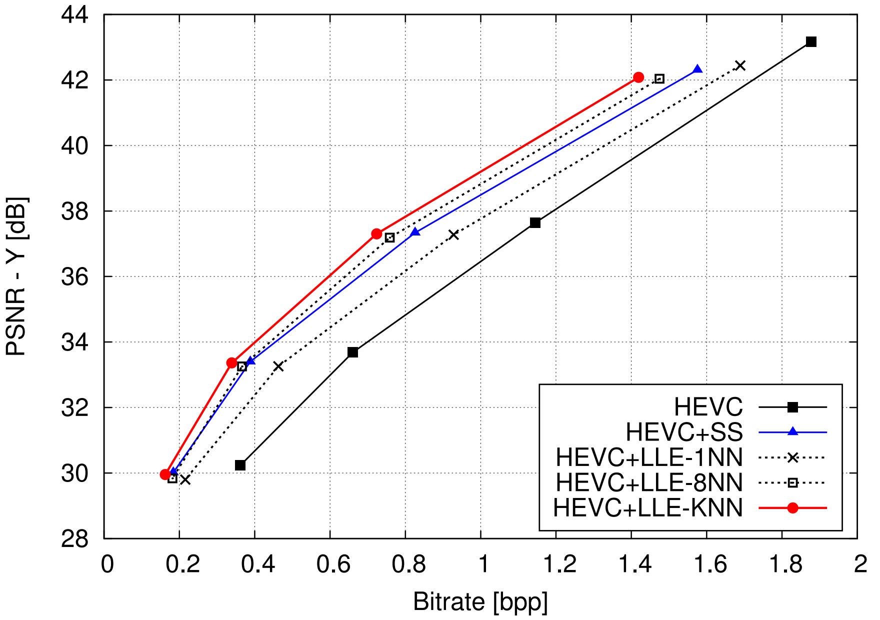


Figure 6.6: Rate-distortion results for Laura holoscopic image.

The Bjontegaard results (BDPSNR and BDRATE) [70] of the main proposed method (HEVC+LLE-KNN) relative to HEVC and HEVC-SS methods are presented in Table 6.1. One may observe that HEVC+LLE-KNN presents an average bitrate saving of 33% when compared to HEVC and 11% when compared to HEVC-SS, specifically proposed for the compression of 3D holoscopic contents.

Regarding the computational complexity aspects, these methods tend to significantly increase the encoding and decoding time of the compression algorithm. This is mainly because the template matching procedure used to select k-NN predictors

Table 6.1: Bjontegaard results of proposed HEVC+LLE-KNN relative to HEVC and HEVC+SS references.

Reference / Image	HEVC		HEVC-SS	
	BDPSNR (dB)	BDRATE (%)	BDPSNR (dB)	BDRATE (%)
Plane and Toys	1.2	-18.5	0.28	-4.9
Demichelis Cut	1.4	-37.4	0.52	-15.3
Demichelis Spark	1.4	-38.5	0.43	-13.5
Laura	3	-37.9	0.67	-11.3
Average	**1.75**	**-33.1**	**0.48**	**-11.3**

needs to be performed in both encoder and decoder sides. Furthermore the estimation of linear coefficients involves solving a linear system (see Subsection 2.5.4). In the proposed implementation of LLE-based prediction method, which did not take into account algorithmic efficient solutions, increased encoding times of one or two orders of magnitude were observed relative to the HEVC standard.

These results demonstrate that prediction approaches based on sparse representation, mostly based on implicit methods, may be more efficient than the explicit block-matching algorithm for exploiting the spatial redundancies present in 3D holoscopic images. The investigated LLE-based prediction method is based on the implicit template matching and least-squares algorithms, requiring a minimal overhead to signal the optimal sparsity level k (between 1 and 8) in contrast to the BMA-based approach that needs to transmit the SS vectors.

These are very interesting results which motivated further research work on sparse representation methods for efficient image prediction. In the following section, we present an improved sparse representation method proposed for intra prediction of generic images.

6.2 The sparse-LSP method for intra prediction

This section describes the proposed method, denominated as sparse-LSP (SLSP), for intra prediction of generic images. Sparse-LSP incorporates the concept of sparse representation into the least-squares prediction algorithm, previously described in Subsection 2.4.2. The idea is to extend the capabilities of LSP algorithm by increasing the size of the linear filter context while keeping a low order model able to learn the representative structural features of the data. The sparse-LSP can also be viewed as a generalisation of the LLE-based prediction method.

The proposed linear model of sparse-LSP is able to exploit not only the linear correlations from the neighbouring samples, as done by traditional LSP algorithms [9], but also from the farther causal reconstructed samples, similarly to LLE method. Traditional linear prediction methods that use a filter context based on the neigh-

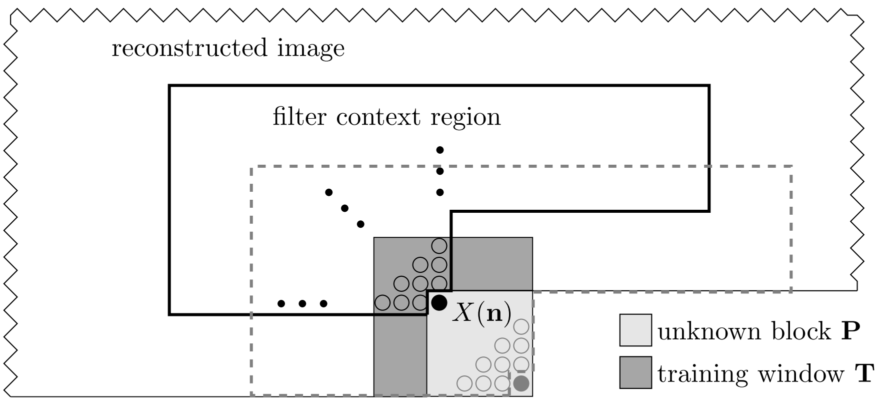


Figure 6.7: Proposed training window and filter context for sparse-LSP algorithm.

bouring samples, as the one illustrated in Figure 2.7 of Section 2.4, provide a good modelling for natural images, that tend to present strong correlations between closer neighbouring samples. On the other hand, the use of larger contexts may be useful when similar textures or objects occur repeatedly along the image signal, as for the case of 3D holoscopic images and some natural images.

6.2.1 Algorithm description

The proposed sparse-LSP method estimates the linear coefficients in a block-by-block basis, which means that the training procedure for coefficient estimation is done once per block, and the same estimated linear model is used to predict all the samples of the unknown block. Instead of using a filter context with fixed shape for the whole image, as used in traditional LSP methods [9], the proposed method employs an adaptive filter context, where most coefficients are null.

The filter context region for the proposed method is illustrated in Figure 6.7. As can be seen, this filter context is much larger than the traditional LSP filter contexts, that use around 10 coefficients. However, in the proposed algorithm, most coefficients are null, due to the imposed sparsity constraint. Since the filter coefficients are estimated once per block, the sparse-LSP method uses a fixed training window **T**, that is defined in the neighbourhood of the unknown block $\mathbb{P}$, similarly to a template area.

The proposed filter context region was designed in such way that all of its positions are available when moving the filter along each sample of the unknown block during the prediction process. In Figure 6.7, the proposed filter context is positioned for both the first sample (represented by a thick black line) and last sample (represented by a dashed gray line) of the unknown block **P**. Note that, as one moves towards the bottom-right corner of the block, some samples to the right of $X(\mathbf{n})$ are not available, as can be seen in Figure 6.7 when predicting the last sample of the block **P** (down-right corner). Due to this fact, only part of the region above $X(\mathbf{n})$ is used by the proposed filter context, in its full extension from the left to right side.

In order to prevent over-fitting issues during the training procedure, *i.e.* the situation in which the estimated models memorise the training data itself rather than learning the underlying linear model, the filter order should not be higher that the number of training samples. Although the illustration of Figure 6.7 suggests that the filter context is larger than the training window **T**, one should notice that only few positions of the filter context will use non-null coefficient, due to the sparse constraint.

In order to achieve a sparse model, *i.e.* to choose the positions of non null coefficients in the proposed filter context region, we employ a searching algorithm based on template matching (TM) procedure (equivalent to k-NN method). The purpose of the searching algorithm is to find the samples of the filter context region which are more similar to the ones of the training window **T**. This can be achieved through the TM algorithm, because the shape of the proposed training window used in the search procedure is equivalent to the template region commonly defined around the unknown block.

A causal search window that corresponds to all positions of the proposed filter context region is used by the TM algorithm. Sparse-LSP uses the TM algorithm to find the k patches most similar to the training window, exactly as done by the k-NN technique in LLE-based prediction method. Although sparse-LSP and LLE methods use the same technique for predictor selection, they differ in relation to the allowed matching points of the search window. Since the LLE method uses the blocks associated to the best matched template patches in order to generate the predicted block, some matching points as the one represented by template patch 2 in Figure 6.8 are not allowed. At such matching point, the associated block is not available, because it partially overlaps the unknown block **P**, as represented by the striped pattern. Thus, all the matching points for which the corresponding block is not fully available are not considered in the LLE method. For the case of the proposed sparse-LSP method, these matching points, including the one of template patch 2 of Figure 6.8, can be selected by the k-NN searching procedure.

The number k of best matched patches (or NNs) estimated by the TM algorithm defines the order of the proposed sparse-LSP filter. The position of each best matched patch relative to the training window is used to define each non null coefficient of the sparse filter context. The coefficients are estimated solving an ordinary least-squares problem, similar to the one described in Subsection 2.4.2. The main difference is the position of the filter coefficients, which can be located anywhere in the filter context region illustrated in Figure 6.7. In the following subsection we describe the mathematical aspects of sparse-LSP and its relation to the LLE-based prediction.

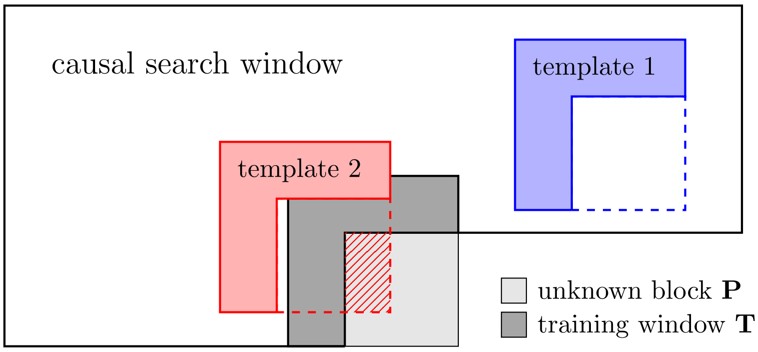


Figure 6.8: Selection of non-null coefficients' positions using TM algorithm (equivalent to k-NN method) within a causal search window.

6.2.2 Mathematical interpretation

Let $X(\mathbf{n})$ denote the image pixel to be linearly predicted, where $\mathbf{n}$ is a two-dimensional vector representing the spatial coordinates in an image. Considering that all the N spatial causal samples of the filter context region presented in Figure 6.7 participate in the prediction process of sparse-LSP, we can write the predicted sample as:

$$\hat{X}(\mathbf{n}) = \sum_{i=1}^{N} a_i X(\mathbf{n} - \mathbf{g}(i)), \tag{6.1}$$

where $\mathbf{g}(i)$ gives the relative position of each sample in the proposed filter context (support), and a_i are the filter coefficients for $i = 1, \dots, N$.

Define the column vector, $\mathbf{t} = [X(\mathbf{n}_0 - \mathbf{h}(1)) \ \dots \ X(\mathbf{n}_0 - \mathbf{h}(M))]^T$, containing the M samples belonging to the training window $\mathbf{T}$, where $\mathbf{n}_0$ denotes the coordinate of the first sample of the predicting block and $\mathbf{h}(j)$ represents the relative position of the samples in the training window, with $j = 1, \dots, M$. Note that, the presented matrix is defined relative to the first sample of the block, $X(\mathbf{n}_0)$, because the training procedure in sparse-LSP is executed once per block, serving to predict the whole block samples.

Consider the following matrix whose rows correspond to the N-sample filter context of sparse-LSP positioned for each sample $\mathbf{h}(j)$ of the training window:

$$\mathbf{V} = \begin{bmatrix} X(\mathbf{n}_0 - \mathbf{h}(1) - \mathbf{g}(1)) \ \dots \ X(\mathbf{n}_0 - \mathbf{h}(1) - \mathbf{g}(N)) \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ X(\mathbf{n}_0 - \mathbf{h}(M) - \mathbf{g}(1)) \dots \ X(\mathbf{n}_0 - \mathbf{h}(M) - \mathbf{g}(N)) \end{bmatrix},$$

where $\mathbf{g}(i)$ is the previously defined relative position of each sample in the filter context.

Each column i of matrix $\mathbf{V}$ corresponds to a displaced version of the training window $\mathbf{T}$ in the causal search window, pointed by $\mathbf{g}(i)$ vector, for $i = 1, \dots, N$.

Based on this observation, vector $\mathbf{t}$ and matrix $\mathbf{V}$ defined for sparse-LSP can be interpreted as the template $\mathbf{b}_c$ and dictionary $\mathbf{A}_c$ defined in Subsection 2.5.4, for the LLE-based prediction method. Actually, matrix $\mathbf{V}$ is constructed similarly to dictionary $\mathbf{A}_c$, based on the patches existing in a predefined causal search window, as illustrated in Figure 6.8. The main difference of sparse-LSP relative to LLE method is that its dictionary (matrix $\mathbf{V}$) is more complete than dictionary $\mathbf{A}_c$ of LLE-based prediction method. As explained in previous subsection, while the matrix $\mathbf{A}_c$ of LLE method does not include those templates whose associated block partially overlaps the unknown block $\mathbf{P}$, the matrix $\mathbf{V}$ of sparse-LSP method contains all the patches (with the shape of the training window) that exist in the search window, without any restriction on the availability of the associated blocks. The only exception is the patch that matches the training window itself (given by vector $\mathbf{t}$), which can not be part of the dictionary (matrix $\mathbf{V}$) since this is the vector to be approximated by the formulated dictionary.

The solution of LSP method, presented in Subsection 2.4.2, is used to find the linear combination of the columns of matrix $\mathbf{V}$ (predictors) that better represents the vector $\mathbf{t}$ (samples of the training window). However, when we consider the sparsity constraint imposed by the sparse-LSP method, the linear combination is restricted to a few columns of matrix $\mathbf{V}$. As previously explained, the selection of the columns of matrix $\mathbf{V}$ that participate in the linear model is done through the TM algorithm (k-NN method). In practice, this procedure consists in comparing the vector $\mathbf{t}$ with all columns of matrix $\mathbf{V}$, and choose the k columns nearest to the vector $\mathbf{t}$, *i.e.* the columns with the lowest matching error.

There are many algorithms proposed in literature to solve linear problems with sparse constraints, providing adaptive solutions closer to the optimal sparse solution. In Chapter 7, some of these methods are investigated to generate sparse linear models. However, in this particular proposal of sparse-LSP, we used the k-NN method for predictor selection, which uses a lower computational complexity, and also allows to generalise the LLE-based prediction method, which can be viewed as a particular case of sparse-LSP method.

Once the predictors (columns of matrix $\mathbf{V}$) were determined using k-NN method, the LSP algorithm proceeds by solving the problem $\min_{\mathbf{a}_k}(\|\mathbf{t} - \mathbf{V}_k\mathbf{a}_k\|_2^2)$, where $\mathbf{V}_k$ is a sub-matrix of $\mathbf{V}$ which only contains the k selected columns, and $\mathbf{a}_k$ is the vector of coefficients. Note that, by using a sub-matrix of $\mathbf{V}$ with the k selected predictors, the solution vector $\mathbf{a}_k$ only applies to that sub-set of predictors, having a total of k coefficients. The columns missing in $\mathbf{V}_k$ are assumed to use a null coefficient in the linear combination. Based on the solution vector $\mathbf{a}_k$, we can build the complete N-sample solution vector $\mathbf{a}$, where the null coefficients correspond to the unused columns of matrix $\mathbf{V}$ and the estimated coefficients (from vector $\mathbf{a}_k$) are attributed to the positions of the selected predictors.

Given the solution vector **a**, the prediction of the sample $\hat{X}(\mathbf{n})$ of the unknown block **P** can be determined by:

$$\hat{X}(\mathbf{n}) = \sum_{i=1}^{N} a_i X(\mathbf{n} - \mathbf{g}(i)) \tag{6.2}$$

where $\mathbf{g}(i)$ gives the relative Euclidean coordinates of the filter context samples, as previously defined.

Similarly to the traditional LSP algorithms, that used a fixed filter order, the sparsity level used in sparse-LSP method can be constant during the whole encoding procedure. This is an implicit procedure, which does not require the transmission of any additional information. Alternatively, an explicit approach, similar to the one used in LLE-based prediction method for 3D holoscopic image coding in Subsection 6.1.1, can be used. Such approach consists in testing several sparsity levels, solving the LSP problem K times, for $k = 1, \ldots, K$, and choosing the solution k_{opt} that gives the lowest prediction error. In this approach the selected sparsity level, k_{opt}, should be transmitted to the decoder.

6.3 Application of sparse-LSP to HEVC standard

As previously discussed, the sparse-LSP method provides higher flexibility, because it may use predictors closer to the unknown sample to predict, similarly to the traditional LSP methods, or use farther predictors, as the LLE-based prediction method. Furthermore, sparse-LSP may select both closer and farther predictors in the same linear model. The use of closer predictors for linear prediction has shown to be important in natural image coding applications, namely using the LSP algorithm in the MMP algorithm [9]. The advantage of farther predictors for the compression of natural images was also demonstrated in literature using the LLE-based prediction in H.264/AVC standard, specially for those images with more complex textures and repeated patterns.

In this section, we show the advantage of the proposed sparse-LSP method for natural image coding, which is able to exploit the properties of both LSP and LLE methods. The sparse-LSP method was implemented in the state-of-the art HEVC standard and compared to the other prediction methods, using the same basis compression technology. In what follows, the main implementation details of the sparse-LSP mode into HEVC standard are described. Experimental results illustrating the advantage of the proposed method are also presented and discussed.

6.3.1 Implementation details

The proposed sparse-LSP prediction method was implemented in HEVC algorithm, being used together with the directional intra prediction framework constituted by planar, DC and 33 angular modes. In order to fairly compare the sparse-LSP with other methods, such as the traditional LSP algorithm, we used the implicit implementation of sparse-LSP, which defines a fixed sparsity level along the encoding procedure. Since these comparisons were made against the LSP implementation proposed in [9], which used a 10th order filter support, we set the sparse-LSP sparsity level to $K = 10$. This means that sparse-LSP always selects 10 predictors from the context filter region, for which it computes non-null linear coefficients.

In order to accommodate the proposed prediction mode in HEVC framework, the directional mode 3 has been replaced by sparse-LSP mode. Thus, no modification to the HEVC bitstream syntax was required. Furthermore, the directional prediction framework of HEVC was almost unaffected, because only one angular mode was replaced. The Lagrangian RD procedure of HEVC encoder, adopted in HM reference software, which is based on the RMD and MPM methods, was used to select the optimal prediction mode for each PU.

As previously explained, the training window and the filter context of sparse-LSP are closely related to the template area and the search window of LLE-based prediction method. The training window includes the left and top neighbouring regions of the unknown block, varying its dimension with the block size. The thickness of the training window was set to $T = 4$, as previously defined for the template area of LLE-based prediction method developed for 3D holoscopic image coding, illustrated in Figure 6.2.

The dimensions of the search window used by the TM algorithm (equivalent to k-NN procedure) to estimate the sparse filter context are given by $D_{V_1} = 64$, $D_{V_2} = P_Y + T$, $D_{H_1} = 64$ and $D_{H_2} = P_X + T$, as represented in Figure 6.2 for the LLE-based prediction method. These values indicate the size of the search window that corresponds to the filter context region of Figure 6.7, where the sparse filter predictors associated to non-null filter coefficients are selected.

Note that, the size of the search window proposed for these experiments is smaller than the one proposed for 3D holoscopic image coding using LLE-based prediction. This is because we are designing sparse-LSP for the compression of natural images and we do not need to meet the previous requirement that imposes a search window large enough to include at least one micro-image of 3D holoscopic contents. The proposed search window has a reasonable size, which allows to exploit repeated patterns in the natural images with a period up to 64 samples.

Table 6.2: BDPSNR and BDRATE results of HEVC using LSP, TM, LLE and SLSP methods relative to the reference HEVC standard, for the selected set of test images.

	LSP		**TM**		**LLE**		**SLSP**	
Image	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
Barbara	0,23	-3,37	0,12	-1,86	0,27	-3,99	**0,43**	**-6,27**
Barbara2	0,02	-0,31	0,07	-1,13	0,20	-3,13	**0,27**	**-4,05**
Snook	0,01	-0,10	0,28	-3,83	1,00	-13,64	**1,05**	**-14,31**
Wool	0,13	-2,25	0,33	-5,70	0,56	-9,49	**0,62**	**-10,38**
Bike	0,15	-1,69	0,49	-5,59	0,45	-5,24	**0,51**	**-5,85**
Pan0_qcif	0,00	0,06	0,36	-5,25	0,92	-13,27	**1,12**	**-15,73**
Roof	0,07	-0,92	0,12	-1,61	0,33	-4,34	**0,44**	**-5,80**
Houses	0,03	-0,41	0,24	-2,98	0,34	-4,21	**0,39**	**-4,72**
Average	0,08	-1,12	0,25	-3,49	0,51	-7,16	**0,60**	**-8,39**

6.3.2 Experimental results

The proposed modified version of HEVC incorporating the sparse-LSP mode is based on the reference software HM-13.0 [72], being denominated by HEVC-SLSP. In order to assess its performance relative to other methods, we created new versions of HEVC algorithm using those methods. The LSP mode, as proposed in [9], based on a 10th order filter, was implemented in HEVC standard, being referred to as HEVC-LSP. Another version of HEVC standard used in these experiments incorporates the LLE-based prediction mode [38] as presented in Subsection 6.1, with a fixed number of predictors, being referred to HEVC-LLE. As the TM algorithm is a particular case of sparse prediction methods, which provides the sparsest representation using only one dictionary element, it has been also used in the comparisons. The version of HEVC algorithm improved by TM algorithm is denominated by HEVC-TM.

Similarly to the proposed HEVC-SLSP algorithm based on sparse-LSP, the LSP, LLE and TM methods compared in these experiments are combined with the directional intra prediction framework of HEVC standard, by replacing the angular 3 mode. For the case of LLE and TM methods, the size of the template area (see Figure 6.2) was set to $T = 4$, as the training window of the sparse-LSP method. The size of the search window for these methods (see Figure 6.2) was defined equal to the one of sparse-LSP method, presented in previous subsection. The sparsity parameter (k-NNs) for LLE-based prediction method was set to $k = 10$, corresponding to the same model order used in sparse-LSP and LSP methods. The used configurations for TM, LLE and sparse-LSP methods ensure that these methods exploit the same causal information, differing only by the procedure used to generate the predicted block, being important for a fair comparison.

Experiments were performed for two sets of sequences. The first one, illustrated

Table 6.3: BDPSNR and BDRATE results of HEVC using LSP, TM, LLE and SLSP methods relative to the reference HEVC standard, for the first frame of HEVC test sequences.

	LSP		**TM**		**LLE**		**SLSP**	
Image	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
(E) Johnny	-0,01	0,21	0,09	-2,31	**0,11**	**-2,53**	0,08	-1,95
(E) KristenAndSara	-0,01	0,17	**0,09**	**-1,69**	0,08	-1,54	0,09	-1,65
(E) FourPeople	0,01	-0,14	0,05	-0,84	0,05	-0,79	**0,05**	**-0,89**
Average	0,00	0,08	0,08	-1,61	**0,08**	**-1,62**	0,07	-1,50
(D) BasketballPass	0,03	-0,41	0,08	-1,17	0,03	-0,52	**0,08**	**-1,20**
(D) BlowingBubbles	0,01	-0,05	0,01	-0,05	0,00	-0,03	**0,01**	**-0,17**
(D) BQSquare	0,00	0,05	0,03	-0,35	0,01	-0,09	**0,04**	**-0,40**
(D) RaceHorses	-0,02	0,15	-0,02	0,21	-0,03	0,38	**0,00**	**0,00**
Average	0,00	-0,06	0,02	-0,34	0,00	-0,07	**0,03**	**-0,44**
(C) BQMall	0,00	-0,05	0,04	-0,68	0,04	-0,72	**0,05**	**-0,78**
(C) BasketballDrill	0,02	-0,50	0,13	-2,61	0,12	-2,51	**0,14**	**-2,96**
(C) PartyScene	0,01	-0,10	0,04	-0,49	0,05	-0,57	**0,05**	**-0,62**
(C) RaceHorses	0,01	-0,15	0,01	-0,24	0,00	-0,05	**0,01**	**-0,24**
Average	0,01	-0,20	0,06	-1,01	0,05	-0,96	**0,06**	**-1,15**
(B) BasketballDrive	0,01	-0,26	0,06	-2,39	**0,07**	**-2,99**	0,06	-2,16
(B) BQTerrace	0,01	-0,17	0,11	-1,73	**0,17**	**-2,69**	0,10	-1,55
(B) Cactus	0,00	-0,03	0,05	-1,43	**0,06**	**-1,55**	0,05	-1,35
(B) Kimono1	0,00	-0,01	0,00	-0,06	**0,01**	**-0,18**	0,01	-0,14
(B) ParkScene	0,00	-0,03	0,01	-0,18	0,01	-0,20	**0,01**	**-0,27**
Average	0,00	-0,10	0,05	-1,16	**0,06**	**-1,52**	0,04	-1,09

in Figures A.12 to A.17 of Section A.2, includes a selected set of test images with complex textures and repeated patterns, which are not efficiently predicted by traditional directional modes. The second set includes the first frame of HEVC test sequences from four classes as proposed in [69], namely the classes B (1920 × 1080), C (832 × 480), D (416 × 240) and E (1280 × 720). Only the luminance channel was encoded and evaluated in these experiments.

Tables 6.2 and 6.3 present the Bjontegaard results [70] (BDPSNR and BDRATE) using the proposed HEVC-SLSP, based on sparse-LSP method, as well as HEVC-LLE, HEVC-TM and HEVC-LSP algorithms, for the compression of the two selected test sets, respectively. The Bjontegaard results were computed relatively to the original HEVC standard. The results for the first test set, made of natural images with complex textures, presented in Table 6.2, show that all the evaluated methods provide significant improvements to HEVC standard. These methods tend to perform better in the presence of complex textures with repeated patterns, due to limitations of directional intra prediction framework for representing these features. Among all the evaluated methods, the sparse-LSP provides the higher coding gains, achieving an average bitrate saving above 8% in relation to the original HEVC standard. The performance of HEVC-LLE is slightly inferior than the one of HEVC-SLSP, providing about 7% of average bitrate savings. The superior performance of sparse-LSP can be justified by its ability to exploit the same predictors of LLE and the predictors closer to the sample to predict. The HEVC-TM, which only uses one predictor, and the HEVC-LSP, based on traditional LSP method, achieve significantly inferior average bitrate savings of about 3.5% and 1%, respectively. The low performance of LSP algorithm is due to its limitations to exploit textured areas with repeated patterns that present a period larger than the LSP filter support.

The results for the first frame of HEVC test sequences, presented in Table 6.3, are less relevant than the ones of the first test set. This is mainly because these images tend to be smoother or have less textured areas with repeated patterns, and hence directional prediction is able to provide an efficient prediction for most cases. The HEVC-LSP method presents the lowest performance, providing average bitrate savings quite inferior to 1% for all test sequences. When using TM, LLE or sparse-LSP algorithms, HEVC shows a more efficient performance, achieving bitrate savings close to 3% for some test sequences. The HEVC-SLSP approach tends to present the best RD performance for most cases, mainly for classes D and C. However, the HEVC-LLE shows to perform better for most Full-HD test sequences (class B) as well as for the Johnny sequence of class E with HD resolution. This higher performance of LLE-based prediction method for most high resolution images can be justified by the larger amount of samples in these signals, which tends to present spatial dependencies between more distant samples. Therefore, the ability of sparse-LSP to select closer predictors, may not provide a significant advantage for some high resolution images. This observation agrees, to some extent, with the results of HEVC-LSP, which uses closer neighbouring predictors and provides a poor performance for these images.

In Figures 6.9 and 6.10, the RD curves of two selected test images, namely Barbara and Wool, are presented. The results for both images show the advantage of HEVC using the sparse-LSP method over the remaining methods, from low to high bitrates. In the case of Barbara image, the performance of HEVC-LSP and HEVC-LLE algorithms is quite similar, being significantly superior to the original HEVC standard. The gains of LSP method in HEVC standard for the Barbara image agree with previous observations reported in literature using the LSP method in MMP algorithm [9].

The results for the Wool image, in Figure 6.10, show that HEVC-LLE performs quite similar to HEVC-SLSP, mainly at lower bitrates. However, the sparse-LSP approach tends to slightly improve at higher bitrates. The RD performance of

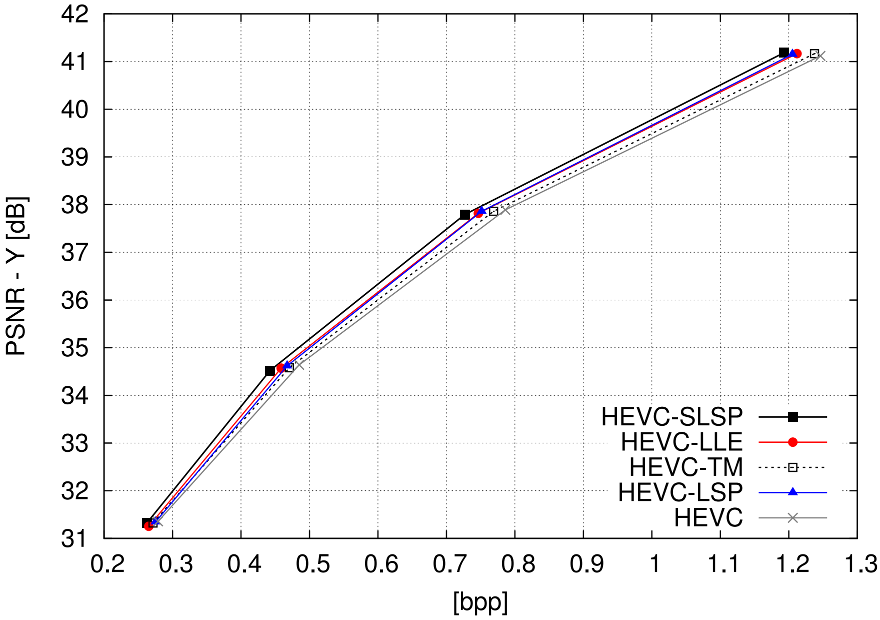


Figure 6.9: Rate-distortion curves comparing SLSP, LLE, TM and LSP methods using HEVC standard, for Barbara image.

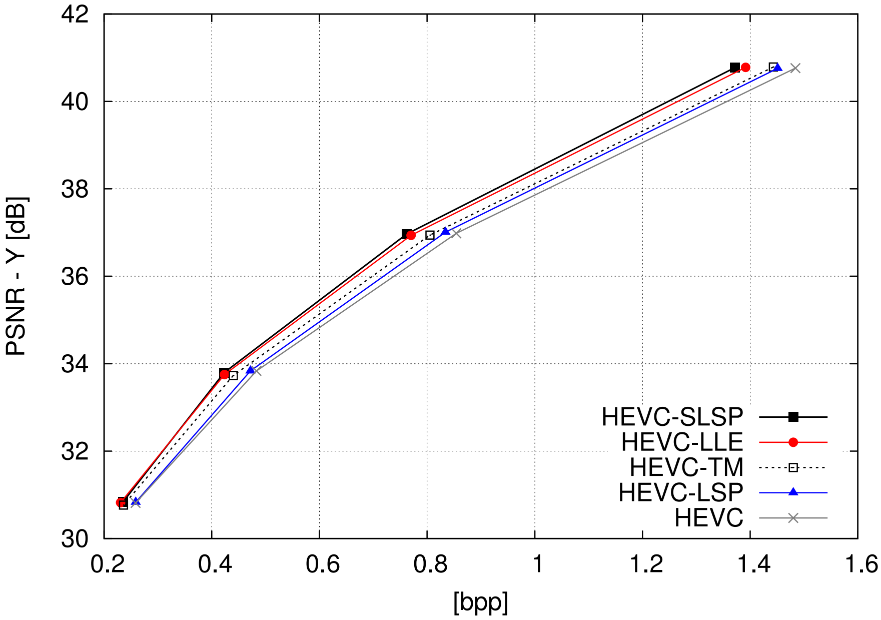


Figure 6.10: Rate-distortion curves comparing SLSP, LLE, TM and LSP methods using HEVC standard, for Wool image.

HEVC-TM is smaller than the one of HEVC-SLSP and HEVC-LLE methods, but its curve is consistently above original HEVC from low to high bitrates. In the case of HEVC-LSP, the achieved gains are almost null at lower bitrates, increasing its performance for higher bitrates. The poor performance of LSP method at lower bitrates can be justified by the higher quantisation effects in the reconstructed causal image at those bitrates, which decreases the learning capabilities of LSP method when using a small filter support.

In general, the presented results reveal that the coding performance of the evaluated methods highly depends on the type of encoded images. Due to their characteristics, images of the first test set tend to result in higher coding gains, specially when using the sparse-LSP method. The main advantage of sparse-LSP method is its ability to generalise the remaining evaluated methods. With simple restrictions on the size of the filter context region and model order, it is possible to transform the sparse-LSP method to operate similarly to TM, LLE or LSP methods. An adaptive selection of those restrictions during the encoding procedure allows to optimise the prediction results for different image characteristics.

In regard to the computational complexity, we observed that coding times of the evaluated encoding algorithms are about one order of magnitude higher when compared to the original HEVC coding times. This is mainly due to the TM searching procedures and least-squares optimisation methods used by the investigated prediction techniques. With exception of TM algorithm, which run uses less computations, the remaining evaluated prediction methods presented a comparable computational complexity. In this work, we did not consider efficient algorithmic implementations for the investigated methods. Thus, we believe that computational complexity problem could be alleviated, either by using more efficient implementations, or by exploiting parallel processing architectures.

6.4 Conclusions

In this chapter, methods based on sparse representation have been investigated for efficient image prediction. In Section 6.1, the Locally Linear Embedding-based prediction has been investigated for 3D holoscopic image coding using the HEVC technology. The working principle of LLE-based prediction method consists on the linear combination of several causal reconstructed patches, which makes it an efficient solution to exploit correlations from the previously encoded micro-images. Experimental results demonstrated the advantage of LLE-based prediction method to exploit the spatial redundancy that characterises the 3D holoscopic images, when compared to the explicit Self-Similarity approach based on intra BMA.

In Section 6.2, a new least-squares-based prediction mode for intra image coding has been investigated. By combining sparse representation with LSP method, an improved prediction method that generalises LLE-based prediction method was developed. The proposed algorithm derives a sparse context for linear prediction using the k-NN algorithm in a large causal search window. The linear coefficients of the selected predictors are estimated by least-squares optimisation, and the estimated linear model is used to predict all the samples of the unknown block.

The proposed method, denominated as sparse-LSP, was implemented in the HEVC standard, and evaluated for still image coding. Section 6.3 discussed the experimental results of sparse-LSP method, showing its superior coding performance for most test images when compared to the other related methods presented in literature, such as the LLE-based intra prediction, the traditional LSP and the TM algorithms. The advantage of sparse-LSP over the LLE-based prediction method results from its higher adaptation, as it may use a more flexible filter context considering predictors from the closer spatial neighbourhood, like traditional LSP filters, as well as more distant predictors, similarly to the LLE-based prediction method.

Chapter 7

Generalised optimal sparse predictors

The main research work presented in the previous chapters of this thesis was focused on two main intra prediction methods, specifically the directional prediction, investigated in Chapters 4 and 5, and the linear prediction using sparse models, investigated in Chapter 6. A common characteristic among these methods is the use of the previously encoded samples in the neighbourhood of the unknown block for the prediction process. However, while the directional prediction only uses information from the closer neighbouring samples, namely the top row and left column adjacent to the block, the linear prediction methods investigated in this thesis exploit information from a larger causal region, being able to use reconstructed samples at a longer distance. Due to their characteristics, the prediction capabilities of these methods tend to depend on the image contents. This has been observed in Chapter 6, where the sparse linear prediction methods achieved better prediction performance in the presence of complex textures and repeated patterns, than the directional prediction methods used by HEVC standard.

In this chapter, we present a new linear prediction framework for the HEVC standard, which generalises the previously investigated directional and linear prediction methods, using optimal sparse predictors combined with geometric transformations. Ultimately, each angular mode of directional prediction can be regarded as a set of very simple linear predictors, a different one for each pixel of the unknown block. However, such an interpretation of each directional mode is too poor to provide useful insights for their design. In Section 7.1 we present a new interpretation for directional prediction as a first-order linear prediction method combined with geometric transformations. Such interpretation of directional prediction motivated the proposed generalised intra prediction framework based on sparse models, as presented in Section 7.2.

In Section 7.3 we describe several algorithms that can be used to estimate the proposed sparse models. We show that sparse models estimated by traditional Orthogonal Matching Pursuit (OMP) method are less efficient for prediction purposes, mainly due to the greedy characteristic of OMP selecting the non-null predictors. Conversely, we demonstrate that not-so-greedy approaches, such as Least-Angle Regression (LAR) and Least Absolute Shrinkage and Selection Operator (LASSO) methods, are able to provide better prediction performance. We also investigate different stopping criteria for the optimisation algorithms used for model estimation.

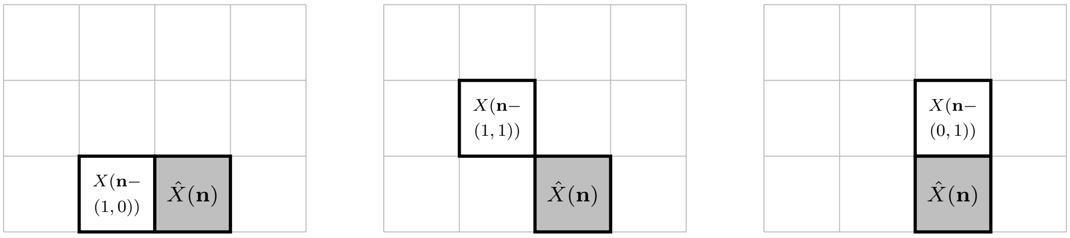
Section 7.4 describes the proposed prediction algorithm, based on generalised optimal sparse predictors for HEVC standard. Several experimental results are presented and discussed in Section 7.5, demonstrating the advantage of the proposed predictive solution in relation to the directional prediction framework of HEVC, as well as other linear prediction methods previously investigated.

7.1 Two-stage interpretation of directional prediction

In its procedure, the angular modes propagate the causal neighbouring pixels from the left column and top row adjacent to the unknown block into the target block in some specific direction. In practice, this can be viewed as predicting pixels with a linear combination of causal samples. In this section, we analyse the sample propagation of HEVC angular modes and present an alternative two-stage interpretation for their design, based on the concepts of linear prediction and geometric transformations.

An angular mode can be easily written as a set of linear predictors, whose filter contexts and weighting coefficients depend on the predicted sample position and mode direction. This interpretation, however, has some issues related to the dependency existing between the filter context and the predicted sample position. Such a dependency results in a more complex linear model formulation, which does not provide valuable insights for designing a generalised linear prediction method. This has motivated an alternative interpretation of angular modes, which at a first-stage only considers three simple linear models associated to the Angular 10 (horizontal), Angular 18 (diagonal) and Angular 26 (vertical) modes. These angular modes have a common characteristic: under the linear prediction interpretation they may be reproduced using a filter with fixed context.

Unlike most angular modes, that use a linear combination of two reference samples for prediction of each sample, the three referred angular modes simply copy the reference neighbouring samples along the block, in either horizontal, diagonal or vertical directions. Such copy of reference samples can be performed by first



(a) Angular 10 - F_H^1 (b) Angular 18 - F_D^1 (c) Angular 26 - F_V^1

Figure 7.1: Sample propagation using a first-order linear filter context for Angular 10, 18 and 26 modes.

Table 7.1: Linear filters and associated geometric transformations.

Filter	Neighbourhood	Transform	Angular modes
F_H^1	left, left-down	$T_H(d)$	2 to 10
F_D^1	left, top-left, top	$T_{Dh}(d)$ / $T_{Dv}(d)$	11 to 18 / 18 to 25
F_V^1	top, top-right	$T_V(d)$	26 to 34

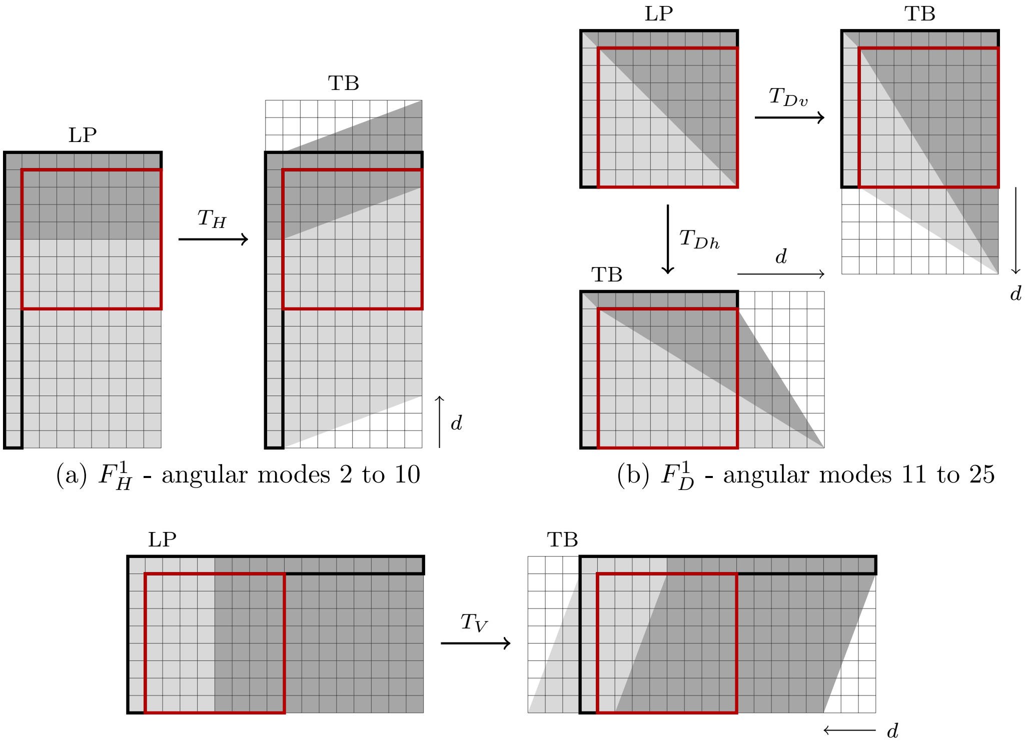
order linear filters, which only depend on the mode direction and use a weighting coefficient equal to 1.

Let $X(\mathbf{n})$ denote the image pixel to be linearly predicted, where $\mathbf{n}$ is a two-dimensional vector with the spatial coordinates in the image. The referred first-order linear models can be written as:

$$\hat{X}(\mathbf{n}) = X(\mathbf{n} - \mathbf{g})\,, \tag{7.1}$$

where the relative position of the first order filter, $\mathbf{g}$, is $(1,0)$, $(1,1)$ or $(0,1)$, for Angular 10, 18 and 26 modes, respectively. These filter contexts are illustrated in Figure 7.1, being designated by F_H^1, F_D^1 and F_V^1. The gray coloured samples represent the target prediction sample.

In order to reproduce the remaining angular modes of HEVC encoder, we propose these first-order linear filters (first-stage) with an additional processing stage based on geometric transformations (second-stage). The idea is to distort the linearly predicted block, so that all angular modes can be reproduced from the three linear filters from Figure 7.1. Different sets of geometric transformations are available, depending on the used linear filter, as indicated in Table 7.1. For instance, in the case of filter F_H^1, which only uses the left and left-down neighbouring regions for prediction (see available regions in Figure 2.4 of Subsection 2.3.1), the geometric transformation $T_H(d)$ is used to reproduce any angular mode between 2 and 10, controlled by parameter d. For linear filter F_D^1, two sets of geometric transformations are defined, namely $T_{Dh}(d)$ and $T_{Dv}(d)$, which generate angular modes 11 to 18 and



(a) F_H^1 - angular modes 2 to 10

(b) F_D^1 - angular modes 11 to 25

(c) F_V^1 - angular modes 26 to 34

Figure 7.2: Geometric transformations applied to linearly predicted blocks in order to reproduce HEVC angular modes.

angular modes 18 to 25, respectively.

The proposed geometric transformations are affine, *i.e.* they preserve collinearity and distance ratios, and parameter d is used to control the magnitude of transformation. When $d = 0$, there is no transformation on the output block (corresponding to the Angular 10, 18 or 26 modes). The proposed transformations that generate all the HEVC angular modes are illustrated in Figure 7.2. For each linear filter, the linearly predicted output (labelled by LP) and the transformed block (labelled by TB) are shown. For clarity, a representative edge is visible in the illustrated blocks.

The $T_H(d)$ and $T_V(d)$ transformations correspond to vertical and horizontal block skews, respectively. As observed in Figure 7.2, the straight lines generated by horizontal and vertical first-order linear filters (F_H^1 and F_V^1) remain straight and also parallel after the block skew transformation. In practice, the $T_H(d)$ (or $T_V(d)$) transformation simply displaces each vertical line (or horizontal line) of the block by an amount proportional to its distance to the horizontal block margin (or vertical block margin). These transformations can be described as:

$$T_H(d) = \begin{bmatrix} 1 & 0 & 0 \\ d & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_V(d) = \begin{bmatrix} 1 & -d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.2}$$

Table 7.2: Transformation matrix parameter d and respective angular modes generated by $T_H(d)$ and $T_V(d)$ transformations.

Value of d	0	2	5	9	13	17	21	26	32
$T_H(d)$ modes	10	9	8	7	6	5	4	3	2
$T_V(d)$ modes	26	27	28	29	30	31	32	33	34

Table 7.3: Transformation matrix parameter d and respective angular modes generated by $T_{Dh}(d)$ and $T_{Dv}(d)$ transformations.

Value of d	4096	1638	910	630	482	390	315	256
$T_{Dh}(d)$ modes	11	12	13	14	15	16	17	18
$T_{Dv}(d)$ modes	25	24	23	22	21	20	19	18

where d is the parameter that defines the amount of skewing. Note that d is negative for $T_V(d)$ transformation because the skewing is applied to the left direction. To ensure that the transformed block fills all samples of the target block, the linear filter step propagates both the left and left-down block neighbourhoods in case of F_H^1, and both the top and top-right block neighbourhoods in case of F_V^1, as illustrated in Figure 7.2. Considering that block dimensions are normalised to 32×32 (the larger block size where prediction is derived), the values of d that correspond to HEVC angular modes, using T_H and T_V transformations, are given in Table 7.2.

Regarding $T_{Dh}(d)$ and $T_{Dv}(d)$ transformations, a slightly different approach is used. The linearly predicted (LP) block is divided into two triangles, which are independently transformed. These transformations should preserve the continuity between the block and the left and top neighbourhoods, in order to follow the propagation effect of angular modes 11 to 25. In the case of $T_{Dh}(d)$ transformations, the triangle below the block diagonal is scaled ($T_{Dh}^s(d)$), while the triangle above the diagonal is horizontally skewed ($T_{Dh}^k(d)$). These affine transformations are given by:

$$T_{Dh}^s(d) = \begin{bmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_{Dh}^k(d) = \begin{bmatrix} 1 & d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.3}$$

where the parameter d defines the intensity of the transformations.

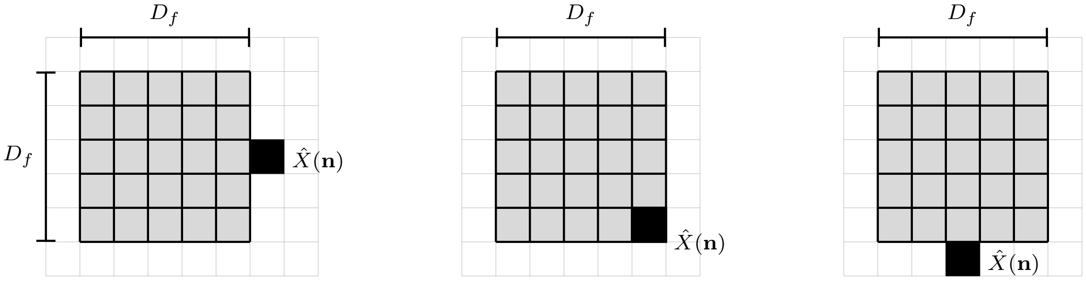
Similarly, $T_{Dv}(d)$ uses scale and skew transformations on the triangles above and below the block diagonal, respectively. The difference is that the transformations are made along the vertical direction, instead of the horizontal one. Considering that block dimensions are normalised to 32×32, the values of d that result in HEVC angular modes, using T_{Dh} and T_{Dv} transformations, are given in Table 7.3. Note that, the proposed model for angular modes 11 to 25 does not produce exactly the same results as HEVC implementation for these modes. This results from small differences on sample interpolation process, which do not compromise the performance of the intended generalisation. Details about implementation of angular modes in HEVC algorithm can be found in [130].

7.2 Generalising directional prediction

The presented two-stage formulation of directional prediction demonstrates that first-order linear prediction models can be used to generate angular intra modes when combined with geometric transformations in a second stage. An interesting extension of this method would consist in replacing the first-order predictors by optimal linear predictors, that would be able to reproduce directional prediction as a particular case. This would require the optimal predictors to exploit a larger causal area than the first order filters. Such observation led us to a generalised intra prediction framework based on optimal sparse predictors.

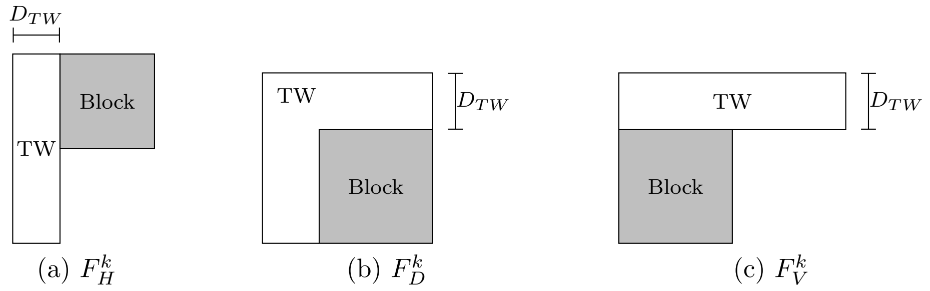
In addition to the first-order filters, that replicate angular prediction modes, the proposed generalised prediction framework uses adaptive high-order linear filters. These filters are adaptively determined in an augmented context area, with squared shape, as shown in Figure 7.3. This means that the three first-order filters F_H^1, F_D^1 and F_V^1 described in Section 7.1 can be replaced by larger filter contexts. In the example of Figure 7.3, the maximum order of linear filters is 25, *i.e.* $D_f = 5$ which results in F_H^{25}, F_D^{24} and F_V^{25}, where superscript indicates maximum filter order.

The block prediction using the F_D^{24} and F_V^{25} filters is applied in raster scan order, from left to right. Due to its shape, the F_H^{25} filter is applied using a vertical scan order, from top to bottom. Note that the samples belonging to the two bottom rows of filter F_H^{25} and the two right columns of filter F_V^{25} may not be available during the block prediction. This is an issue when non-null coefficients are defined in these positions. To solve it, the last available samples in the missing rows and columns are padded along horizontal and vertical directions, respectively.



(a) F_H^{25} (b) F_D^{24} (c) F_V^{25}

Figure 7.3: High-order linear filters for the proposed generalised intra prediction framework.



(a) F_H^k (b) F_D^k (c) F_V^k

Figure 7.4: Training window regions associated to the three types of linear filters used in the proposed intra prediction framework.

The use of three independent filters is motivated by the geometrical transformation-based interpretation of directional intra prediction previously presented. These filters allow to exploit different block neighbouring regions, namely the left, left-top, and top regions. Unlike the first-order filters, that use a fixed coefficient equal to 1 to reproduce angular modes (see Equation 7.1), the proposed augmented filters are locally optimised in a causal training window (TW), based on least-squares regression with a sparsity constraint. The estimated linear model is used for predicting all samples of the target block.

Depending on the filter context, a different TW, corresponding to a different neighbouring region is defined, as shown in Figure 7.4. For instance, filter F_H^k, which only combines causal samples from the left of the target sample $\hat{X}(\mathbf{n})$, is estimated in the neighbouring region defined on the left side of the current block. The size of the TW depends on the size of the block and dimension D_{TW}, represented in Figure 7.4.

Although the proposed linear models may use all the samples within the $D_f \times D_f$ context area, it is not expected that they are all relevant for the prediction process. The least-squares-based training procedure should attribute higher weighting coefficients for more important samples and lower weighting coefficients to the less relevant samples. However, when the number of filter coefficients (or variables) is large in relation to the number of training samples (observations), over-fitting issues may occur. In these cases, the estimated models tend to memorise the training data rather than learn the underlying linear model. As a consequence, these models may present a poor predictive performance. This problem is aggravated when the filter size is larger than TW size, resulting in an under-determined problem, that provides infinite solutions.

In order to overcome these problems, regularisation constraints are typically added to the ordinary least-squares regression problem. We investigated the use of a sparsity-based regularisation function for the least-squares problem. The classical LSP optimisation problem (see Equation 2.2) using a sparsity constraint can be given by:

$$\arg\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{C}\mathbf{a}\|_2^2 \quad \text{subject to} \quad \|\mathbf{a}\|_0 \leq t\,, \tag{7.4}$$

where $\mathbf{a}$ is the sparse coefficients vector, $\mathbf{C}$ is the correlation matrix computed as described in Subsection 2.4.2, $\mathbf{y}$ is the training data and t is a bound on sparsity (number of non-zero coefficients).

In general, searching the exact solution of this problem is NP-hard [131], and thus computationally intractable. To solve this problem, there are various algorithms in the literature that provide approximated solutions with feasible complexity. Some of these algorithms are reviewed in the following section.

7.3 Sparse model estimation algorithms

In this section we review some algorithms to estimate approximate solutions to the constrained least-squares problem described by Equation 7.4. These methods include the iterative greedy matching pursuit-based algorithms [41, 132], as well as less greedy solutions, such as least angle regression [133], LASSO [134] and Elastic Net regression [135]. From this point onwards, we will use the notation of Equation 7.4, assuming that $\mathbf{y}$ is centred to have zero mean, and the columns of $\mathbf{C}$ are normalised to unit l_2-norm and centred to have zero mean.

7.3.1 Matching Pursuit algorithms

The Matching Pursuit (MP) [41] algorithm is one of the greediest approaches for variable selection. It searches the predictor (column of matrix $\mathbf{C}$) that best fits the residual at each iteration and adds it to the support (or model). Note that, due to the previous normalisation to unit l_2-norm and centring to zero mean assumptions, the fitting procedure is equivalent to the predictor correlation. The l_2-norm of the error of the objective function (*i.e.* the residual) is often used as the stopping criterion.

Since MP does not orthogonalise the residue in relation to the current model predictors, the same predictor can be chosen again in later iterations. Because of this, MP may potentially require a large number of iterations to converge. The Orthogonal Matching Pursuit (OMP) [132] is an improved version of MP which solves the predictor repetition and the convergence issue. The main difference from MP is that all the solution coefficients are updated after at each iteration by computing the orthogonal projection of the signal onto the set of selected predictors. This procedure ensures that each new residual is orthogonal to the already chosen predictors.

Although OMP is more computationally complex than MP, it tends to use less iterations due to the residue orthogonalisation step. Furthermore, OMP provides more accurate sparse solutions than the MP algorithm. A description of MP and OMP algorithms for sparse prediction methods has been previously presented in Subsection 2.5.2.

7.3.2 Least Angle Regression

Besides their sometimes excessively greedy nature, due to the aggressive fitting, MP methods tend to choose only one predictor in the presence of a group of highly correlated predictors. This may exclude useful predictors. A less greedy version of the traditional forward selection methods has been proposed in [133], denominated as Least Angle Regression (LAR). The LAR method has very interesting properties and can be easily modified to behave like other model selection algorithms, namely LASSO and Incremental Forward Stagewise methods.

The LAR algorithm can be briefly described in what follows. Similarly to forward selection algorithms, LAR starts with all coefficients equal to zero and finds the predictor, $\mathbf{c}_1$, which is most correlated with the response $\mathbf{y}$. To find the second predictor, LAR moves along the direction of $\mathbf{c}_1$, updating the residual along the way, until some other predictor, $\mathbf{c}_2$, becomes as correlated as $\mathbf{c}_1$ with the current residual. These two predictors form the most correlated set, named as active set. Then, LAR proceeds along a new direction that is equiangular to both predictors $\mathbf{c}_1$ and $\mathbf{c}_2$ (*i.e.* along the least angle direction), and it stops when a third predictor, $\mathbf{c}_3$, has as much correlation as $\mathbf{c}_1$ and $\mathbf{c}_2$ with the current residual, adding it to the active set. This process continues by using the equiangular direction between the three found predictors, until a fourth predictor is obtained, and so on. LAR terminates its procedure when the number of desired predictors is achieved or when the full solution path is obtained.

The LAR algorithm is very efficient, since it can solve the whole solution path (*i.e.* the sequence of all solutions from 0 up to p predictors), at the cost of solving the Ordinary Least-Squares (OLS) problem for the full set of p predictors. More details about LAR algorithm and its algebraic derivation can be found in [133].

7.3.3 LASSO regression

The Least Absolute Shrinkage and Selection Operator (LASSO) [134] has been proposed in statistics literature for estimation of linear models, using an l_1-norm constraint on the coefficients' vector. An equivalent formulation used in signal processing literature is known as Basis Pursuit [136]. The LASSO constraint is a convex relaxation of the intractable l_0-norm problem described in Equation 7.4. It can provide approximate solutions to the l_0-norm problem formulation with reasonable computational complexity, through the use of convex optimisation techniques. A useful unconstrained formulation of the LASSO problem for Equation 7.4, based on a Lagrangian multiplier λ, is given by:

$$\arg\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y} - \mathbf{Ca}\|_2^2 + \lambda\|\mathbf{a}\|_1 \ . \tag{7.5}$$

Due to the nature of l_1-norm, LASSO enjoys two main features: model selection and coefficient shrinkage. The model selection feature guarantees that estimated solutions using LASSO are sparse and thus it can provide accurate approximations to the original l_0-norm problem formulation. With $\lambda = 0$, the problem in Equation 7.5 becomes the classical OLS minimisation problem. As λ increases, LASSO tends to shrink the OLS coefficients towards zero, which results in decreased coefficients' variance and a more biased solution. Consequently, the solution becomes more stable and the predicted values present lower variance, improving the overall prediction accuracy.

The LASSO problem can be solved using various optimisation algorithms, including a modified version of the previously presented LAR algorithm. As pointed out above, the advantage of LAR algorithm is its ability to solve the full solution path (from $\lambda = 0$ up to very large λ values) with efficient computational complexity. Unlike the original LAR algorithm, the LASSO-modified version of LAR may remove predictors during its iterations. When one coefficient changes its sign, the associated predictor is removed and the next equiangular direction is computed with the remaining predictors. Due to the possible predictor dropping, the LASSO-modification of LAR may slightly increase the computational complexity, when compared to the original LAR algorithm [133].

7.3.4 Elastic Net regression

The Elastic Net (EN) regression [135] has been proposed as an extension to the LASSO formulation, in order to improve its prediction performance for some scenarios, *e.g.* when the number of predictors (p) is much larger than the number of samples (n). In such a scenario, LASSO can select at most n out of p predictors, which is a problem when the number of relevant predictors is superior to n. This limitation is intrinsic to the convex optimisation problems and it is also present on the LAR algorithm, as previously discussed. Another interesting feature of EN is the ability to include groups of correlated predictors into the model, which is often desirable in prediction scenarios. In contrast to EN, LASSO lacks the grouping effect, selecting only one arbitrary predictor from a group of correlated ones.

The EN problem can be formulated as an extended version of the LASSO problem, adding an l_2-norm component (the Ridge penalty) to the l_1-norm penalty of LASSO. Considering the LASSO problem of Equation 7.5, its EN extension can be written as:

$$\arg\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y} - \mathbf{C}\mathbf{a}\|_2^2 + \lambda_1\|\mathbf{a}\|_1 + \lambda_2\|\mathbf{a}\|_2^2 \,. \tag{7.6}$$

Unlike the model estimation methods previously presented, the EN uses two parameters to adjust the influence of the used penalties: the sparsity parameter λ_1 and the grouping parameter λ_2. EN can be transformed in the LASSO problem by setting $\lambda_2 = 0$, or in the ridge regression when $\lambda_1 = 0$. By setting both parameters equal to zero, the problem reduces to the classical OLS regression. The use of ridge penalty in classical OLS is known to improve its prediction performance through a bias-variance trade-off. However, ridge regression does not provide sparse solutions like LASSO. As previously discussed, LASSO is able to enforce both sparsity and shrinkage properties. The combination of ridge with LASSO is advantageous in many situations, because it converts the convex problem into a strictly convex one. This guarantees that similar predictors receive similar coefficients, justifying the grouping property of EN.

For the resolution of the EN problem, described in Equation 7.6, standard methods used to solve the LASSO problem can also be used but based on an augmented dataset. The number of samples in the augmented dataset is $n + p$ and the modified matrix $\mathbf{C}$ has rank p, which means the EN solution may use all the p predictors. This eliminates the limitation of the original LASSO solution, that cannot use more than $\min(n, p)$ non-zero coefficients. An attractive approach in terms of computational complexity to solve EN is to use the LAR algorithm with LASSO modification. Note that the λ_1 parameter defines the early stopping of LASSO algorithm, while λ_2 parameter participates in the formation of the augmented dataset. In order to solve some issues of EN method, which introduces extra bias to the estimated solutions, degrading the prediction performance, the EN estimated coefficients can be re-scaled. More details about the EN can be found in [135].

7.4 Proposed algorithm based on adaptive sparse predictors for HEVC

In this section we describe the algorithm of the proposed generalised sparse linear prediction method for the HEVC standard. In order to improve filter design flexibility, the maximum size of filter context is adaptively chosen. We also investigate adaptive early stopping conditions for the evaluated model estimation algorithms presented in Section 7.3.

The proposed two-stage intra prediction framework has been incorporated into

HEVC standard by replacing the 33 angular modes and keeping the original planar and DC modes. The linear filtering stage is used as described in Section 7.2. The maximum filter context size, given by parameter D_f (see Figure 7.3) should be provided. This parameter is important because it indicates the maximum distance of non-zero filter coefficients from the predicted block (or maximum filter size). The use of the D_f parameter can be comparable to a step penalty function, which gives equal importance (weight 1) to samples within a squared context area, discarding the outside samples (weight 0). As an alternative, more complex and smooth penalties might be used, for instance, smooth functions that exponentially decrease the importance weight of the coefficient when its distance to $\hat{X}(\mathbf{n})$ increases.

In the proposed algorithm, we consider the straightforward step-based penalty solution, where D_f indicates the maximum filter size, as illustrated in Figure 7.3. In addition, we use an adaptive approach, in which the maximum filter size is explicitly transmitted to the decoder. We minimise the amount of encoded side information by using only two possible filter sizes, specifically $D_f = 1$ and $D_f = 31$, which are represented by a binary flag (f_o). A higher number of filter sizes (D_f values) could be used, however, it would require additional signalling bits. Note that $D_f = 1$ provides similar results to the HEVC angular modes (assuming coefficient equal to 1), which corresponds to the filter contexts shown in Figure 7.1.

For $D_f = 31$, up to 961 non-zero coefficients are available, but only a few should be selected by the model estimation algorithms described in Section 7.3. A point of concern for these algorithms is the choice of an appropriate stopping condition. In the case of the LASSO problem, the stopping criterion is related to the λ parameter. The most straightforward solution for this problem is to use a fixed number of non-zero predictors, similar to traditional LSP methods. In such an approach, which we call KNP (k non-zero predictors), the optimisation algorithm should stop whenever k, the predefined number of predictors, is reached.

In addition to trivial KNP method, we investigated alternative early stopping criteria, based on adaptive solutions. Statistical methods, such as *Akaike Information Criterion* (AIC) [137], *Bayesian Information Criterion* (BIC) [138] or *Mallows* C_p [139], have been used as stopping criteria for efficient model selection. However, in our scenario, where the number of predictors (maximum filter context size) is usually larger than the number of training samples (*i.e.* $p \gg n$), these methods are not recommended, because they turn into a measure of training error only. A popular solution to adaptively find the optimal number of non-zero coefficients, when only training data is available, is the Cross-Validation (CV) method [140]. However, because the CV has high computational complexity, we decided to investigate two other methods to find the optimal number of predictors under the evaluated model estimation algorithms.

The first adaptive solution is based on the Variation of the Training Error (VTE) at each iteration. The idea behind this method is to stop the optimisation algorithm when the variation of mean squared error in the training window is inferior to a predefined threshold, ρ_E, in terms of percentage. The second used method is based on the Geometrical Stopping Criterion (GSC), originally proposed in [141] for the LAR algorithm.

GSC is based on the fact that the angle, $\theta_{j,n}$, between the residue and the predictor $\mathbf{c}_j$ tends to 90° as the algorithm iteration n increases. Thus, the stopping criterion is defined by $\Delta\boldsymbol{\theta}_n \leq \sigma_{\boldsymbol{\theta}_1}$, where $\boldsymbol{\theta}_n = [\theta_{1,n} \ldots \theta_{j,n} \ldots \theta_{p,n}]$, $\Delta\boldsymbol{\theta}_n = \max(\boldsymbol{\theta}_n) - \min(\boldsymbol{\theta}_n)$ and $\sigma_{\boldsymbol{\theta}_1}$ is the standard deviation of the angles at first iteration. In practice, this method forces the algorithm to stop when the difference between the maximum and minimum angles is smaller than their standard deviation at the first iteration. Since this approach is static, we made a simple modification that allows for tuning, introducing the parameter ρ_G, yielding $\Delta\boldsymbol{\theta}_n \leq \rho_G.\sigma_{\boldsymbol{\theta}_1}$.

The optimal tuning parameters for KNP, VTE and GSC stopping criteria, the parameters k, ρ_E and ρ_G, respectively, are fixed for the whole image coding process. We obtained the optimal values through experimental tests (see Section 7.5) and we verified that they are approximately consistent for several images with different characteristics. Thus, the only encoded side information in the proposed intra prediction framework is the optimal maximum filter context size (represented by f_o), and the optimal geometric transformation selected in the second stage of the proposed algorithm (represented by t_o).

In order to encode these symbols (f_o and t_o) we have made minimal changes to the HEVC bitstream syntax. Since they replace the angular modes, the optimal geometric transformation (t_o) is encoded using the existing symbols and entropy coding solution for the angular modes. This means that the 33 symbols for geometric transformations are jointly encoded with the planar and DC modes by using the existing MPM scheme to predict the geometric transformation and the Context-Adaptive Binary Arithmetic Coding (CABAC) [63] for entropy coding using either the MPM index coding or a fixed-length-code of 5 bits.

Regarding the optimal filter context size (that can be $D_f = 1$ or $D_f = 31$), an additional binary flag (f_o) is signalled in the bitstream after encoding the optimal geometric transformation (t_o). The statistical redundancy of this binary symbol is exploited using CABAC, based on 33 possible contexts that correspond to each geometric transformation, being initialised with uniform distributions.

Algorithm 7.1 summarizes the encoder side procedure of the proposed method based on adaptive sparse linear predictors for the prediction of one PU in HEVC standard.

Algorithm 7.1: Prediction algorithm based on adaptive sparse linear predictors (encoder).

```
Input: causal reconstructed image of current PU;
Output: predicted block $\hat{P}$ for current PU;
evaluate Planar and DC modes and compute costs $J_{Planar}$ and $J_{DC}$;
set $D_f = 1$;
use first-order filters: $F_H^1$, $F_D^1$ and $F_V^1$ (see Figure 7.1), to generate 3 first-stage output blocks;
for each available geometric transformation $t$; do
    use geometric transformation $t$ over one of the previous generated first-stage output blocks (selected one depends on $t$ - see Table 7.1);
    save previous result as temporary predicted block $P(D_f = 1, t)$;
    compute RD cost $J(D_f = 1, t)$;
end for
set $D_f = 31$;
estimate sparse models (using OMP, LAR, LASSO or EN with some stopping criterion: KNP, VTE or GSC) for 3 filter context configurations with $D_f = 31$ (see Figure 7.3);
use estimated $k$-sparse filters: $F_H^k$, $F_D^k$ and $F_V^k$, to generate the 3 first-stage output blocks;
for each available geometric transformation $t$; do
    use geometric transformation $t$ over one of the previous generated first-stage output blocks (selected one depends on $t$ - see Table 7.1);
    save previous result as temporary predicted block $P(D_f = 31, t)$;
    compute RD cost $J(D_f = 31, t)$;
end for
find $D_f^*, t^* = \min_{D_f, t} J(D_f, t)$;
if $J(D_f^*, t^*) < \min(J_{Planar}, J_{DC})$; then
    set final predicted block $\hat{P} = P(D_f^*, t^*)$;
    set $t_o = t^*$;
    set $f_o = 0$ if $D_f^* == 1$, or $f_o = 1$ if $D_f^* == 31$;
    encode $t_o$ and $f_o$ symbols using CABAC;
else
    if $J_{Planar} < J_{DC}$; then
        set Planar mode output as final predicted block $\hat{P}$ and encode its flag;
    else
        set DC mode output as final predicted block $\hat{P}$ and encode its flag;
    end if
end if
```

7.5 Experimental results

The proposed intra prediction framework has been evaluated for still image compression using the HEVC reference software HM-14.0 version [72]. Four coding approaches using different predictor estimation algorithms have been used in the experiments: HEVC-OMP, HEVC-LAR, HEVC-LASSO and HEVC-EN. The used test images are organised in two sets. Set 1 includes test images with complex textured areas and repeated patterns, as illustrated in Figures A.12 to A.17 of Section A.2. Set 2 includes first frame of HEVC test sequences from classes B (1920×1080), C (832×480), D (416×240) and E (1280×720) [69]. Only the luminance component has been considered in the experiments. In the following, we analyse the rate-distortion (RD) performance of these algorithms as well as their optimal parameters.

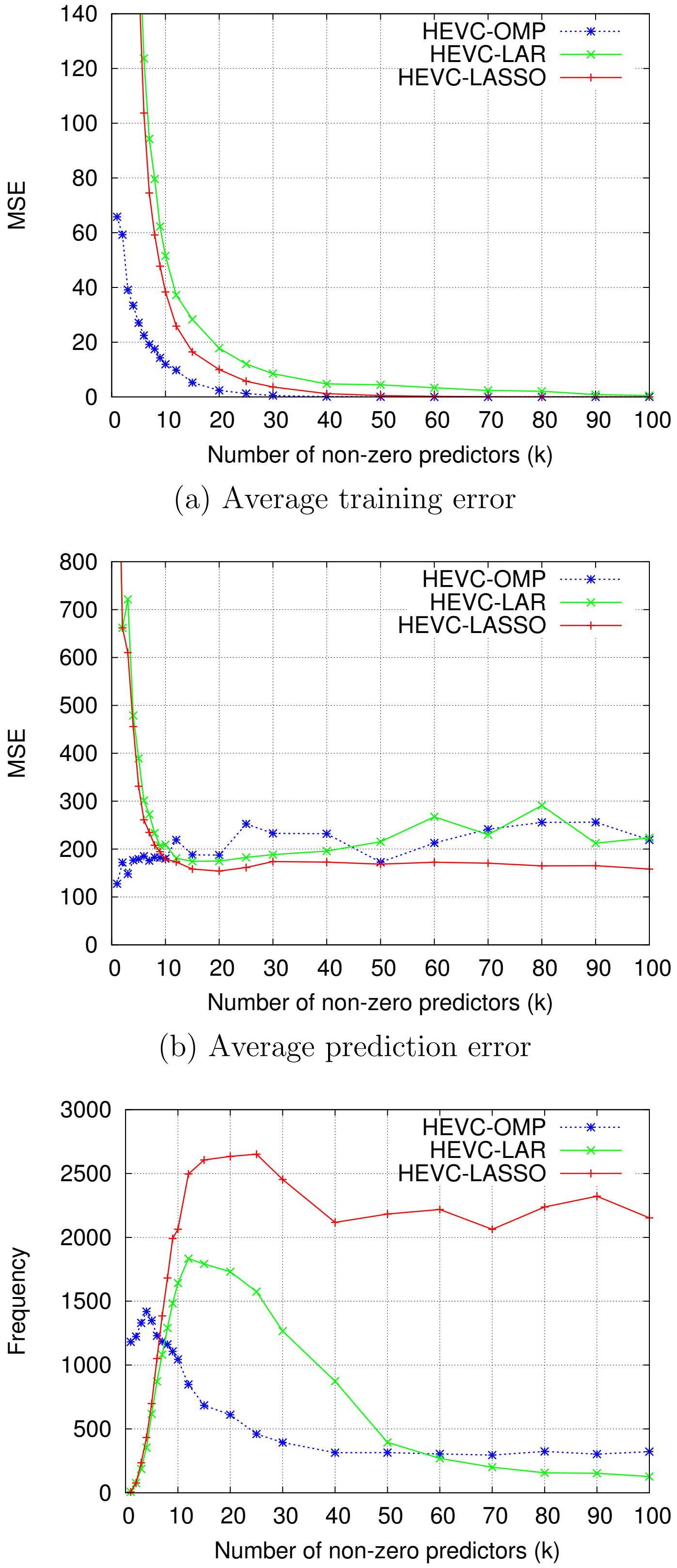
7.5.1 Effect of sparsity constraints

In this subsection, we analyse the influence of the sparsity constraints on the linear prediction models solved by OMP, LAR and LASSO algorithms. In order to evaluate the importance of the sparsity constraint, Barbara image was encoded using a modified version (explained below) of HEVC-OMP, HEVC-LAR and HEVC-LASSO with QP=27, and the KNP stopping method. Different sparsity levels, in the range of $k = 1, ..., 100$ were tested.

In Figure 7.5 we present some statistics about training and prediction error, as well as mode usage frequency versus the number of non-zero predictors (k). These results only correspond to those blocks chosen by the encoder to be predicted by the proposed method using the high-order filter case with $D_f = 31$, as defined in Section 7.4, that are the cases where the presented optimisation algorithms are effectively used. In order to better interpret the error of linear prediction, we disabled the geometric transformations for the high-order filters.

The average training error of Figure 7.5 corresponds to the average MSE obtained in the training window for the blocks predicted using sparse linear predictors. Such training error gives the average modelling capabilities of the sparse linear predictors in the training windows, defined around left and top regions of the predicted block. The results of Figure 7.5a show that the average approximation error tends to decrease as the number of non-zero predictors (k-value) increases, regardless of the predictor estimation algorithm. This is an expected result because the linear predictors are optimised for the training window. Thus, as more predictors are added to the linear model, the better are the modelling capabilities, decreasing the approximation error. This error may be exactly zero, namely when the number of predictors equals the number of samples in the training window, assuming linearly independent predictors.

According to Figure 7.5a, the training error obtained by OMP algorithm tends to be lower than the one of LAR and LASSO, for the same sparsity value. This can be justified by the highly greedy behaviour of OMP, which provides a good approximation using less iterations, *i.e.* less predictors. The average prediction error on



(a) Average training error

(b) Average prediction error

(c) Mode usage frequency

Figure 7.5: Influence of linear filter sparsity on (a) average training error, (b) average prediction error and (c) mode usage frequency, for compression of Barbara image (QP 27) using OMP, LAR and LASSO algorithms, with varying sparsity level $k = 1, ..., 100$ (number of non-zero predictors).

the block, shown in Figure 7.5b, presents a different behaviour than the one of the training error. The greedy OMP algorithm tends to provide better prediction performance (lower prediction MSE) for smaller k-values. Regarding LAR and LASSO algorithms, Figure 7.5b shows that these methods tend to provide better prediction performance for larger k-values, more specifically when $k > 10$.

It is important to note that lower average prediction error on Figure 7.5b does not imply better RD performance. The presented results correspond to the average prediction error of the blocks chosen by the encoder. Those blocks may vary with each experiment based on different k-values, due to different RD decisions which take into account not only the prediction but also the residue approximation cost. Thus, to better understand these results, the number of times that the sparse linear prediction methods were used (usage frequency) is illustrated in Figure 7.5c. We may observe that OMP-based prediction is more frequent when smaller k-values are used (with maximum at $k = 4$), decreasing its frequency as more predictors are considered. These results are somewhat in line with the average prediction errors of OMP algorithm, so we should expect a better overall RD performance when fewer predictors are used.

In regard to LAR and LASSO algorithms, the usage frequency results tend to improve as the number of non-zero predictors increases, until reaching a maximum around $k = 15$. For higher k-values the usage frequency of these methods decreases again, mainly for the LAR algorithm. The prediction errors, shown in Figure 7.5b, of LAR and LASSO methods also reach their minima close to $k = 15$, although they do not increase significantly for higher k-values. These observations suggest that there is an optimal number of non-zero coefficients, which provides the highest RD performance. When we compare the results for the three methods, we observe that LASSO provides the best prediction results and higher usage frequency. In the next subsection we analyse the RD performance of these methods using different sparsity levels and three different stopping criteria.

7.5.2 Regularisation parameters for optimal RD performance

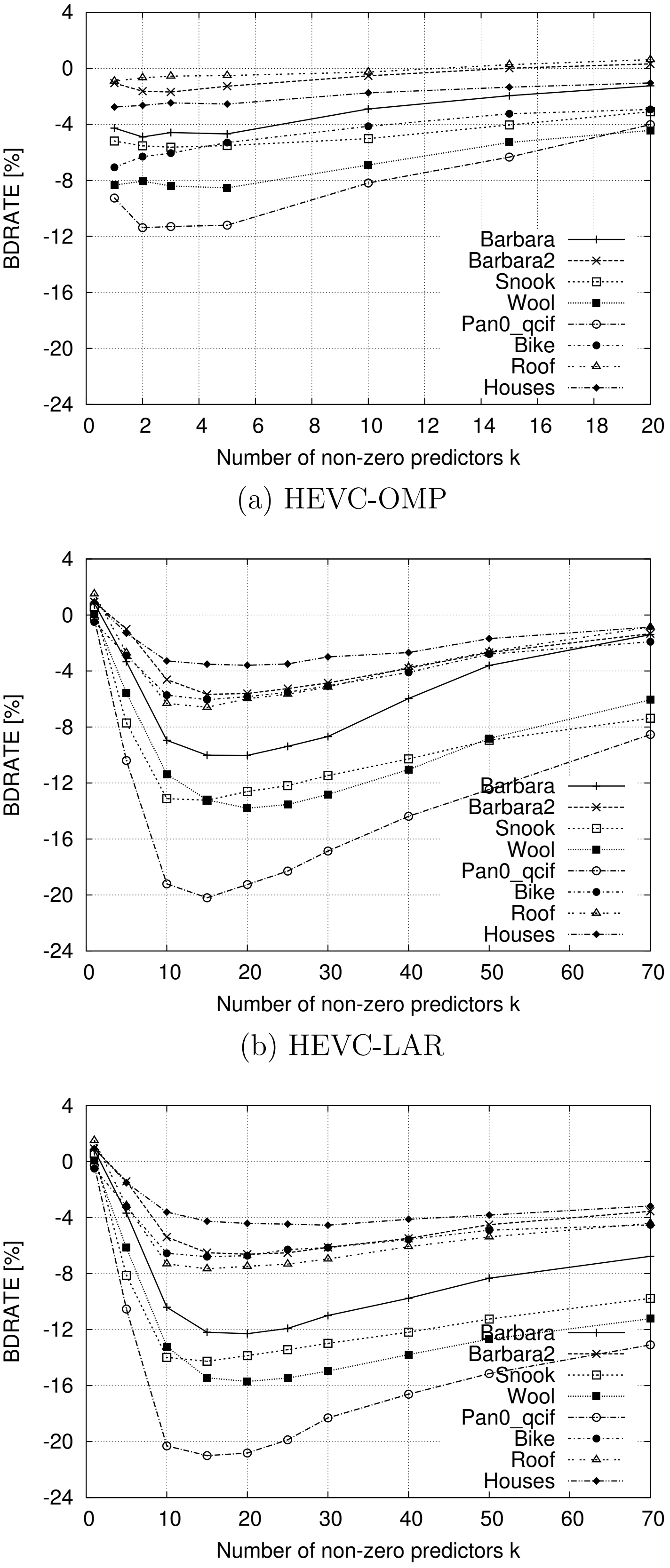
Figure 7.6 presents the Bjontegaard Delta Rate (BDRATE) [70] results as a function of the number of non-zero predictors, using HEVC-OMP, HEVC-LAR and HEVC-LASSO encoders with KNP criterion and test set 1. BDRATE results were computed against the original HEVC standard. In the case of the greediest approach, the HEVC-OMP, one may see that bitrate savings tend to be more significant (*i.e.* higher negative BDRATE values) for small k-values. Less greedy approaches, specifically the HEVC-LAR and HEVC-LASSO schemes, present a different behaviour compared to HEVC-OMP. Their optimal RD performance is obtained with the number of predictors $k = 15$, that is rather consistent for all tested images. Moreover, it can be observed that the maximum bitrate savings using these schemes are significantly higher than using HEVC-OMP. These results support the observations presented in Subsection 7.5.1.

The results using the VTE stopping criterion are illustrated in Figure 7.7 for HEVC-OMP, HEVC-LAR and HEVC-LASSO schemes and test set 1, with BDRATE results computed against the original HEVC standard. The performance of HEVC-OMP approach increases as ρ_E goes from 1% to 10%. Between 10% and close to 100% its performance is not consistent, as it may decrease or not, depending on the test image. Results for the HEVC-LAR and HEVC-LASSO schemes reveal that an optimal value for ρ_E, that works for all tested images, can be found. Regarding HEVC-LAR, the optimal VTE stopping criterion value is around 0.4%, while HEVC-LASSO performs better when ρ_E is 0.1%.

Figure 7.8 presents the BDRATE results when the geometrical stopping criterion is used on HEVC-OMP, HEVC-LAR and HEVC-LASSO schemes with test set 1. In this scenario, HEVC-OMP seems to perform more efficiently on average with $\rho_G = 0.5$. HEVC-LAR and HEVC-LASSO achieve the best results for most images at $\rho_G = 2$ and $\rho_G = 1$, respectively.

Another solution investigated in this work is based on the EN method (HEVC-EN). As explained in Section 7.4, EN can be reproduced using the LASSO algorithm with modified input matrices. The first parameter of EN, λ_1, is related to the stopping criterion of iterative LASSO algorithm, while the second parameter, λ_2, associated to l_2-norm constraint, is incorporated in the modified input matrices. Figure 7.9 presents the BDRATE results of HEVC-EN for test set 1, using optimal stopping parameter for LASSO and a variable value for λ_2. The plot of Figure 7.9a uses LASSO implementation with $k = 15$ non-zero predictors, while the plot of Figure 7.9b uses VTE stopping criterion with $\rho_E = 0.1\%$. In order to analyse the impact of l_2-norm constraint, BDRATE values have been computed relative to the results of LASSO. Note that LASSO is a particular case of the EN method, specifically when $\lambda_2 = 0$.

From Figure 7.9, we may observe that adding l_2-norm constraint to LASSO problem does not provide consistent gains. For some cases, BDRATE performance slightly increases, namely for Pan0_qcif image (for both plots of Figure 7.9) and Barbara image for the plot of Figure 7.9b, with $\lambda_2 = 0.0001$. However, other images present a decrease of RD performance or null gain, independently of the λ_2 value. Such results lead us to conclude that the LASSO approach is preferable to the EN method. Furthermore, the EN requires an extra regularisation parameter, λ_2, which adds complexity without consistent advantages.

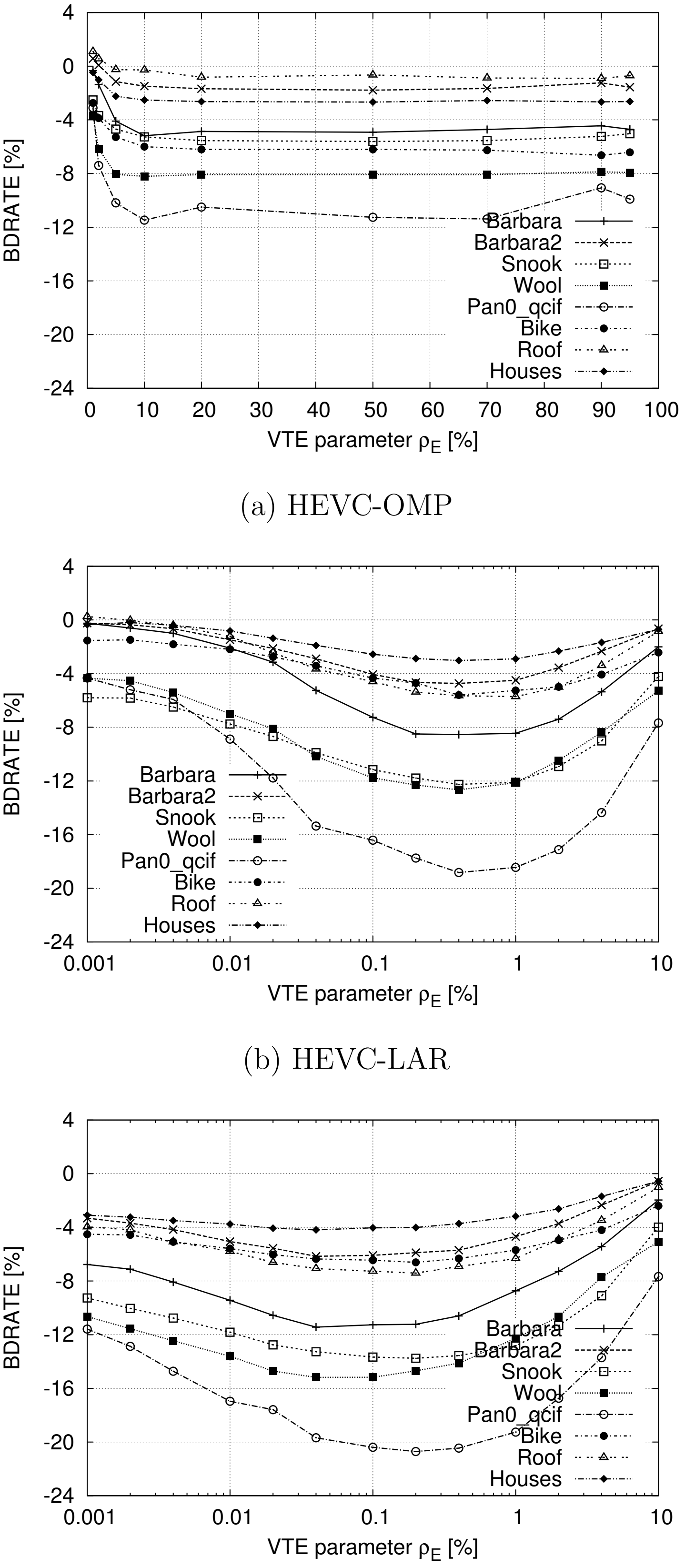


(a) HEVC-OMP

(b) HEVC-LAR

(c) HEVC-LASSO

Figure 7.6: BDRATE results of proposed HEVC with generalised intra prediction framework for test set 1, as function of the number k of non-zero predictors (KNP stopping criterion) used by OMP, LAR and LASSO methods.

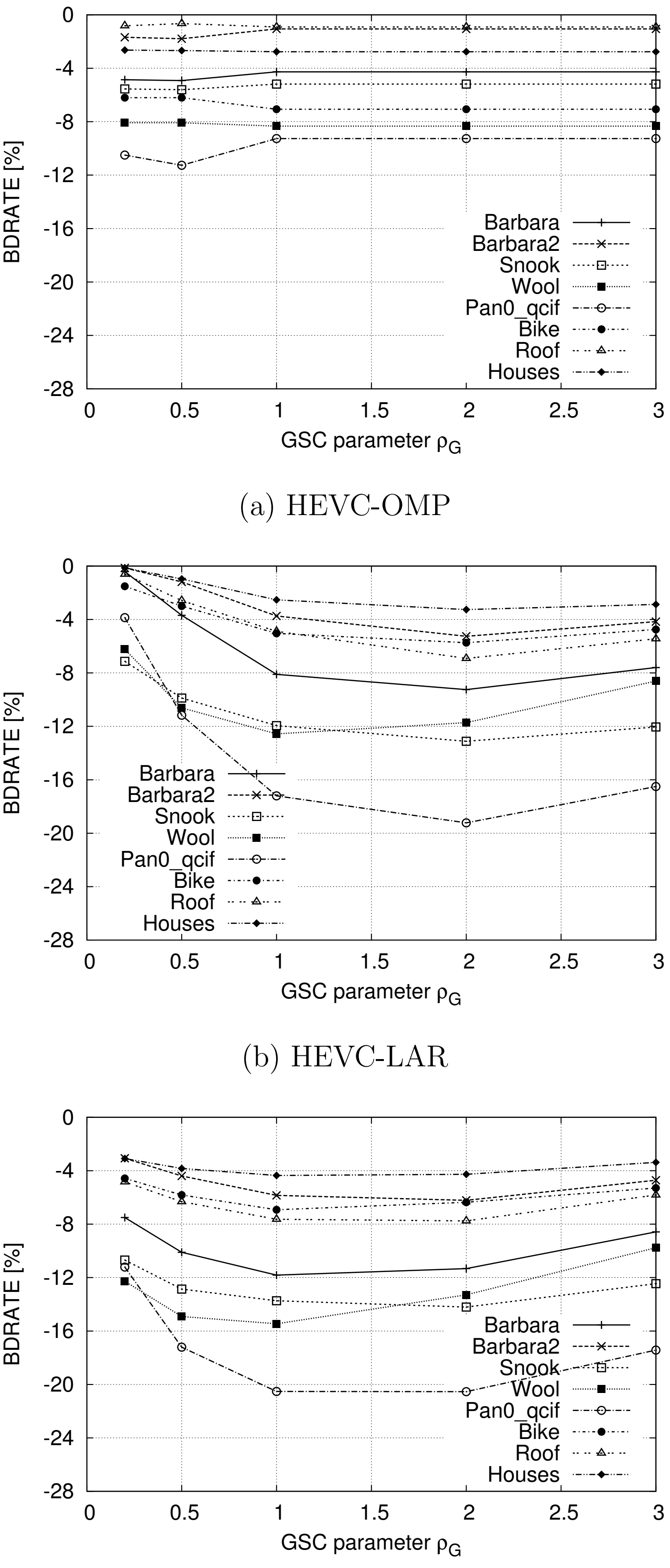


(a) HEVC-OMP

(b) HEVC-LAR

(c) HEVC-LASSO

Figure 7.7: BDRATE results of proposed HEVC with generalised intra prediction framework for test set 1, as function of the VTE-based stopping criterion threshold, ρ_E.

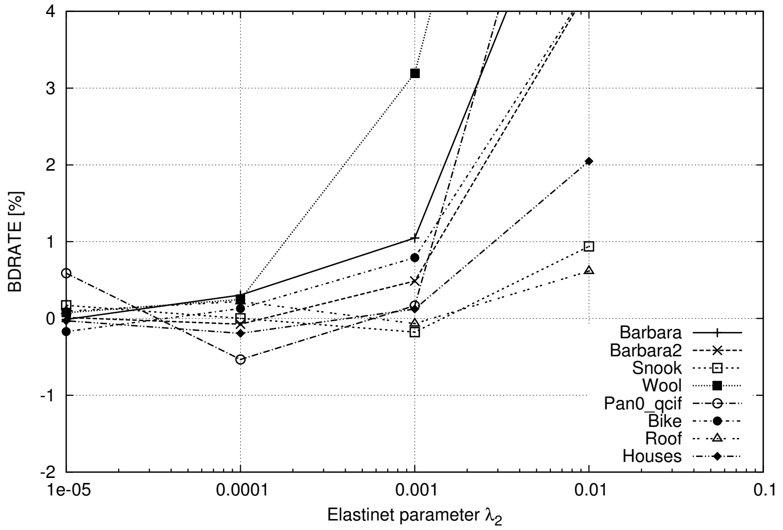


(a) HEVC-OMP

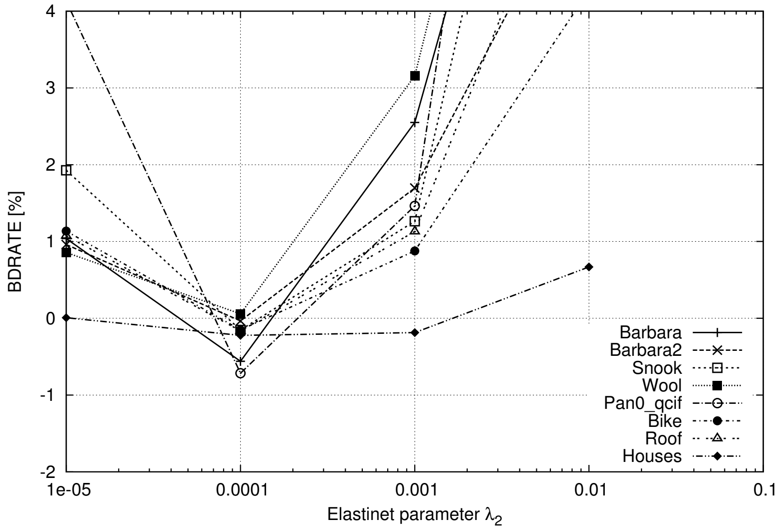
(b) HEVC-LAR

(c) HEVC-LASSO

Figure 7.8: BDRATE results of proposed HEVC with generalised intra prediction framework for test set 1, as function of the geometrical stopping criterion threshold, ρ_G.



(a) HEVC-EN using KNP with $k = 15$



(b) HEVC-EN using VTE criterion with $\rho_E = 0.1\%$

Figure 7.9: BDRATE results for test set 1 using HEVC-EN based on Elastic Net as function of parameter λ_2, with constant parameter λ_1 given by (a) KNP with $k = 15$ non-zero predictors and (b) VTE criterion with $\rho_E = 0.1\%$. HEVC-LASSO is used as reference method to compute BDRATE results.

In order to provide a clear comparison between the OMP, LAR and LASSO, Table 7.4 shows the previously presented BDRATE results for the optimal regularisation parameter values. These optimal parameters have been derived from Figures 7.6, 7.7 and 7.8, by choosing the stopping criterion thresholds that roughly provide the maximum average bitrate savings for all the images of test set 1. For the case of OMP method, we defined $k = 2$, $\rho_E = 10\%$ and $\rho_G = 0.5$ as being the optimal parameters for the KNP, VTE and GSC methods, respectively. Similarly, the choice for LAR has been $k = 15$, $\rho_E = 0.4\%$ and $\rho_G = 2$. For LASSO, we set

$k = 15$, $\rho_E = 0.1\%$ and $\rho_G = 1$.

The bold BDRATE values of Table 7.4 indicate the best results among the three stopping criteria for each optimisation method and test image. When we compare OMP, LAR and LASSO, using the same stopping criteria, we observe that LASSO always provides higher bitrate savings. As previously discussed the greedy OMP method presents the worst performance, being significantly inferior to LAR and LASSO methods. These results also reveal that the use of l_1-norm constraint (LASSO) tends to be more effective than LAR procedure for image prediction.

Regarding the results for different stopping criteria, they depend on the optimisation algorithm. For the case of OMP, apparently there is no better stopping criterion. Different images reach optimal RD results using different stopping criteria. In respect to the not-so-greed algorithms, LAR and LASSO, we observe more consistent results for all images. The VTE-based stopping criterion demonstrates an inferior performance than the one of traditional KNP and adaptive GSC methods. Both LAR and LASSO present the best average results using the traditional KNP solution. For few images, GSC provides a superior RD performance. However, such performance difference is insignificant in all cases (less than 0.5%).

The better performance of KNP stopping criterion over the adaptive VTE method can be explained by the poor relation between training and predictor errors. As can be seen in Figure 7.5, the training error tends to decrease monotonically as more predictors are added to the model. On the other hand, the prediction error has a different behaviour and it may increase or decrease with the inclusion of new predictors. This uncertain relationship between training and prediction errors does not guarantee that VTE stopping method is the best one.

7.5.3 RD performance relative to other intra prediction methods

In this section we compare the proposed prediction framework based on the generalised optimal sparse predictors (GOSP) with the existing state-of-the-art prediction methods for image compression. For this purpose, we used the best configuration of our method, based on LASSO algorithm with KNP stopping criterion using $k = 15$ non-zero predictors. The intra prediction methods in these experiments include the traditional LSP algorithm, the sparse-LSP (SLSP) method proposed in Chapter 6, the TM algorithm and the LLE method [38]. The traditional LSP has been used as described in Subsection 2.4.2 for block-based image prediction using $k = 10$ closest predictors. For the SLSP, TM and LLE methods, the same configuration parameters (*e.g.* training window size, number of predictors or maximum filter size) as proposed in Section 6.3, were used. Furthermore, all the comparing methods were incorpo-

Table 7.4: BDRATE (%) results for test set 1 using the proposed HEVC algorithm with generalised intra prediction framework based on OMP, LAR and LASSO, with three proposed stopping criteria (KNP, VTE and GSC).

Image	OMP-KNP	OMP-VTE	OMP-GSC	LAR-KNP	LAR-VTE	LAR-GSC	LASSO-KNP	LASSO-VTE	LASSO-GSC
Barbara	-4,90	**-5,18**	-4,92	**-10,01**	-8,54	-9,25	**-12,19**	-11,26	-11,82
Barbara2	-1,63	-1,49	**-1,79**	**-5,67**	-4,73	-5,26	**-6,51**	-6,08	-5,84
Snook	-5,53	-5,27	**-5,61**	**-13,23**	-12,25	-13,12	**-14,26**	-13,67	-13,73
Wool	-8,06	**-8,21**	-8,08	**-13,21**	-12,67	-11,72	-15,45	-15,17	**-15,46**
Pan0_qcif	-11,38	**-11,47**	-11,26	**-20,20**	-18,82	-19,23	**-21,00**	-20,39	-20,53
Bike	**-6,30**	-6,00	-6,20	**-6,05**	-5,60	-5,74	-6,79	-6,45	**-6,92**
Roof	**-0,67**	-0,29	-0,65	-6,59	-5,65	**-6,91**	**-7,66**	-7,27	-7,64
Houses	-2,64	-2,52	**-2,68**	**-3,51**	-3,03	-3,26	-4,26	-4,03	**-4,36**
Average	-5,14	-5,05	**-5,15**	**-9,81**	-8,91	-9,31	**-11,01**	-10,54	-10,79

Table 7.5: BDRATE results for test set 1 using proposed HEVC algorithm with generalised optimal sparse predictors based on LASSO method, compared with other related prediction methods.

Image	**LSP**	**TM**	**LLE**	**SLSP**	**GOSP**
Barbara	-3,37	-1,86	-3,99	-6,27	**-12,19**
Barbara2	-0,31	-1,13	-3,13	-4,05	**-6,51**
Snook	-0,10	-3,83	-13,64	**-14,31**	-14,26
Wool	-2,25	-5,70	-9,49	-10,38	**-15,45**
Pan0_qcif	0,06	-5,25	-13,27	-15,73	**-21,00**
Bike	-1,69	-5,59	-5,24	-5,85	**-6,79**
Roof	-0,92	-1,61	-4,34	-5,80	**-7,66**
Houses	-0,41	-2,98	-4,21	**-4,72**	-4,26
Average	-1,12	-3,49	-7,16	-8,39	**-11,01**

rated in HEVC standard by replacing the Angular 3 mode of directional prediction framework. Note that the proposed GOSP is the only method that generalises directional prediction, and replaces all the angular modes of HEVC standard. Only planar and DC modes have been maintained.

Tables 7.5 and 7.6 present the BDRATE results of the proposed solution, based on generalised optimal sparse predictors (GOSP), as well as the compared methods, all using the HEVC standard, for the compression of test images of set 1 and set 2, respectively. BDRATE results were computed relative to the RD performance of original HEVC standard, using HM-14.0 software with intra main profile. Images of test set 1 present more complex textures and repeated patterns that can be better exploited by sparse predictors. This is clearly shown in the results of Table 7.5. For most images, GOSP provides the higher RD performance, achieving up to 6% bitrate savings over SLSP, namely for Barbara and Pan0_qcif. In the worst cases, GOSP performance is quite similar to the one of existing methods, with performance differences inferior to 1%. This is case of Snook and Houses images, that present similar performance with GOSP and SLSP methods.

The results of Table 7.6 refer to HEVC test sequences of classes B, C, D and E. One may observe that the proposed solution presents the best average BDRATE results for classes B, C and E. For class D, which consists on very low resolution images (416 × 240), the GOSP method performed slightly worse than original HEVC standard, as evidenced by the positive BDRATE values. In fact, all the evaluated methods present very small BDRATE gains for class D, being most values inferior to 1%. There are also some images in classes B, C and E, where GOSP performs slightly inferior than SLSP. Such limited performance of GOSP method for these images, as well as class B images, can be explained by the higher number of bits used to signal the intra prediction modes. The bitrate overhead given by geometric transformations

Table 7.6: BDRATE results for test set 2 using proposed HEVC algorithm with generalised sparse optimal predictors based on LASSO method, compared with other related prediction methods.

Image	**LSP**	**TM**	**LLE**	**SLSP**	**GOSP**
(E) Johnny	0,21	-2,31	-2,53	-1,95	**-5,34**
(E) KristenAndSara	0,17	**-1,69**	-1,54	-1,65	-1,08
(E) FourPeople	-0,14	-0,84	-0,79	**-0,89**	-0,56
Average	0,08	-1,61	-1,62	-1,50	**-2,33**
(D) BasketballPass	-0,41	-1,17	-0,52	**-1,20**	0,03
(D) BlowingBubbles	-0,05	-0,05	-0,03	**-0,17**	0,70
(D) BQSquare	0,05	-0,35	-0,09	**-0,40**	0,61
(D) RaceHorses	0,15	0,21	0,38	**0,00**	0,58
Average	-0,07	-0,34	-0,07	**-0,44**	0,48
(C) BQMall	-0,05	-0,68	-0,72	**-0,78**	-0,39
(C) BasketballDrill	-0,50	-2,61	-2,51	-2,96	**-4,05**
(C) PartyScene	-0,10	-0,49	-0,57	**-0,62**	-0,07
(C) RaceHorses	-0,15	-0,24	-0,05	**-0,24**	-0,14
Average	-0,20	-1,01	-0,96	-1,15	**-1,16**
(B) BasketballDrive	-0,26	-2,39	-2,99	-2,16	**-3,63**
(B) BQTerrace	-0,17	-1,73	**-2,69**	-1,55	-2,53
(B) Cactus	-0,03	-1,43	-1,55	-1,35	**-1,62**
(B) Kimono1	-0,01	-0,06	**-0,18**	-0,14	-0,01
(B) ParkScene	-0,03	-0,18	-0,20	**-0,27**	0,06
Average	-0,10	-1,16	-1,52	-1,09	**-1,55**

(5 bits) and the filter size D_f (1 bit average for uniform distribution) results in a total of 6 bits to represent each intra prediction mode, being larger than the 5 bit word used in the other evaluated HEVC-based methods to signal the angular modes and linear prediction technique. The GOSP uses CABAC context modelling to encode the filter size D_f symbol, so its representation can be inferior to 1 bit, when its distribution is non-uniform. However, for most test images with smaller resolutions, the CABAC encoder may not be able to learn the symbol statistics. In these situations, a slight performance penalty may occur for GOSP method. However, it is important to notice that for most images where GOSP is not the best solution, the performance difference is insignificant, being far below to 1%.

Despite the referred limitations, mainly for small images, the GOSP method provides significant RD gains for several test images, specially for the Johnny, BasketballDrive and BasketballDrill. These results demonstrate the relevance of the proposed solution based on LASSO method and geometric transformations, over the directional prediction framework of HEVC standard, as well as the other compared algorithms using SLSP, LLE, TM and LSP methods. Note that, unlike the compared algorithms, that combine HEVC directional prediction with linear prediction methods, the proposed GOSP solution replaces the intra prediction framework of HEVC standard (only keeping Planar and DC modes). However, GOSP may provide similar results to directional prediction when filter size $D_f = 1$ is selected.

In general, the results of Tables 7.5 and 7.6 show that proposed prediction framework may significantly improve the RD coding performance of HEVC for most test images. In the worst scenario, the proposed method provides a RD performance similar to the one of the HEVC standard. The main issue of GOSP is the higher signalling cost for the intra prediction modes, which may limit the provided coding gains, mainly for small images. A possible solution is to use a binary representation for geometric transformations, alternative to the fixed-length-code of 5 bits. Although the used binary representation can be acceptable when $D_f = 1$ is used, we believe that a more efficient binary representation could applied when larger filters ($D_f = 31$) are used. This is because larger filters tend to capture dependencies between farthest pixels, resulting in output predicted blocks less susceptible to high intensive deformations, as given by some of the used geometric transformations.

Regarding the computational complexity of the proposed method, there is, inevitably, a higher number of operations because it needs to solve the LASSO problem and apply geometric transformations. The impact of these computations is higher in the encoder side, where it tests all the prediction possibilities for different block sizes. In this work, we did not particularly address the computational complexity problem, but the experiments showed that coding times may increase two orders of magnitude. However we believe that the method could be largely improved by using more efficient algorithmic implementations, as well as parallel processing architectures.

7.6 Conclusions

In this chapter we developed a generalised intra prediction framework that unifies the HEVC angular prediction modes with the linear prediction method. In Section 7.1 an alternative interpretation of directional prediction was presented based on first-order linear prediction filters combined with geometric transformations. Such interpretation provided new insights for the design of directional prediction, motivating the proposal of a generalised intra prediction framework, as described in Section 7.2. In the proposed framework, the first-order linear filters related to directional prediction were replaced by optimal sparse models adaptively estimated in the causal neighbourhood of the block.

In Section 7.3 we briefly described the predictor selection algorithms investigated in this work for sparse linear modelling. These methods included the OMP algorithm, which provides a greedy solution for predictor selection, as well as less greedy solutions, such as the LAR and LASSO algorithms. The Elastic Net method, which provides a grouping effect in predictor selection, was also investigated. Section 7.4 presented the proposed prediction algorithm for HEVC standard, explaining some aspects related to the stopping criteria of model selection methods and used filter context size.

The experimental tests presented in Section 7.5 revealed that sparse model estimation using greedy algorithms tends to result in a lower prediction performance. Conversely, the use of LAR and LASSO methods for sparse model estimation showed to provide a more efficient image prediction performance. Different stopping criteria methods were also evaluated for the proposed sparse model estimation algorithms. The results demonstrated that using a fixed number of predictors along the encoding procedure is an efficient solution. When compared to the directional prediction framework, as well as other linear prediction methods proposed in literature, the use of the proposed prediction framework in HEVC standard showed significant coding improvements for most images, specially for images with complex textures and repeated structures. The presented results allowed a better understanding of the strengths and limitations of directional and linear prediction methods and showed that sparsity is the key to unify them.

Chapter 8

Conclusions and future work

This thesis investigated new efficient intra prediction methods for image and video coding using the current state-of-the-art compression algorithms. The importance of these techniques is justified by the spread of digital image and video contents, as well as the emerging 3D video formats and high video resolutions, which impose increased bitrate requirements. Some of the most recent video compression algorithms used in this thesis include the state-of-the-art HEVC standard, the 3D extension of HEVC for multiview video plus depth coding, known as 3D-HEVC, and the alternative MMP algorithm based on the pattern-matching paradigm, which has shown a noticeable performance for image coding in the last years. In the following, the chapters of this thesis are briefly summarised, presenting the main conclusions and suggested future work regarding each proposed contribution.

In Chapter 1, after a brief introduction describing the main motivations for this research work, the objectives, contributions and outline of this thesis were presented.

Chapter 2 described the main prediction techniques investigated in this thesis, in particular, the directional prediction methods and motion-compensated-prediction used in current state-of-the-art compression algorithms, the linear prediction methods based on least-squares regression, and some dictionary-based prediction methods using sparse representations.

Chapter 3 presented a description of the main 2D and 3D image and video compression algorithms used in this thesis, namely the HEVC standard, the MMP algorithm and the 3D extension of HEVC for multiview video plus depth coding. Regarding the 2D codecs, a comparison between HEVC, MMP and the previous H.264/AVC standard was performed, in order to assess the performance of these algorithms for image and video coding. The performed experiments revealed some issues of MMP algorithm for still image coding, when compared to the transform-based HEVC standard, motivating the research of improved prediction techniques for MMP. Regarding the 3D codecs, an experiment using the 3D-HEVC and MV-HEVC standards for the compression of multiview video plus depth data was presented.

The results demonstrated the importance of the new coding tools incorporated in 3D-HEVC standard specifically for the compression of depth maps.

Chapter 4 described the first contribution of this thesis, which proposed an enhanced intra prediction framework for the MMP algorithm. Instead of using only 8 directional modes, a more advanced prediction framework based on 33 angular modes, similar to the ones of HEVC standard, was investigated for the MMP paradigm. The use of more accurate prediction methods for still image coding in MMP algorithm tends to produce a residual signal with a more concentrated probability distribution, which favours the adaptation of MMP dictionary, leading consequently to a more efficient compression performance. The directional prediction was optimised for the flexible block partitioning scheme of MMP algorithm, for instance, by disabling less important modes at certain block sizes. The entropy coding of directional modes was also improved using adaptive mode pruning, based on the smoothness of neighbouring samples. In order to better exploit the higher number of prediction directions, larger block sizes were also introduced in MMP algorithm. In the proposed approach, both 64 × 64 and 16 × 16 block sizes can be used as starting points for the flexible block partitioning scheme of MMP algorithm.

The experimental evaluation of the MMP algorithm demonstrated the advantage of the proposed improvements for its intra prediction framework. When compared to the transform-based coding standards for still image coding, the new MMP algorithm presented significant coding gains over the H.264/AVC and a quite competitive performance in relation to the HEVC algorithm, presenting an average result very close to the one of the state-of-the-art solution. For some images, in particular the synthetic images, the new MMP algorithm achieved the highest coding performance. Experiments for stereo image coding showed that the improvements to the MMP algorithm are equally important for independent and dependent images, achieving a coding performance similar to the one of the recent MV-HEVC standard.

As suggestion for future work, more efficient solutions could be investigated for the intra prediction framework of the MMP algorithm. The high computational complexity burden is a well-known limitation of these methods that belong to the family of pattern-matching coding paradigm. When a higher number of prediction modes is used, the computational complexity further increases due to the larger amount of generated residual blocks that need to be tested by the MMP residue coding method. The investigation of faster sub-optimal methods to select the intra prediction modes is a possible solution to mitigate this issue. Also, the use of larger block sizes significantly increases the computational complexity of MMP algorithm, mainly due to the higher number of possible partitioning combinations of a block. This problem could be minimised by combining the flexible block partitioning scheme with the quadtree block partitioning, by limiting the minimum block size of flexible partitioning for each quadtree level.

Chapter 5 presented the second contribution of this thesis, a new highly predictive encoder for the compression of depth map signals, denominated as Predictive Depth Coding algorithm (PDC). PDC was motivated by the investigated prediction methods for the MMP algorithm, combining the directional prediction with a flexible block partitioning scheme. A simplified depth modelling mode is also used to complement directional prediction modes, specifically when the depth block contains edges that cannot be well predicted from the causal neighbouring samples. For the compression of residue signal, a straightforward linear approximation method is employed. In this method, most residuals are encoded using a null pattern, while a few residuals for a specific set of prediction modes can be encoded using non-null linear functions.

The presented PDC algorithm is an enhanced version of a preliminary depth map encoder, previously investigated in this work. This preliminary encoder presented several designing differences relative to the final proposed PDC algorithm, mostly regarding to the flexible block partitioning scheme and the residue coding method. Although this algorithm provided a superior RD performance than most depth map coding algorithms presented in literature, it have not competed with the recently standardised 3D-HEVC standard. The proposed PDC algorithm, not only improved the coding performance of the original method, but also reduced its overall computational complexity.

The performance of the PDC algorithm was assessed for the three view coding scenario and compared to the state-of-the-art 3D-HEVC standard, specifically developed for the compression of multiview video plus depth data. Experiments were performed using the SSE and VSO methods for evaluation of the depth map distortion. The VSO method was originally developed for depth map coding in 3D-HEVC, in order to optimise the encoded depth maps for view synthesis purposes. This method is extremely important when using the recommended evaluation procedure by MPEG, based on the quality of the texture views synthesised using the encoded depth maps. The experimental results using the all-intra coding configuration demonstrated the superiority of PDC algorithm over 3D-HEVC standard, using either SSE or VSO methods for distortion measure. An additional experiment, based on the common test conditions, using a modified version of 3D-HEVC, where the anchor I-frames were encoded by PDC algorithm, confirmed the relevance of the proposed depth map intra coding algorithm.

For future work, PDC algorithm could be extended to perform inter prediction, specifically to exploit temporal, inter-view and inter-component dependencies. One interesting inter prediction solution would be to combine motion-compensated prediction with the flexible block partitioning scheme of PDC algorithm. Another interesting research line is the investigation of some of the proposed prediction techniques for the 3D-HEVC standard, such as the constrained depth modelling mode or the adaptive pruning of directional modes based on the smoothness of neighbouring causal pixels.

In Chapter 6, an alternative class of prediction methods based on linear prediction and sparse representation was investigated to further improve the coding performance of current state-of-the-art image compression standards. First, the LLE-based prediction method was researched for efficient prediction of 3D holoscopic images based on the HEVC compression technology, resulting in the third contribution of this thesis. The LLE-based prediction method consists in a linear combination of block patches which are implicitly estimated in the causal reconstructed image. Such estimation is performed in both encoder and decoder sides using the causal neighbouring template area adjacent to the unknown block. This kind of prediction method, based on previously encoded image patches, is interesting for holoscopic image prediction, because these signals present a strong spatial correlation among neighbouring micro-images. Experimental tests using the proposed HEVC-based holoscopic image encoder demonstrated significant coding gains, when compared to other methods proposed in literature for the compression of 3D holoscopic contents.

There are some suggestions of future work that could be exploited to further improve the performance of the proposed holoscopic image encoder. Regarding the causal search window, where the predictors of LLE-based prediction method are selected, several searching constraints could be used, not only to reduce computational complexity, but also to improve coding performance, by preventing inefficient predictors from being selected using the implicit template matching method. These constraints on the search procedure could be defined based on the grid structure of micro-images that characterise the holoscopic image, specifically considering the micro-image dimensions. Another suggestion to improve the algorithm performance is to combine the LLE-based prediction method with the explicit Block Matching Algorithm for predictor selection, as proposed by the Self-Similarity method. These methods have different characteristics that could complement each other when used together.

As a second contribution of Chapter 6, the sparse-LSP method for still image prediction was developed. Sparse-LSP is a linear prediction technique which defines a filter context in a large causal area, where only a few positions may use non-null coefficients, forming a sparse filter context. It can be regarded as an extension of the LLE-based prediction method, allowing to predict one pixel using neighbouring causal pixels, similarly to traditional LSP methods, as well as more distant causal pixels, similarly to the LLE-based prediction method. The predictor selection method of sparse-LSP is the k-NN algorithm as used in LLE-based prediction method. Experimental tests using the proposed sparse-LSP method combined with directional prediction in HEVC standard for still image coding showed to improve the overall coding performance of HEVC, specially for those images with textured areas made of repeated structures. When compared to LLE-based prediction method, sparse-LSP demonstrated a superior RD performance for many test images.

An interesting future work for investigation would be to extend the sparse-LSP method for inter-prediction coding. This could be done by augmenting the filter context to the previous encoded frames. Such an approach could provide an implicit solution for motion compensation, similarly to template matching methods, with the advantage of combining several spatial and temporal predictors either from one or several reference frames. As sparse-LSP generalises the LSP, TM and LLE-based prediction methods an interesting future work would be to develop a flexible version of sparse-LSP method able to evaluate different filter orders and context area constraints during the encoding procedure, in order to operate similarly to other prediction methods. The optimal filter order and context area parameters for each encoded block could be explicitly transmitted to the decoder. Another topic for future work is the research of less computationally complex implementations of sparse-LSP method.

Chapter 7 presented the last contribution of this thesis, a new intra prediction framework that generalises the main prediction techniques investigated along this research work, namely the directional and linear prediction methods. The proposed method results from an alternative interpretation of directional prediction, where each angular mode is regarded as the output of a first-order linear prediction filter deformed by a geometric transformation procedure. Such an interpretation of directional prediction led us to the generalised prediction framework, which replaced the first-order filters by sparse linear filters, optimally estimated in a causal training window. In this chapter, several methods for predictor selection and estimation were investigated and evaluated for image prediction. Due to its greedy characteristic, the sparse models estimated by OMP algorithm showed to present some issues for image prediction. On the other hand, the less greedy LAR and LASSO methods demonstrated to achieve more efficient prediction results, in particular when using a predefined number of non-null coefficients.

The proposed intra prediction framework was incorporated in the HEVC standard, by replacing the angular prediction modes. In order to signal the selected geometric transformation in the proposed method, the bitstream syntax for angular modes in the HEVC algorithm was used. Furthermore, an additional signalling bit was introduced in order to explicitly select either the minimal filter size (the first-order filter) or the maximum filter size based on sparse modelling. Experimental tests using the new prediction framework in the HEVC algorithm for still image coding showed to outperform the original HEVC standard based on directional prediction, for most cases. When compared to other methods based on linear prediction and sparse representation, including the sparse-LSP method proposed in Chapter 6, the new prediction solution presented the best performance results for many test images, specially for the ones with complex textures and repeated patterns.

Although the presented prediction scheme provided good results, there are some topics for future work that could be addressed to improve the overall compression performance. For instance, the signalling of the geometric transformations could be enhanced, taking into account the estimated linear filter context. The characteristics of the linear filter, *e.g.*, the distance between the unknown pixel and the estimated predictors, could be used to determine the expected probability of each geometric transformation, which also depends on the intensity of the caused deformation. An study could be performed to find out the probabilities of each geometric transformation, for different types of filter contexts, and thus optimise their entropy coding using CABAC algorithm. Another research topic is the investigation of alternative stopping criteria for the LASSO method used for adaptive predictor selection. The use of explicit solutions, that signal the number of selected predictors to the decoder could be investigated. An interesting future work would be also to extend this method for inter image prediction, by augmenting the filter context across previous encoded frames. The use of the proposed prediction scheme with other compression algorithms, such as MMP, could also be investigated in the future.

As a summary, one may conclude that the use of new efficient prediction methods in current state-of-the-art image and video compression algorithms may provide significant coding gains. In particular, the alternative MMP algorithm was significantly improved by using a larger number of directional prediction modes. The directional prediction was also shown to be relevant for depth map coding when combined with a flexible block partitioning scheme. Additionally, linear prediction methods based on sparse representation demonstrated to provide a state-of-the-art coding performance using the HEVC standard. The achieved results shed an interesting light over the importance of intra prediction, allowing a better understanding of the strengths and limitations of both directional and linear prediction techniques from an image coding point of view.

Appendix A

Test signals

This appendix illustrates some of the original test signals used in simulations throughout the thesis, namely test stereo images, grayscale natural test images and holoscopic test images.

A.1 Test stereo images

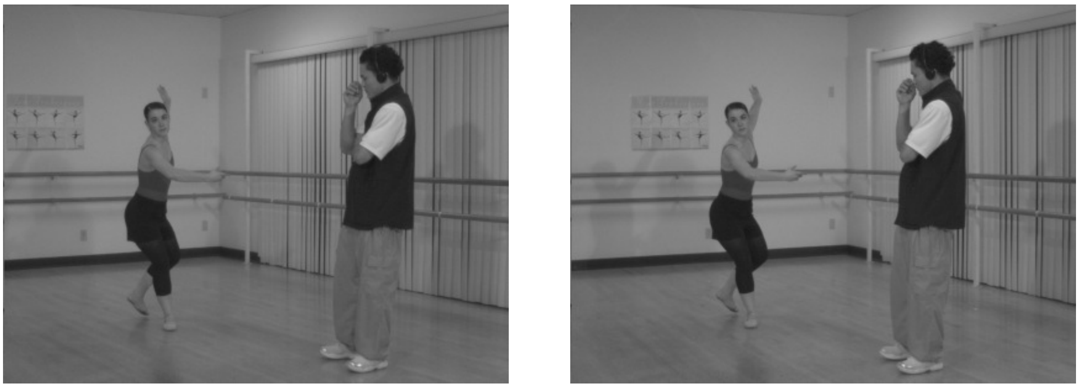


Figure A.1: Stereo test image Ballet, cameras 5 and 3 (1024 × 768)

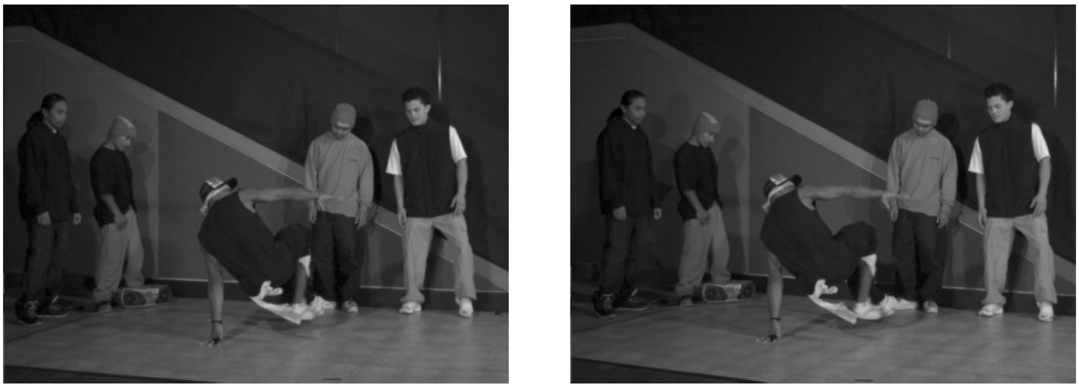


Figure A.2: Stereo test image Breakdancers, cameras 5 and 3 (1024 × 768)

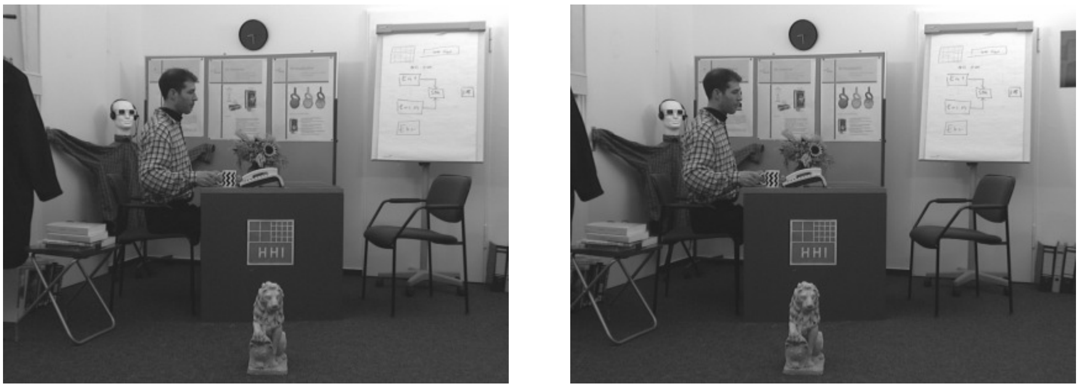


Figure A.3: Stereo test image Book Arrival, cameras 10 and 8 (1024 × 768)

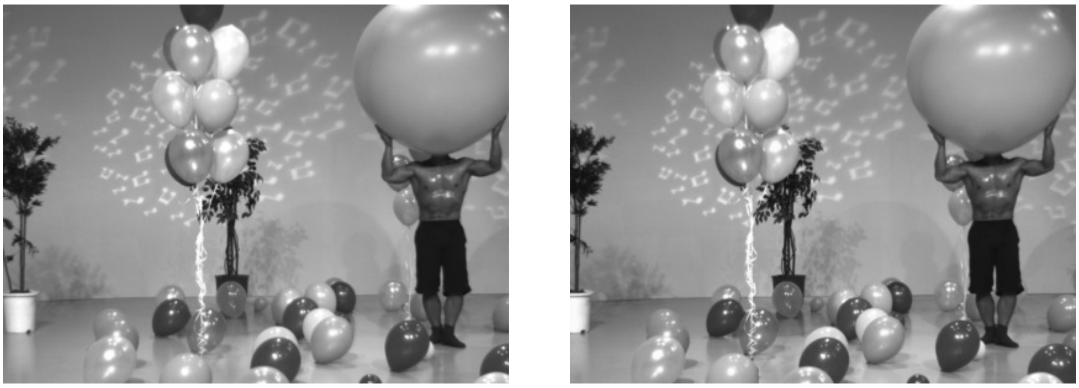


Figure A.4: Stereo test image Balloons, cameras 3 and 5 (1024 × 768)

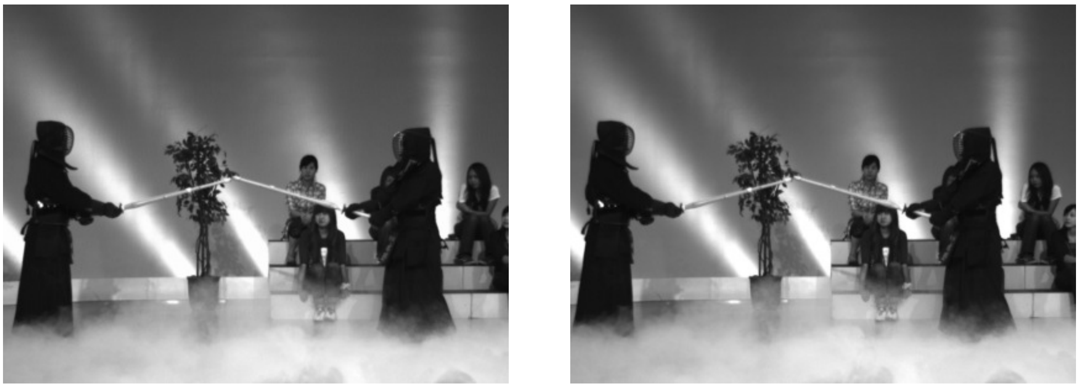


Figure A.5: Stereo test image Kendo, cameras 3 and 5 (1024 × 768)

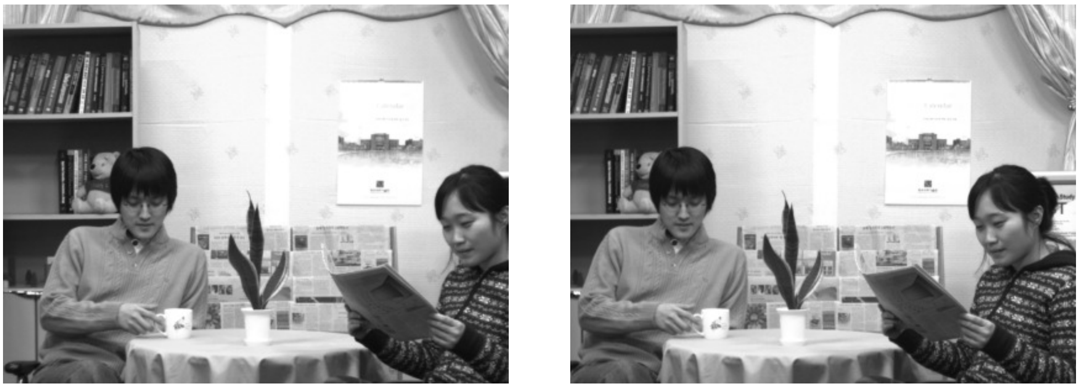


Figure A.6: Stereo test image Newspaper, cameras 4 and 6 (1024 × 768)

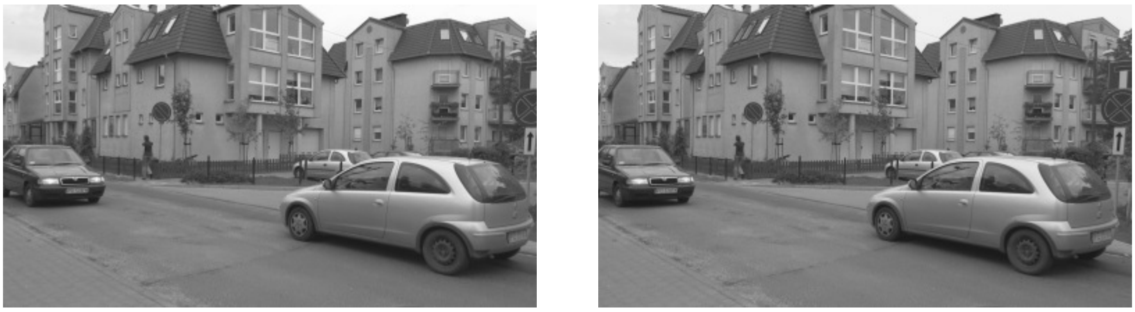


Figure A.7: Stereo test image Poznan Street, cameras 4 and 3 (1920 × 1088)

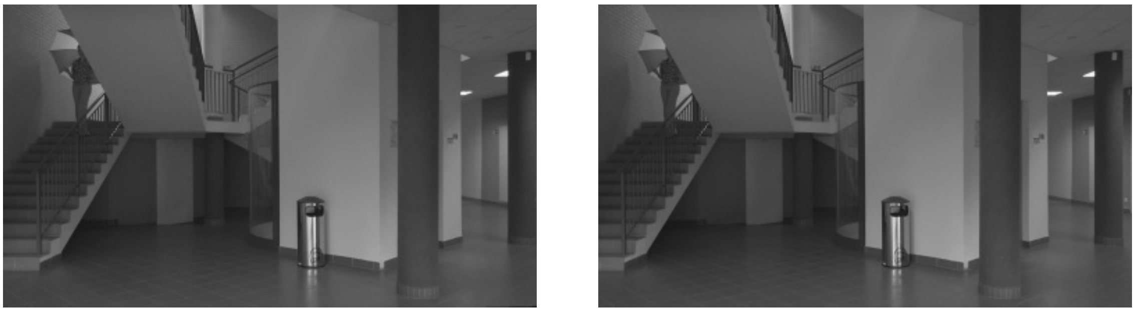


Figure A.8: Stereo test image Poznan Hall2, cameras 7 and 6 (1920 × 1088)

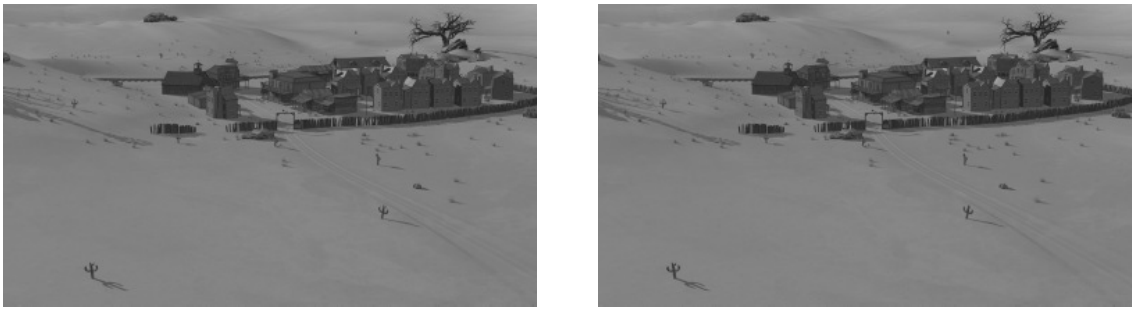


Figure A.9: Stereo test image GT Fly, cameras 5 and 1 (1920 × 1088)

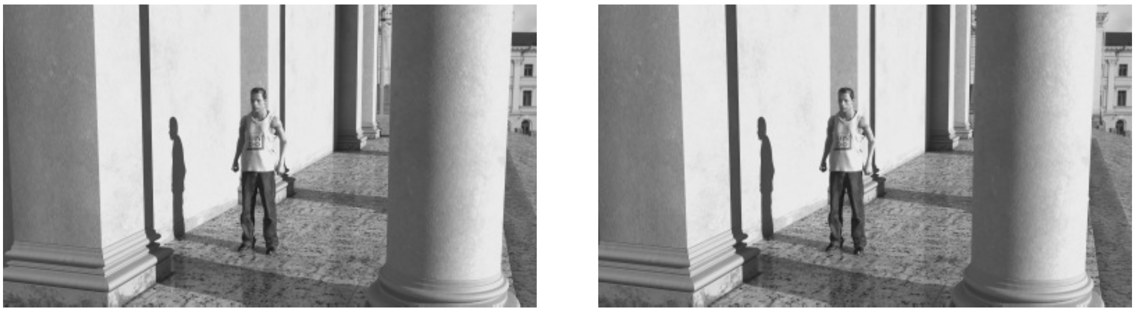


Figure A.10: Stereo test image Undo Dancer, cameras 5 and 9 (1920 × 1088)

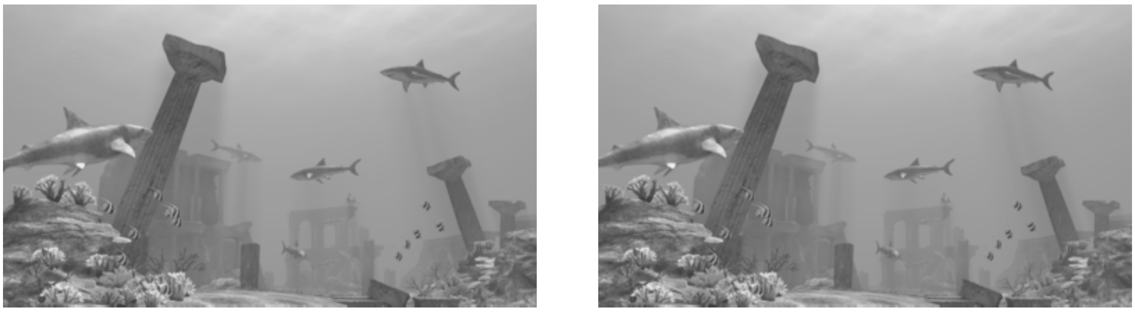


Figure A.11: Stereo test image Shark, cameras 1 and 5 (1920 × 1088)

A.2 Test images

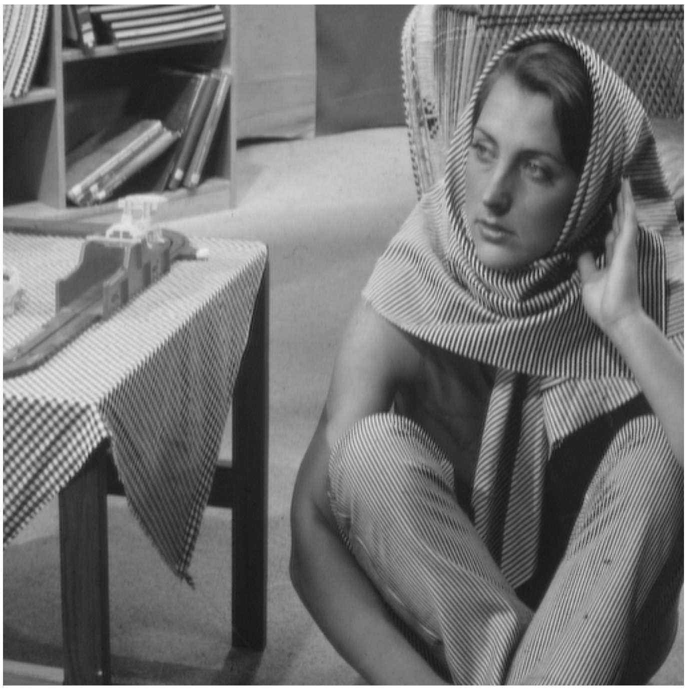


Figure A.12: Grayscale natural test image Barbara (512 × 512).

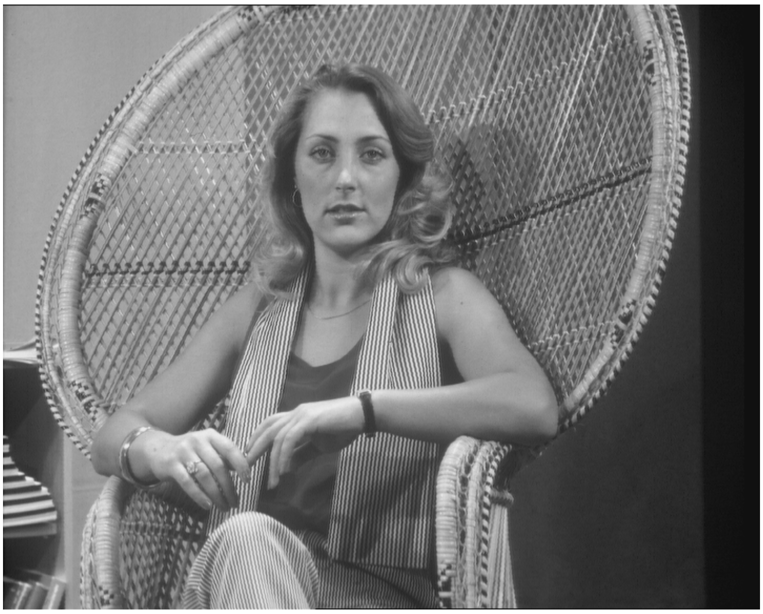


Figure A.13: Grayscale natural test image Barbara2 (720 × 576).

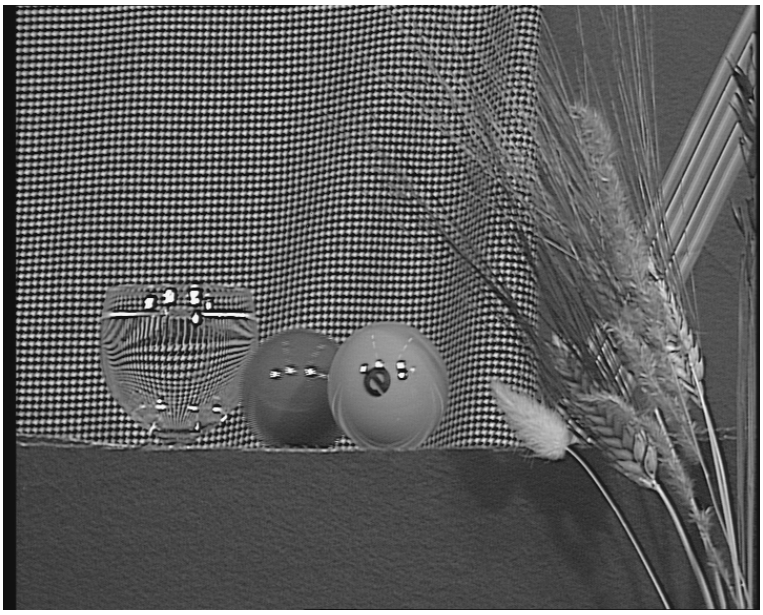


Figure A.14: Grayscale natural test image Snook (720 × 576).

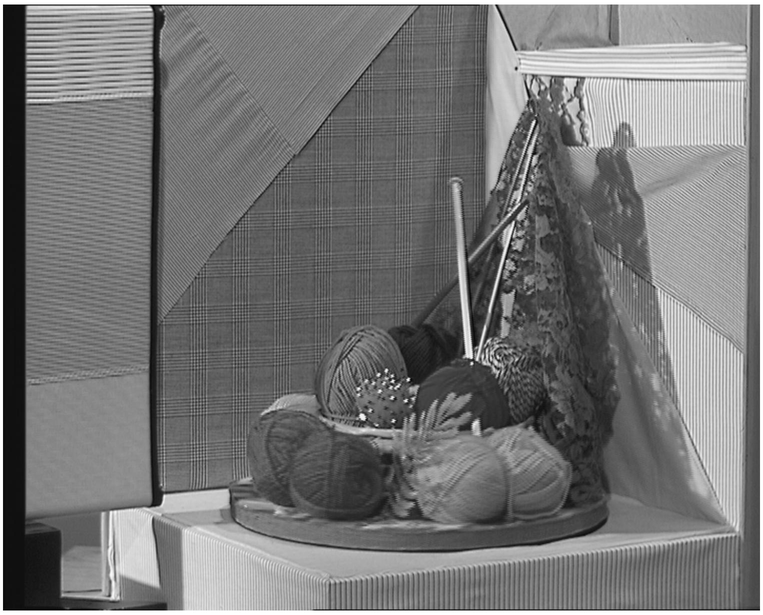


Figure A.15: Grayscale natural test image Wool (720 × 576).

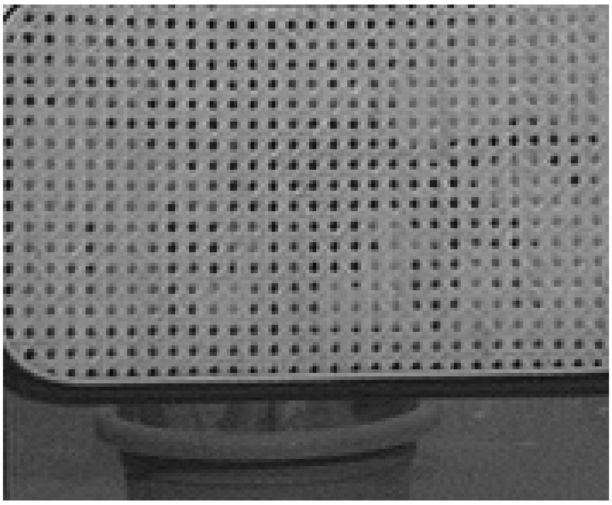


Figure A.16: Grayscale natural test image Pan0_qcif (176 × 144).

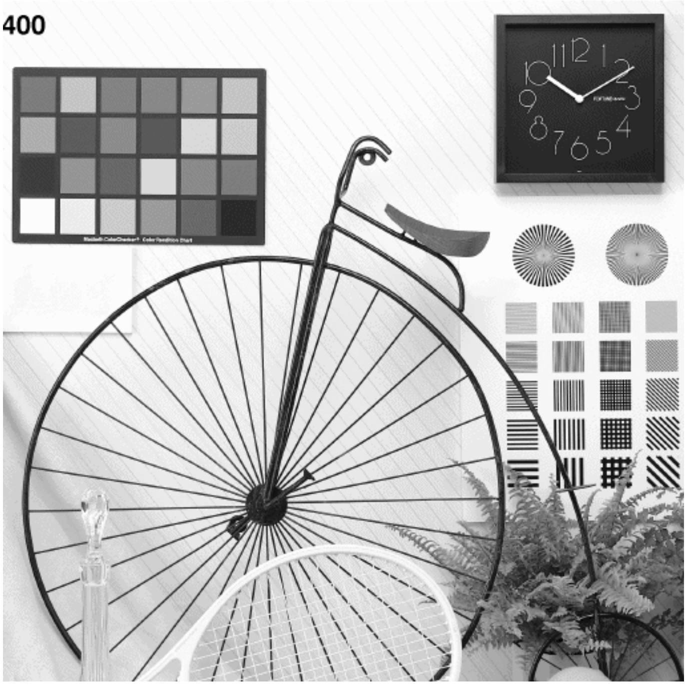


Figure A.17: Grayscale natural test image Bike (512 × 512).

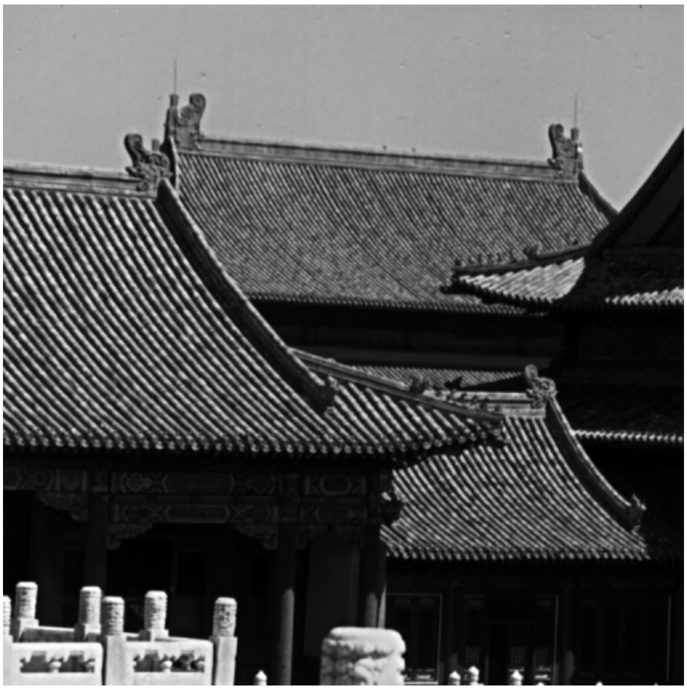


Figure A.18: Grayscale natural test image Roof (512 × 512).

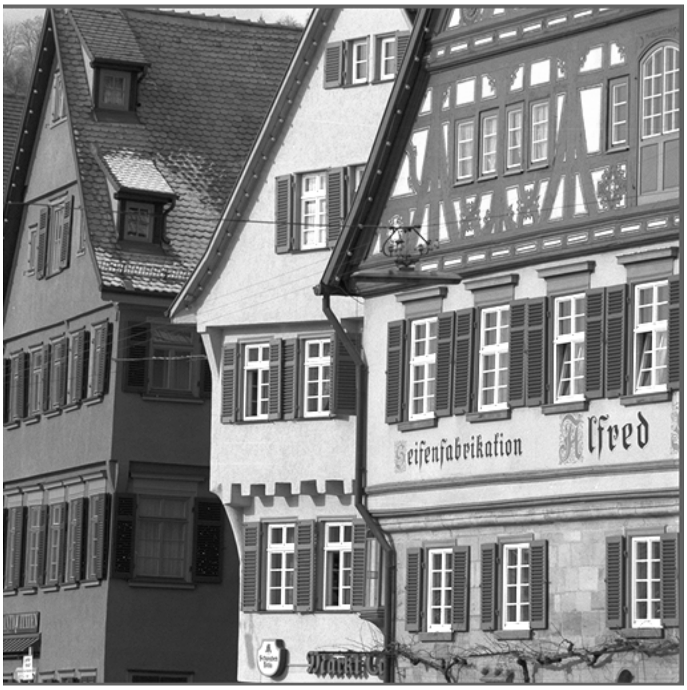


Figure A.19: Grayscale natural test image Houses (512 × 512).

A.3 Holoscopic images

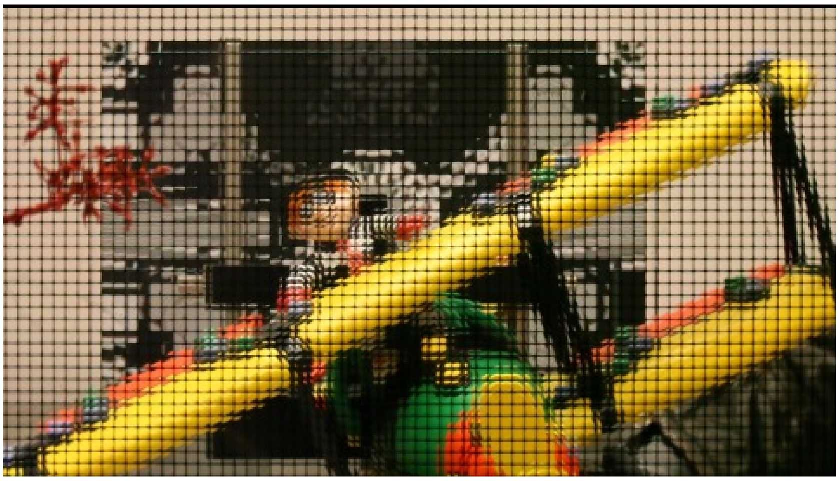


Figure A.20: First frame of holoscopic test image Plane and Toy (1920 × 1088)

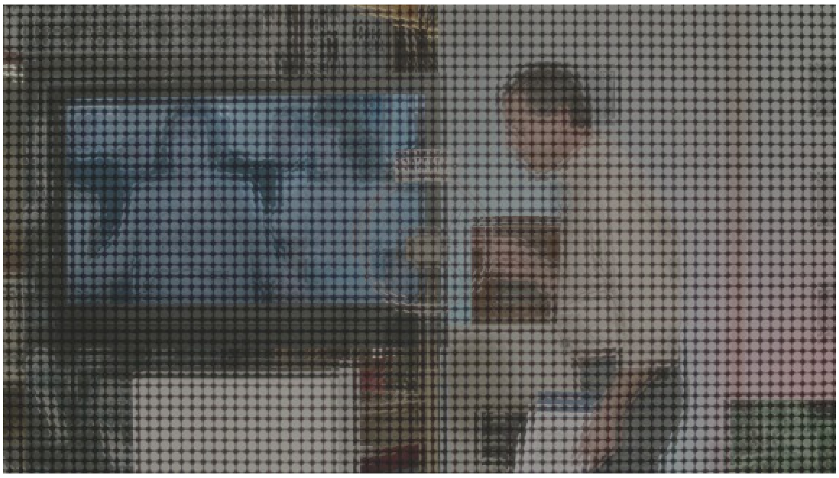


Figure A.21: First frame of holoscopic test image Demichelis Cut (2880 × 1620)

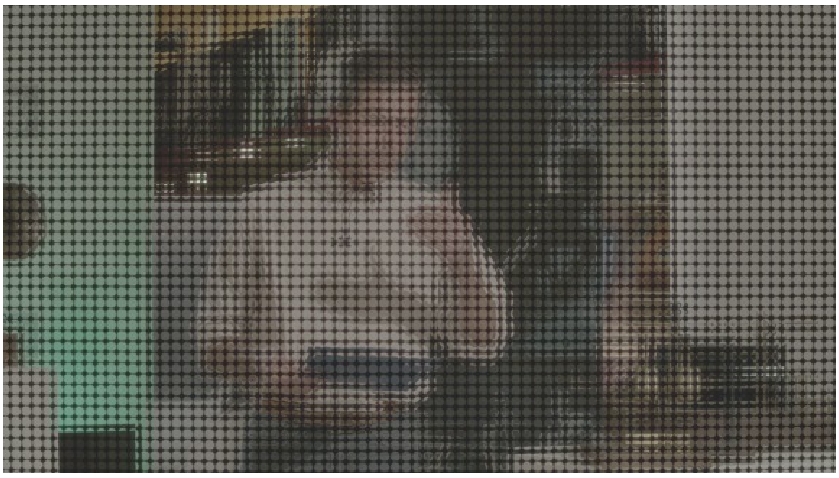


Figure A.22: First frame of holoscopic test image Demichelis Spark (2880 × 1620)

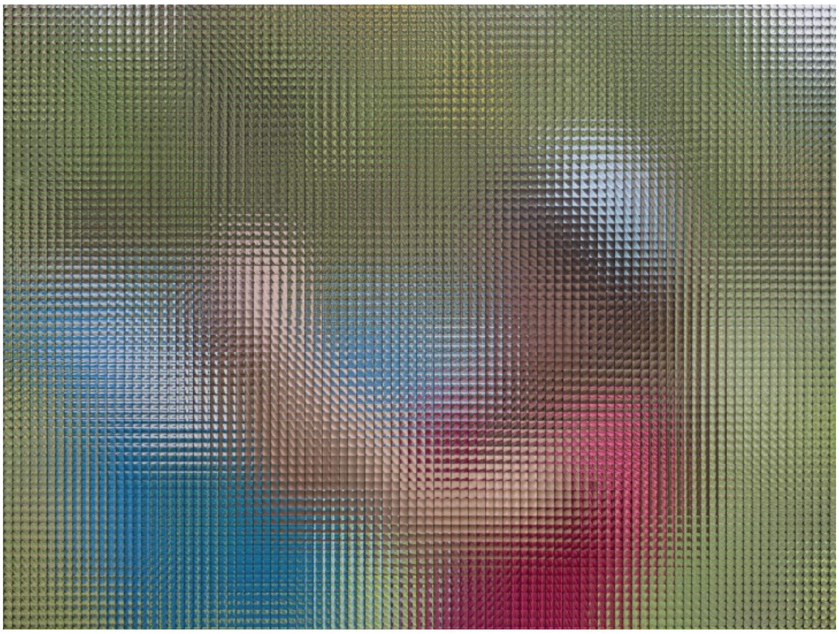


Figure A.23: Holoscopic test image Laura (7240 × 5432)

Appendix B

Published papers

This appendix presents a list of the published papers that resulted from the research work done during this thesis proposal. Submitted papers are also presented.

B.1 Journal papers

J1. Lucas, L. F. R.; Wegner, K.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Intra Predictive Depth Map Coding using Flexible Block Partitioning", *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4055-4068, November 2015.

B.2 Conference papers

C1. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Efficient depth map coding using linear residue approximation and a flexible prediction framework", In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '12*, pp. 1305-1308, Orlando, USA, September 2012.

C2. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Codificação eficiente de mapas de profundidade com base em predição e aproximação linear", In: *Proceedings of the XXX Simpósio Brasileiro de Telecomunicações, SBrT '12*, Brasília, Brazil, September 2012.

C3. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Improving the Multidimensional Multiscale Parser algorithm for intra image coding", In: *Proceedings of the Conference of Telecommunications, Conftele '13*, Castelo Branco, Portugal, May 2013.

C4. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Predictive depth map coding for efficient view synthesis", In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '13*, pp. 2058-2062, Melbourne, VIC, Australia, September 2013.

C5. Lucas, L. F. R.; Conti, C.; Nunes, P.; Soares, L. D.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC", In: *Proceedings of the European Signal Processing Conference, EUSIPCO '14*, pp. 11-15, Lisbon, Portugal, September 2014.

C6. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Improving the emergent 3D-HEVC standard with a highly predictive depth map coding algorithm", In: *Proceedings of the Conference of Telecommunications 2015, Conftele '15*, Aveiro, Portugal, September 2015.

C7. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Sparse least-squares prediction for intra image coding", In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '15*, pp. 1115-1119, Quebec City, QC, Canada, September 2015.

B.3 MPEG documents

D1. Lucas, L. F. R.; Wegner, K.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Intra depth-map coding using flexible segmentation, constrained depth modeling modes and simplified/pruned directional prediction", *ISO/IEC JTC 1/SC 29/WG 11*, Sapporo, Japan, 7-11 July 2014.

B.4 Submitted journal papers

S1. Lucas, L. F. R.; Rodrigues, N. M. M.; da Silva, E. A. B.; Pagliari, C. L.; de Faria, S. M. M., "Image coding using generalized optimal predictors", *submitted to IEEE Transactions on Image Processing.*

S2. Lucas, L. F. R.; Rodrigues, N. M. M.; Pagliari, C. L.; da Silva, E. A. B.; de Faria, S. M. M., "Recurrent Pattern Matching Based Stereo Image Coding Using Linear Predictors", *submitted to Multidimensional Systems and Signal Processing, Springer.*

Bibliography

[1] ITU-T, ISO/IEC JTC1. *High efficiency video coding.* ITU-T Recommendation H.265 and ISO/IEC 23008-2, April 2013 (and subsequent editions).

[2] SULLIVAN, G., OHM, J., HAN, W.-J., et al. "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1649–1668, December 2012.

[3] ITU-T. *Video codec for audiovisual services at p × 64 kbit/s.* ITU-T Recommendation H.261, March 1993.

[4] ITU-T, ISO/IEC JTC1. *Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video.* ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2), July 1995.

[5] ITU-T, ISO/IEC JTC1. *Advanced video coding for generic audiovisual services.* ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), May 2003 (and subsequent editions).

[6] WIEGAND, T., SULLIVAN, G., BJNTEGAARD, G., et al. "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 560–576, July 2003.

[7] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "On Dictionary Adaptation for Recurrent Pattern Image Coding", *IEEE Transactions on Image Processing*, v. 17, n. 9, pp. 1640–1653, September 2008.

[8] FRANCISCO, N., RODRIGUES, N., DA SILVA, E., et al. "Scanned Compound Document Encoding Using Multiscale Recurrent Patterns", *IEEE Transactions on Image Processing*, v. 19, n. 10, pp. 2712–2724, October 2010.

[9] GRAZIOSI, D. B., RODRIGUES, N., DA SILVA, E., et al. "Improving Multiscale Recurrent Pattern Image Coding with Least-Squares Prediction". In:

Proceedings of the IEEE International Conference on Image Processing, ICIP '09, pp. 2813–2816, Cairo, Egypt, November 2009.

[10] LUCAS, L., RODRIGUES, N., DE FARIA, S., et al. "Intra-prediction for color image coding using YUV correlation". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '10*, pp. 1329–1332, Hong Kong, China, September 2010.

[11] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Multiscale Recurrent Pattern Predictive Image Coding with Template Matching". In: *Proceedings of the Conference on Telecommunications - ConfTele '07*, Peniche, Portugal, May 2007.

[12] LUCAS, L., RODRIGUES, N., DA SILVA, E., et al. "Adaptive least squares prediction for stereo image coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '11*, pp. 2013–2016, Brussels, Belgium, September 2011.

[13] ITU-R. *Digital methods of transmitting television information: encoding parameters of digital television for studios.* ITU-R Recommendation BT.601, 1987.

[14] CUTLER, C. "Differential quantization of communication signals". US Patent 2,605,361, July 1952. Available at: <`http://www.google.com/patents/US2605361`>. [Accessed: January 5th, 2016].

[15] SHANNON, C. E. "A mathematical theory of communication", *The Bell System Technical Journal*, v. 27, n. 3, pp. 379–423, 1948.

[16] MITRA, S. *Digital Signal Processing.* 3 ed. New York, NY, McGraw-Hill Science/Engineering/Math, 2005.

[17] MARAGOS, P., SCHAFER, R., MERSEREAU, R. "Two-dimensional linear prediction and its application to adaptive predictive coding of images", *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 32, n. 6, pp. 1213–1229, December 1984.

[18] WEINBERGER, M. J., SEROUSSI, G., SAPIRO, G. "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS", *IEEE Transactions on Image Processing*, v. 9, pp. 1309–1324, August 2000.

[19] WU, X., MEMON, N. "Context-based adaptive lossless image coding", *IEEE Transactions on Communications*, v. 45, n. 4, pp. 437–444, April 1997.

[20] MATSUDA, I., OZAKI, N., UMEZU, Y., et al. “Lossless coding using variable block-size adaptive prediction optimized for each image”. In: *Proceedings of the European Signal Processing Conference, EUSIPCO ’05*, Antalya, Turkey, September 2005.

[21] LI, X., ORCHARD, M. “Edge-directed prediction for lossless compression of natural images”, *IEEE Transactions on Image Processing*, v. 10, n. 6, pp. 813–817, June 2001.

[22] NGUYEN, T., MARPE, D. “Performance analysis of HEVC-based intra coding for still image compression”. In: *Proceedings of the Picture Coding Symposium, PCS ’12*, pp. 233–236, Krakow, Poland, May 2012.

[23] MÜLLER, K., SCHWARZ, H., MARPE, D., et al. “3D High-Efficiency Video Coding for Multi-View Video and Depth Data”, *IEEE Transactions on Image Processing*, v. 22, n. 9, pp. 3366–3378, September 2013.

[24] VETRO, A., WIEGAND, T., SULLIVAN, G. “Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard”, *Proceedings of the IEEE*, v. 99, n. 4, pp. 626–642, April 2011.

[25] WU, X., BARTHEL, K.-U., ZHANG, W. “Piecewise 2D autoregression for predictive image coding”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’98*, v. 3, pp. 901–904, Chicago, IL, USA, October 1998.

[26] WEINLICH, A., REHM, J., AMON, P., et al. “Massively parallel lossless compression of medical images using least-squares prediction and arithmetic coding”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’13*, pp. 1680–1684, Melbourne, VIC, Australia, September 2013.

[27] KAU, L.-J., LIN, Y.-P. “Adaptive lossless image coding using least squares optimization with edge-look-ahead”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 52, n. 11, pp. 751–755, November 2005.

[28] YE, H., DENG, G., DEVLIN, J. C. “A weighted least squares method for adaptive prediction in lossless image compression”. In: *Proceedings of the Picture Coding Symposium, PCS ’03*, pp. 489–493, Saint-Malo, France, April 2003.

[29] GARCIA, D., DE QUEIROZ, R. “Least-Squares Directional Intra Prediction in H.264/AVC”, *IEEE Signal Processing Letters*, v. 17, n. 10, pp. 831–834, October 2010.

[30] CHEN, J., HAN, W.-J. “Adaptive linear prediction for block-based lossy image coding”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, pp. 2833–2836, Cairo, Egypt, November 2009.

[31] LI, X. “Least-square prediction for backward adaptive video coding”, *EURASIP Journal on Applied Signal Processing*, v. 2006, n. 1, pp. 126–126, 2006.

[32] LI, X. “Video Processing Via Implicit and Mixture Motion Models”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 17, n. 8, pp. 953–963, 2007.

[33] LEE, S., CHO, N. “Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, pp. 1037–1040, Cairo, Egypt, November 2009.

[34] ZHANG, X., GISQUET, C., FRANCOIS, E., et al. “Chroma Intra Prediction Based on Inter-Channel Correlation for HEVC”, *IEEE Transactions on Image Processing*, v. 23, n. 1, pp. 274–286, January 2014.

[35] MARTIN, A., FUCHS, J.-J., GUILLEMOT, C., et al. “Sparse representation for image prediction”. In: *Proceedings of European Signal Processing Conference, EUSIPCO ’07*, pp. 1255–1259, Poznan, Poland, September 2007.

[36] TURKAN, M., GUILLEMOT, C. “Sparse approximation with adaptive dictionary for image prediction”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, pp. 25–28, Cairo, Egypt, November 2009.

[37] TURKAN, M., GUILLEMOT, C. “Image prediction: Template matching vs. sparse approximation”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’10*, pp. 789–792, Hong Kong, China, September 2010.

[38] TURKAN, M., GUILLEMOT, C. “Image Prediction Based on Neighbor-Embedding Methods”, *IEEE Transactions on Image Processing*, v. 21, n. 4, pp. 1885–1898, April 2012.

[39] CHERIGUI, S., GUILLEMOT, C., THOREAU, D., et al. “Correspondence Map-Aided Neighbor Embedding for Image Intra Prediction”, *IEEE Transactions on Image Processing*, v. 22, n. 3, pp. 1161–1174, March 2013.

[40] TURKAN, M., GUILLEMOT, C. “Dictionary Learning for Image Prediction”, *Journal of Visual Communication and Image Representation*, v. 24, n. 3, pp. 426–437, April 2013.

[41] MALLAT, S., ZHANG, Z. “Matching pursuits with time-frequency dictionaries”, *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3397–3415, December 1993.

[42] PATI, Y., REZAIIFAR, R., KRISHNAPRASAD, P. S. “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition”. In: *Proceedings of the Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, v. 1, pp. 40–44, Pacific Grove, CA, USA, November 1993.

[43] TAN, T., BOON, C., SUZUKI, Y. “Intra Prediction by Template Matching”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’06*, pp. 1693–1696, Atlanta, GA, USA, October 2006.

[44] TAN, T. K., BOON, C. S., SUZUKI, Y. “Intra Prediction by Averaged Template Matching Predictors”. In: *Proceedings of the IEEE Consumer Communications and Networking Conference, CCNC ’07*, pp. 405–409, Las Vegas, NV, USA, January 2007.

[45] ZHENG, Y., YIN, P., ESCODA, O., et al. “Intra prediction using template matching with adaptive illumination compensation”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’08*, pp. 125–128, San Diego, CA, USA, October 2008.

[46] YU, S.-L., CHRYSAFIS, C. *New intra prediction using intra-macroblock motion compensation*. Joint Video Team of MPEG and VCEG, Document JVT-C151, May 2002.

[47] CHERIGUI, S., GUILLEMOT, C., THOREAU, D., et al. “Hybrid template and block matching algorithm for image intra prediction”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP ’12*, pp. 781–784, Kyoto, Japan, March 2012.

[48] SULLIVAN, G., WIEGAND, T. “Rate-distortion optimization for video compression”, *IEEE Signal Processing Magazine*, v. 15, n. 6, pp. 74–90, November 1998.

[49] GRAZIOSI, D., RODRIGUES, N., DA SILVA, E., et al. “Lossy and lossless image encoding using multi-scale recurrent pattern matching”, *IET Image Processing*, v. 7, n. 6, pp. 556–566, August 2013.

[50] GRAZIOSI, D. B. *Contribuições à compressão de imagens com e sem perdas utilizando recorrência de padrões multiescalas*. PhD thesis, COPPE - Universidade Federal do Rio de Janeiro, April 2011.

[51] DUARTE, M., CARVALHO, M., DA SILVA, E., et al. "Multiscale recurrent patterns applied to stereo image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 15, n. 11, pp. 1434–1447, November 2005.

[52] GRAZIOSI, D. B., RODRIGUES, N. M. M., PAGLIARI, C. L., et al. "Multiscale recurrent pattern matching approach for depth map coding". In: *Proceedings of the Picture Coding Symposium, PCS '10*, pp. 294–297, Nagoya, Japan, December 2010.

[53] FILHO, E. B. L., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Electrocardiographic Signal Compression using Multiscale Recurrent Patterns", *IEEE Transactions on Circuits and Systems I*, v. 52, n. 12, pp. 2739–2753, December 2005.

[54] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Improving H.264/AVC Inter Compression with Multiscale Recurrent Patterns". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '06*, pp. 1353–1356, Atlanta, GA, USA, October 2006.

[55] FRANCISCO, N., RODRIGUES, N., DA SILVA, E., et al. "Efficient Recurrent Pattern Matching Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 8, pp. 1161–1173, August 2012.

[56] FRANCISCO, N., RODRIGUES, N., DA SILVA, E., et al. "Video compression using 3D multiscale recurrent patterns". In: *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '13*, pp. 1412–1415, Beijing, China, May 2013.

[57] FRANCISCO, N. C. *Contribuições à codificação eficiente de imagem e vídeo utilizando recorrência de padrões multiescala*. PhD thesis, COPPE - Universidade Federal do Rio de Janeiro, November 2012.

[58] XU, J., JOSHI, R., COHEN, R. "Overview of the Emerging HEVC Screen Content Coding Extension", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 26, n. 1, pp. 50–62, January 2016.

[59] CAO, X., LAI, C., WANG, Y., et al. "Short Distance Intra Coding Scheme for High Efficiency Video Coding", *IEEE Transactions on Image Processing*, v. 22, n. 2, pp. 790–801, February 2013.

[60] PIAO, Y., MIN, J., CHEN, J. *Encoder improvement of unified intra prediction*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-C207, 3rd Meeting: Guangzhou, China, October 2010.

[61] BROSS, B., KIRCHHOFFER, H., SCHWARZ, H., et al. *Fast intra encoding for fixed maximum depth of transform quadtree*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-C311, 3rd Meeting: Guangzhou, China, October 2010.

[62] PURNACHAND, N., ALVES, L., NAVARRO, A. "Improvements to TZ search motion estimation algorithm for multiview video coding". In: *Proceedings of the IEEE International Conference on Systems, Signals and Image Processing, IWSSIP '12*, pp. 388–391, Vienna, Austria, April 2012.

[63] MARPE, D., SCHWARZ, H., WIEGAND, T. "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 620–636, 2003.

[64] LIU, S., XU, X., LEI, S., et al. "Overview of HEVC extensions on screen content coding", *APSIPA Transactions on Signal and Information Processing*, v. 4, pp. 1–12, September 2015.

[65] JOSHI, R., LIU, S., SULLIVAN, G., et al. *High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 5*. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-V1005, 22nd Meeting: Geneva, Switzerland, October 2015.

[66] FRANCISCO, N. C., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Multiscale Recurrent Pattern Image Coding With a Flexible Partition Scheme". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '08*, pp. 141–144, San Diego, CA, USA, October 2008.

[67] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Universal Image Coding using Multiscale Recurrent Patterns and Pre-

diction". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '05*, v. 2, pp. 245–248, Genoa, Italy, September 2005.

[68] WITTEN, I. H., NEAL, R. M., CLEARY, J. G. "Arithmetic Coding for Data Compression", *Communications of the ACM*, v. 30, n. 6, pp. 520–540, June 1987.

[69] BOSSEN, F. *Common HM test conditions and software reference configurations.* Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-L1100, 12th Meeting: Geneva, Switzerland, January 2013.

[70] BJONTEGAARD, G. "Calculation of Average PSNR Differences Between RD-curves", *ITU-T SG 16 Q.6 Video Coding Experts Group (VCEG), Document VCEG-M33*, April 2001.

[71] "JM Reference Software for H.264/AVC". 2012. Available at: <`http://iphome.hhi.de/suehring/tml/download/old_jm/`>. [Accessed: January 5th, 2016].

[72] "HM Reference Software for HEVC". 2013. Available at: <`https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/`>. [Accessed: January 5th, 2016].

[73] PENNEBAKER, W., MITCHEL, J. *JPEG: Still Image Data Compression Standard.* 1 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 1992.

[74] TAUBMAN, D. S., MARCELIN, M. *JPEG2000: Image Compression Fundamentals, Standards and Practice.* 2 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 2001.

[75] CHEN, Y., TECH, G., WEGNER, K., et al. *Test Model 10 of 3D-HEVC and MV-HEVC.* Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-J1003, 10th Meeting: Strasbourg, France, October 2014.

[76] SMOLIC, A., KIMATA, H., VETRO, A. "Development of MPEG Standards for 3D and Free Viewpoint Video". In: *Proceedings of the SPIE Conference Optics East 2005: Communications, Multimedia & Display Technologies*, v. 6016, pp. 262–273, November 2005.

[77] KUBOTA, A., SMOLIC, A., MAGNOR, M., et al. “Multiview Imaging and 3DTV”, *IEEE Signal Processing Magazine*, v. 24, n. 6, pp. 10–21, November 2007.

[78] TANIMOTO, M., TEHRANI, M., FUJII, T., et al. “Free-Viewpoint TV”, *IEEE Signal Processing Magazine*, v. 28, n. 1, pp. 67–76, January 2011.

[79] KAUFF, P., ATZPADIN, N., FEHN, C., et al. “Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability”, *Signal Processing: Image Communication*, v. 22, n. 2, pp. 217–234, February 2007.

[80] ISO/IEC JTC1/SC29/WG11 (MPEG). *Report on Experimental Framework for 3D Video Coding.* Document N11631, Guangzhou, China, October 2010.

[81] SMOLIC, A., MÜLLER, K., MERKLE, P., et al. “An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution”. In: *Proceedings of the Picture Coding Symposium, PCS ’09*, pp. 1–4, Chicago, IL, USA, May 2009.

[82] STELMACH, L., TAM, W. J., MEEGAN, D., et al. “Stereo image quality: effects of mixed spatio-temporal resolution”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 10, n. 2, pp. 188–193, March 2000.

[83] VETRO, A. “Frame compatible formats for 3D video distribution”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’10*, pp. 2405–2408, Hong Kong, China, September 2010.

[84] MORVAN, Y. *Acquisition, compression and rendering of depth and texture for multi-view video.* PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2009.

[85] CHEN, Y., WANG, Y.-K., UGUR, K., et al. “The emerging MVC standard for 3D video services”, *EURASIP Journal on Advances in Signal Processing*, v. 2009, pp. 1–13, January 2008.

[86] MERKLE, P., SMOLIC, A., MÜLLER, K., et al. “Efficient Prediction Structures for Multiview Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 17, n. 11, pp. 1461–1473, November 2007.

[87] SULLIVAN, G., BOYCE, J., CHEN, Y., et al. "Standardized Extensions of High Efficiency Video Coding (HEVC)", *IEEE Journal of Selected Topics in Signal Processing*, v. 7, n. 6, pp. 1001–1016, December 2013.

[88] KONRAD, J., HALLE, M. "3-D Displays and Signal Processing", *IEEE Signal Processing Magazine*, v. 24, n. 6, pp. 97–111, November 2007.

[89] READ, J. C. "Viewer experience with stereoscopic 3D television in the home", *Displays*, v. 35, n. 5, pp. 252 – 260, December 2014.

[90] SAKAMOTO, Y., HAYASHI, N., KATO, A., et al. "3D Holo TV System Based on Multi-view Images". In: *Proceedings of the IEEE/ACIS International Conference on Computer and Information Science, ICIS '10*, pp. 831–835, Yamagata, Japan, August 2010.

[91] MÜLLER, K., MERKLE, P., WIEGAND, T. "3-D Video Representation Using Depth Maps", *Proceedings of the IEEE*, v. 99, n. 4, pp. 643–656, April 2011.

[92] SMOLIC, A., MÜLLER, K., DIX, K., et al. "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '08*, pp. 2448–2451, San Diego, CA, USA, October 2008.

[93] KAWAKITA, M., IIZUKA, K., NAKAMURA, H., et al. "High-definition real-time depth-mapping TV camera: HDTV Axi-Vision Camera", *Optics Express*, v. 12, n. 12, pp. 2781–2794, June 2004.

[94] SCHARSTEIN, D., SZELISKI, R. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *International Journal of Computer Vision*, v. 47, pp. 7–42, April 2002.

[95] MERKLE, P., MORVAN, Y., SMOLIC, A., et al. "The effects of multiview depth video compression on multiview rendering", *Signal Processing: Image Communication*, v. 24, pp. 73–88, January 2009.

[96] MORVAN, Y., DE WITH, P. H. N., FARIN, D. "Platelet-based coding of depth maps for the transmission of multiview images". In: *Proceedings of the SPIE Stereoscopic Displays and Virtual Reality Systems XIII*, v. 6055, pp. 93–100, San Jose, CA, USA, January 2006.

[97] SARKIS, M., ZIA, W., DIEPOLD, K. "Fast Depth Map Compression and Meshing with Compressed Tritree". In: *Proceedings of the 9th Asian Conference on Computer Vision, ACCV '09*, v. 5995, pp. 44–55, 2010.

[98] FEHN, C. "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV". In: *Proceedings of the SPIE Stereoscopic Displays and Virtual Reality Systems XI*, v. 5291, pp. 93–104, San Jose, CA, USA, January 2004.

[99] MÜLLER, K., SMOLIC, A., DIX, K., et al. "Reliability-based generation and view synthesis in layered depth video". In: *Proceedings of the IEEE Workshop on Multimedia Signal Processing, MMSP '08*, pp. 34–39, Cairns, Qld, Australia, October 2008.

[100] BOURGE, A., GOBERT, J., BRUL, F. "MPEG-C PART 3: Enabling the introduction of video plus depth contents". In: *Proceedings of the Workshop on Content Generation and Coding for 3D-Television*, Eindhoven, The Netherlands, June 2006.

[101] LIPPMANN, G. "Epreuves Reversibles Donnant la Sensation du Relief", *Journal de Physique Théorique et Appliquée*, v. 7, n. 1, pp. 821–825, November 1908.

[102] LUCAS, L., CONTI, C., NUNES, P., et al. "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC". In: *Proceedings of the European Signal Processing Conference, EUSIPCO '14*, pp. 11–15, Lisbon, Portugal, September 2014.

[103] ELHARAR, E., STERN, A., HADAR, O., et al. "A Hybrid Compression Method for Integral Images Using Discrete Wavelet Transform and Discrete Cosine Transform", *Journal of Display Technology*, v. 3, n. 3, pp. 321–325, September 2007.

[104] AGGOUN, A. "Compression of 3D Integral Images Using 3D Wavelet Transform", *Journal of Display Technology*, v. 7, n. 11, pp. 586–592, November 2011.

[105] OLSSON, R. "Empirical Rate-Distortion Analysis of JPEG 2000 3D and H.264/AVC Coded Integral Imaging Based 3D-Images". In: *Proceedings of the 3DTV Conference*, pp. 113–116, Istanbul, Turkey, May 2008.

[106] CONTI, C., LINO, J., NUNES, P., et al. "Spatial prediction based on self-similarity compensation for 3D holoscopic image and video coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '11*, pp. 961–964, Brussels, Belgium, September 2011.

[107] CONTI, C., NUNES, P., SOARES, L. "New HEVC prediction modes for 3D holoscopic video coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '12*, pp. 1325–1328, Orlando, FL, USA, September 2012.

[108] CHEN, Y., HANNUKSELA, M. M., SUZUKI, T., et al. "Overview of the MVC+D 3D video coding standard", *Journal of Visual Communication and Image Representation*, v. 25, n. 4, pp. 679–688, May 2014.

[109] HANNUKSELA, M. M., CHEN, Y., SUZUKI, T., et al. *3D-AVC Draft Text 8*. Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-F1002, 6th Meeting: Geneva, Switzerland, October 2013.

[110] TECH, G., WEGNER, K., CHEN, Y., et al. *3D-HEVC Draft Text 6*. Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-J1001, 10th Meeting: Strasbourg, France, October 2014.

[111] SCHWARZ, H., BARTNIK, C., BOSSE, S., et al. "Extension of High Efficiency Video Coding (HEVC) for multiview video and depth data". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '12*, pp. 205–208, Orlando, FL, USA, September 2012.

[112] MÜLLER, K., VETRO, A. *Common Test Conditions of 3DV Core Experiments*. Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document JCT3V-G1100, 7th Meeting: San Jose, CA, USA, January 2014.

[113] "HTM Reference Software for 3D-HEVC". 2015. Available at: <`https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/`>. [Accessed: January 5th, 2016].

[114] LUCAS, L. F. R. *Using Least-Squares Prediction for Compression of Colour and Stereo Images*. MSc thesis, Escola Superior de Tecnologia e Gestão - Instituto Politécnico de Leiria, June 2011.

[115] RODRIGUES, N. *Multiscale Recurrent Pattern Matching Algorithms for Image and Video Coding*. PhD thesis, Universidade de Coimbra, October 2008.

[116] LUCAS, L. F. R., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Stereo image coding using dynamic template-matching prediction". In: *Proceed-*

ings of the IEEE International Conference on Computer as a Tool, EUROCON '11, pp. 1–4, Lisbon, Portugal, April 2011.

[117] FRANCISCO, N. C., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "A generic post-deblocking filter for block based image compression algorithms", *Signal Processing: Image Communication*, v. 27, n. 9, pp. 985–997, October 2012.

[118] KRISHNAMURTHY, R., CHAI, B., TAO, H., et al. "Compression and transmission of depth maps for image-based rendering". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '01*, v. 3, pp. 828–831, Thessaloniki, Greece, October 2001.

[119] MERKLE, P., MÜLLER, K., WIEGAND, T. "Coding of depth signals for 3D video using wedgelet block segmentation with residual adaptation". In: *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '13*, pp. 1–6, San Jose, CA, USA, July 2013.

[120] MERKLE, P., MÜLLER, K., MARPE, D., et al. "Depth Intra Coding for 3D Video based on Geometric Primitives", *IEEE Transactions on Circuits and Systems for Video Technology*, February 2015.

[121] MERKLE, P., BARTNIK, C., MÜLLER, K., et al. "3D video: Depth coding based on inter-component prediction of block partitions". In: *Proceedings of the Picture Coding Symposium, PCS '12*, pp. 149–152, Krakow, Poland, May 2012.

[122] MÜLLER, K., MERKLE, P., TECH, G., et al. "3D video coding with depth modeling modes and view synthesis optimization". In: *Proceedings of the Asia-Pacific Signal Information Processing Association, 2012 Annual Summit and Conference, APSIPA '12*, pp. 1–4, Hollywood, CA, USA, December 2012.

[123] JAGER, F. "Simplified depth map intra coding with an optional depth lookup table". In: *Proceedings of the International Conference on 3D Imaging, IC3D '12*, pp. 1–4, Liège, Belgium, December 2012.

[124] ZITNICK, C., KANG, S., UYTTENDAELE, M., et al. "High-quality video view interpolation using a layered representation", *ACM SIGGRAPH and ACM Transactions on Graphics*, pp. 600–608, August 2004.

[125] FELDMANN, I., MUELLER, M., ZILLY, F., et al. *HHI Test Material for 3D Video.* ISO/IEC JTC1/SC29/WG11 (MPEG), Document M15413, Archamps, France, April 2008.

[126] TANIMOTO, M., FUJII, T., FUKUSHIMA, N. *1D Parallel Test Sequences for MPEG-FTV*. ISO/IEC JTC1/SC29/WG11 (MPEG), Document M15378, Archamps, France, April 2008.

[127] TANIMOTO, M., FUJII, T., SUZUKI, K. *View synthesis algorithm in view synthesis reference software 2.0 (VSRS2.0)*. ISO/IEC JTC1/SC29/WG11 (MPEG), Document M16090, Lausanne, Switzerland, February 2009.

[128] "Platelet results". 2012. Available at: <`http://vca.ele.tue.nl/demos/mvc/PlateletDepthCoding.tgz`>. [Accessed: January 5th, 2016].

[129] WEGNER, K., STANKIEWICZ, O., TANIMOTO, M., et al. *Enhanced View Synthesis Reference Software (VSRS) for Free-viewpoint Television*. ISO/IEC JTC1/SC29/WG11 (MPEG), Document M31520, October 2013.

[130] LAINEMA, J., BOSSEN, F., HAN, W.-J., et al. "Intra Coding of the HEVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1792–1801, December 2012.

[131] NATARAJAN, B. K. "Sparse Approximate Solutions to Linear Systems", *SIAM Journal on Computing*, v. 24, n. 2, pp. 227–234, April 1995.

[132] PATI, Y., REZAIIFAR, R., KRISHNAPRASAD, P. S. "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition". In: *Proceedings of the Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, Pacific Grove, CA, USA, November 1993.

[133] EFRON, B., HASTIE, T., JOHNSTONE, I., et al. "Least angle regression", *Annals of Statistics*, v. 32, n. 2, pp. 407–499, 2004.

[134] TIBSHIRANI, R. "Regression Shrinkage and Selection Via the Lasso", *Journal of the Royal Statistical Society, Series B*, v. 58, pp. 267–288, 1996.

[135] ZOU, H., HASTIE, T. "Regularization and variable selection via the Elastic Net", *Journal of the Royal Statistical Society, Series B*, v. 67, pp. 301–320, 2005.

[136] CHEN, S. S., DONOHO, D. L., SAUNDERS, M. A. "Atomic Decomposition by Basis Pursuit", *SIAM Journal on Scientific Computing*, v. 20, n. 1, pp. 33–61, August 1998.

[137] AKAIKE, H. "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, v. 19, n. 6, pp. 716–723, December 1974.

[138] SCHWARZ, G. "Estimating the Dimension of a Model", *The Annals of Statistics*, v. 6, pp. 461–464, 1978.

[139] MALLOWS, C. L. "Some comments on C_p", *Technometrics*, v. 15, n. 4, pp. 661–675, November 1973.

[140] STONE, M. "Cross-validatory choice and assessment of statistical predictions", *Journal of the Royal Statistical Society, Series B*, v. 36, n. 2, pp. 111–147, 1974.

[141] VALDMAN, C., DE CAMPOS, M., APOLINARIO, J. "A geometrical stopping criterion for the LAR algorithm". In: *Proceedings of the European Signal Processing Conference, EUSIPCO '12*, pp. 2104–2108, Bucharest, Romania, August 2012.