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

CONTRIBUIÇÕES À COMPRESSÃO DE IMAGENS COM E SEM PERDAS UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Danillo Bracco Graziosi

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
Abril de 2011

CONTRIBUIÇÕES À COMPRESSÃO DE IMAGENS COM E SEM PERDAS UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Danillo Bracco Graziosi

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. Weiler Alves Finamore, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
ABRIL DE 2011

Graziosi, Danillo Bracco

Contribuições à compressão de imagens com e sem perdas utilizando recorrência de padrões multiescalas/Danillo Bracco Graziosi. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIX, 247 p.: il.; $29,7$cm.

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

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2011.

Referências Bibliográficas: p. 229 – 247.

1. Casamento de Padrões Multiescala. 2. Compressão de images estáticas. 3. Compressão com perdas. 4. Compressão sem perdas. 5. Compressão multivistas. 6. Predição em imagens. 7. Imagens estereoscópicas. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Aos meus pais.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador, Prof. Eduardo A. B. da Silva, pela sua grande capacidade de orientação e por todo o trabalho que desenvolvemos em conjunto. Trabalhar com alguém tão brilhante é com certeza muito motivador, e isso claramente se reflete no volume de trabalho desenvolvido nos últimos anos de tese. Além disso, gostaria de agradecer a oportunidade que me foi oferecida de ir trabalhar em Portugal. Essa mudança foi fundamental para a conclusão do meu Doutorado, e além de ter me ajudado a focar no trabalho, me mostrou um canto do mundo pra mim desconhecido, onde consegui fazer a minha pesquisa e conhecer novos amigos. Por tudo isso, e muito além disso, Eduardo, meu muito obrigado.

Em seguida gostaria de agradecer aos meus amigos de Leiria, em especial ao meu orientador Dr. Sérgio M. M. de Faria e ao Dr. Nuno M. M. Rodrigues, que foram peças fundamentais nesta tese. Ao Nuno, pelo código do MMP, pela paciência de me explicar sua tese, e por todos os outros ótimos momentos, quando fui acolhido em sua casa ou quando demos umas voltinhas de bicicleta. Ao Sérgio, por ter sido quase que um pai pra mim (ele só é muito novo pra isso). Pelos conselhos e amizade, pela orientação no meu trabalho e também em muitos outros aspectos da minha vida. Meu muito obrigado.

Um agradecimento especial merecem o Prof. Murilo B. de Carvalho e a Profa. Carla L. Pagliari, que tanto me ajudaram com os capítulos de compressão sem perdas e compressão multivistas. As suas contribuições com certeza foram fundamentais para esta tese, e para as publicações que derivaram dela. Muito obrigado por terem gasto o seu tempo comigo em conversas sem fim no skype, e por terem me ajudado com fórmulas indecifráveis ou com artefatos invisíveis em imagens codificadas.

Gostaria de agradecer também os meus colegas do laboratório do Instituto de Telecomunicações, Diego, Luis, Nelson e Sylvain, pelas dicas no Latex, pela ajuda com o código, e por momentos dentro e fora do laboratório. Obrigado também aos colegas do LPS, que sempre me ajudaram, dos poucos momentos onde estive no laboratório no Rio, quanto das vezes que me ajudaram por email ou pelo Gtalk.

A minha família, meu muito obrigado pelo apoio incondicional que sempre mostraram por toda esta minha jornada. Meu tio Orlando, tia Sandra, meus primos

Luciana e Gustavo, meu irmão Guilherme, meus pais Angelo e Rita, toda a minha família sempre esteve do meu lado, e o seu apoio foi um combustível de ânimo para que eu completasse essa jornada.

De resto, gostaria de agradecer a todas as pessoas que diretamente ou indiretamente contribuiram para esta tese, de alguma forma ou de outra. Me desculpem se não menciono o nome de todos aqui, mas com certeza vocês foram lembrados, durante este meu percurso penoso, mas muito gratificante. A todos, os meus sinceros agradecimentos.

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

CONTRIBUIÇÕES À COMPRESSÃO DE IMAGENS COM E SEM PERDAS UTILIZANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Danillo Bracco Graziosi

Abril/2011

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

Programa: Engenharia Elétrica

O trabalho aqui desenvolvido visa investigar o paradigma de compressão de imagens utilizando recorrência de padrões em múltiplas escalas, também conhecido como **MMP** *(Multidimensional Multiscale Parser).*

A primeira contribuição desta tese é uma proposta para reduzir a complexidade computacional do MMP. Em seguida, foi incorporado ao MMP um preditor adaptativo baseado em mínimos quadrados através de uma adaptação não-trivial para blocos, aumentando o seu desempenho taxa-distorção para a codificação com perdas de imagens naturais.

No âmbito desta tese, avaliou-se também o desempenho de compressão sem perdas do algoritmo. Pela primeira vez, um limite teórico foi desenvolvido para o MMP, codificando imagens sem perdas. Mostramos que o algoritmo é capaz de atingir a entropia de uma fonte ergódica, sem memória, com alfabeto finito. A partir deste resultado, desenvolvemos métodos para melhorar a eficiência de compressão sem perdas do MMP.

Por fim, o MMP foi usado para codificar imagens multivistas e seus respectivos mapas de profundidade. O algoritmo se mostrou particularmente eficaz para os mapas de profundidade. Nesta tese propomos uma arquitetura de codificação de imagens multivistas baseada no MMP, com relativo sucesso para síntese de vistas virtuais.

Dessa forma, as contribuições desta tese ajudam o MMP a afirmar-se como um algoritmo de compressão de imagens com desempenho taxa-distorção estado-da-arte para vários tipos de imagem e para qualquer taxa desejada.

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

CONTRIBUTIONS TO LOSSY AND LOSSLESS IMAGE COMPRESSION USING MULTISCALE RECURRENT PATTERN MATCHING

Danillo Bracco Graziosi

April/2011

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

Department: Electrical Engineering

The aim of this thesis is to investigate the image compression paradigm using pattern matching at multiple scales, also known as **MMP** *(Multidimensional Multiscale Parser).*

The first contribution of this thesis is a proposal for reducing MMP's computational complexity, a well-known critical part of the algorithm. Next, a non-trivial adaptation for blocks of an adaptive predictor based on least-squares was incorporated into MMP, increasing its rate-distortion performance for lossy coding of smooth images.

In this thesis, the algorithm's compression performance for lossless image coding was also evaluated. For the first time a performance bound for MMP's lossless compression capability was developed, showing that the algorithm is able to reach the entropy of an ergodic, memoryless source, and also providing indication on how to improve the algorithm's lossless compression performance. Proposals for algorithm modifications were made, showing MMP's efficiency for lossless coding as well.

At last, MMP was used for coding multiview images and their respective depth maps. The algorithm proved to be particularly effective for depth map compression. In this thesis we propose a MMP-based multiview image coding architecture, that is able to successfully synthesize virtual views.

In this way, contributions of this thesis strengthen MMP's position as an image compression algorithm that achieves state-of-the-art rate-distortion performance for several different kinds of images at any desired rate.

Sumário

Lista de Figuras **xii**

Lista de Tabelas **xvii**

Lista de Símbolos **xviii**

1 Introdução **1**

1.1 História da codificação 1

1.2 Motivações 3

1.3 Organização da tese 4

1.4 Contribuições da tese 5

1.4.1 Contribuição para a codificação de imagens com perdas 5

1.4.2 Contribuição para codificação de imagens sem perdas 6

1.4.3 Contribuição para a codificação de imagens 3D 6

2 Codificador de imagens com casamento de padrões multiescala **7**

2.1 Casamento aproximado de padrões usando multiescalas 7

2.2 Reduzindo a complexidade computacional 11

2.2.1 Resultados experimentais do MMP-FAST 12

2.3 Conclusões 12

3 MMP utilizando predição com critério de mínimos quadrados **16**

3.1 Predição usando o critério dos mínimos quadrados 16

3.2 Incorporando o LSP ao MMP-FP 18

3.2.1 Adaptando o LSP para predição em blocos 18

3.3 Resultados experimentais 19

3.4 Conclusões 20

4 Compressão sem perdas usando o MMP **23**

4.1 Compressão de imagens sem perdas 23

4.2 Modificando o algoritmo MMP para codificação sem perdas 24

4.2.1 Predição em algoritmos sem perda 24

4.2.2 Restrição do histograma 26

4.2.3 Remapeamento do erro de predição 26

4.2.4 Malha de realimentação para o erro de predição 28

4.3 Resultados experimentais 28

4.4 Conclusões 30

5 Compressão multivistas usando o MMP 31

5.1 O novo formato para imagens 3D: textura com mapas de profundidade 31

5.2 Usando o MMP para codificar mapas de profundidade 33

5.2.1 Avaliação das vistas reconstruídas usando mapas de disparidade codificados 33

5.2.2 Restrição de bordas para codificação de mapas de disparidade 36

5.3 Usando o MMP para codificar conjuntamente textura e mapas de profundidade 39

5.3.1 Alocação ótima de bits entre textura e profundidade 39

5.3.2 Codificação conjunta de vistas e mapas de disparidade 39

5.4 Conclusões 40

6 Conclusão 42

6.1 Conclusões discutidas por contribuição 42

6.1.1 Codificador de imagens com casamento de padrões multiescala 42

6.1.2 MMP utilizando predição com critério de mínimos quadrados . 43

6.1.3 Compressão sem perdas usando o MMP 43

6.1.4 Compressão multivistas usando o MMP 44

6.2 Perspectivas futuras para o MMP e tópicos em aberto 45

A Introduction 47

A.1 Image compression using multiscale recurrent pattern matching . . . 47

A.2 Motivations 47

A.3 Thesis outline 48

A.4 Thesis contributions 49

A.4.1 Lossy image coding 49

A.4.2 Lossless image Coding 49

A.4.3 3D image coding 49

B Multidimensional multiscale parser algorithm 51

B.1 Multiscale recurrent pattern matching 51

B.2 MMP-Intra 57

B.2.1 MMP-II 60

B.2.2 MMP-FP 62

B.3 MMP computational complexity analysis 64
B.3.1 Formal derivation of MMP-FP's computational complexity . . 65
B.4 MMP-FAST 69
B.4.1 Experimental results 69
B.5 Conclusions 73

C Least-squares prediction in MMP 75
C.1 Image modeling 75
C.1.1 Least-squares predictors 76
C.2 Block implementation of LSP predictor 80
C.3 Incorporating LSP predictor into the MMP encoder 82
C.4 Experimental results for MMP-LSP 85
C.5 Conclusions 89

D Lossless image compression using MMP 93
D.1 Lossless compression 93
D.2 Reviews of state-of-the-art lossless compression algorithms 94
D.3 Experimental results and benchmark 105
D.4 Theoretical bounds on MMP's lossless performance 106
D.4.1 Proof of the convergence of $\lim_{n\to\infty} \frac{n_{\text{total}}}{n}$ 113
D.5 Proposal for enhancing MMP lossless image compression performance 114
D.5.1 Study on the effects of prediction for MMP lossless coding . . 115
D.5.2 Study of the effects of residue coding techniques for the lossless MMP 118
D.5.3 Comparison with state-of-the-art lossless algorithms 122
D.6 Conclusions 125

E Multiview image compression 126
E.1 Review of the state-of-the-art in multiview coding 126
E.1.1 The 3D operation chain: capturing, coding and displaying . . 127
E.2 Depth image based rendering 136
E.3 3D standardization 142

F Depth coding using MMP 146
F.1 Coding depth information with MMP 146
F.1.1 Evaluating coded depth maps 147
F.2 Edge-aware coding of depth maps 164
F.2.1 Edge identification algorithm 164
F.2.2 Edge coding restriction 166
F.2.3 Experimental results 166

F.3 Conclusions 172

G Texture and depth coding using MMP 173

G.1 Multiview image coding 173

G.2 Independent multiview coding using the MMP algorithm 174

G.2.1 Texture coding with MMP 175

G.2.2 Optimal bit allocation for independent depth and texture coding with MMP 175

G.3 Encoding texture and depth jointly through warping 183

G.3.1 Joint encoding architecture 183

G.3.2 Efficiency of the warped image as a predictor 184

G.4 Experimental results 190

G.4.1 Coding results 191

G.4.2 Subjective analysis 191

G.5 Conclusions 196

H Conclusions and perspectives 197

H.1 Conclusions and discussion for each individual contribution 197

H.1.1 Multiscale multidimensional parser 197

H.1.2 Least-squares prediction in MMP 198

H.1.3 Lossless image compression using MMP 199

H.1.4 Depth coding using MMP 199

H.1.5 Joint texture and depth coding using MMP 200

H.2 Key issues and open questions 201

H.3 Future perspectives for image coding 201

I Pseudo-codes 203

I.1 Main function of the MMP encoder 203

I.2 Block optimization 205

I.3 Entropy coding 211

I.4 Dictionary update 214

J List of publications 216

K Test set 218

K.1 Smooth images 218

K.2 Compound images 224

K.3 3D images 225

Referências Bibliográficas 229

Lista de Figuras

1.1 Sistema de captura de imagens da sonda espacial Kaguya (Selene) . . 2

2.1 Casamento de padrões multiescala 8
2.2 Segmentação diádica 9
2.3 Modos de predição Intra do H.264/AVC usados no MMP-Intra 10
2.4 Resultados experimentais para as imagens LENA e GOLD 14
2.5 Resultados experimentais para as images PP1205 e PP1209 15

3.1 Vizinhança de treinamento usada para a predição LSP 17
3.2 Implementação do LSP em blocos 19
3.3 Curvas taxa-distorção para imagens suaves 21
3.4 Curvas taxa-distorção para imagens compostas 22

4.1 Alteração dos modos de predição Intra para compressão sem perdas . 25
4.2 Restrição do histograma 26
4.3 Remapeamento do resíduo 27
4.4 Resultados para diversas técnicas de compressão sem perdas 28
4.5 Compressão sem perdas de imagens suaves 29
4.6 Resultados de compressão sem perdas para imagens compostas e computadorizadas 30

5.1 Diversas medidas qualitativas (PSNR e distância de Hausdorff) para mapas de profundidade 34
5.2 Diversas medidas qualitativas (AE e SSIM) para mapas de profundidade 35
5.3 Desempenho da vista reconstruída com mapas de profundidade codificados 37
5.4 Algoritmo *edge-aware* para codificação de mapas de profundidade . . 38
5.5 Desempenho do codificador conjunto multivistas e respectivas profundidades 41

B.1 Self similar portions of the Lena image 52
B.2 Multiscale pattern matching. 53

B.3 Dyadic block segmentation 54
B.4 Dictionary update procedure 55
B.5 General pseudo-code for the MMP algorithm 57
B.6 Prediction segmentation 58
B.7 Prediction modes used in MMP-INTRA algorithm. 59
B.8 Histogram plot for prediction residues of the Lena image 59
B.9 Dictionary redundancy 61
B.10 Possible block sizes for the MMP-FP algorithm. 63
B.11 Flexible block segmentation for the Lena image 64
B.12 Generic block segmentation 65
B.13 Experimental results for images LENA and GOLD 71
B.14 Experimental results for images PP1205 and PP1209 72
B.15 Dictionary growth 73

C.1 Training neighborhood for LSP prediction 77
C.2 Edge directed property 78
C.3 Wiener filter structure 80
C.4 LSP block prediction implementation 81
C.5 LSP block prediction implementation for causal areas 82
C.6 LSP parameter optimization 83
C.7 Results for LSP prediction 84
C.8 Residue probability distribution 84
C.9 Rate distortion curves for smooth images 86
C.10 Rate distortion curves for compound images 87
C.11 Percentages of choice for each prediction mode for image Barbara . . 88
C.12 Percentages of choice for each prediction mode for image Lena 88
C.13 Barbara LSP prediction 90
C.14 Subjective evaluation for Barbara image 91
C.15 Lena LSP prediction 92

D.1 JPEG-LS block diagram 95
D.2 CALIC block diagram 97
D.3 GAP neighborhood 97
D.4 CALIC context modeling and error feedback 99
D.5 TMW block diagram 100
D.6 MRP block classification optimization 103
D.7 Modified prediction mode 117
D.8 Histogram restriction 119
D.9 Residue remapping 121

E.1 Chain of operation for a 3D system 127
E.2 View-geometry representation 128
E.3 Acquisition system for Free-Viewpoint of static scenes 130
E.4 Camera arrangements 131
E.5 Camera normalization 131
E.6 Auto-stereoscopic display 135
E.7 Lenticular and parallax barriers 136
E.8 Advanced 3D video concept 136
E.9 Pinhole camera model 137
E.10 Epipolar geometry 139
E.11 VSRS general mode 140
E.12 VSRS 1D mode 141
E.13 MPEG vision 143
E.14 3D rendering capability 144
E.15 FTV standardization 145

F.1 Rate-distortion performance for coded disparity maps with ground truth 148
F.2 Rate-distortion performance for coded depth maps with non-aligned camera arrangement 149
F.3 Rate-distortion performance for coded depth maps with horizontal alignment 150
F.4 Percentage of different pixels between the coded disparity maps and the ground truth 152
F.5 Hausdorff distance of coded depth maps 153
F.6 SSIM of coded depth maps with horizontal camera alignment 154
F.7 Rate-distortion performance for reconstructed views using uncoded texture and coded depth maps 156
F.8 Subjective analysis of reconstructed views using coded depth data . . 157
F.9 Reconstruction subjective comparison using two different DIBR algorithms 159
F.10 Reconstruction objective comparison using two different DIBR algorithms 160
F.11 Comparison of coded depth maps used in the reconstruction with coded texture for the Microsoft sequences 161
F.12 Comparison of coded depth maps used in the reconstruction with coded texture for the MPEG sequences 162
F.13 Subjective comparison of coded depth maps used in the reconstruction with coded texture for Book Arrival sequence 163

F.14 Edge from depth maps detected by the proposed algorithm 165
F.15 Optimization of the threshold criteria for edge aware coding of the Microsoft sequences 167
F.16 Optimization of the threshold criteria for edge aware coding of the MPEG sequences 168
F.17 Comparison of reconstructed frames using the edge aware coding procedure 169
F.18 Rate distortion performance of edge aware MMP for the Microsoft sequences 170
F.19 Rate distortion performance of edge aware MMP for the MPEG sequences 171

G.1 Rate-distortion curves for the reference view of the first frame of Microsoft sequences 176
G.2 Rate-distortion curves for the reference view of the first frame of MPEG sequences 177
G.3 Rate-distortion performance of auxiliary views for the Microsoft sequences 178
G.4 Rate-distortion performance of auxiliary views for the MPEG sequences179
G.5 Bitrate allocation optimization for the Microsoft sequences 181
G.6 Bitrate allocation optimization for the MPEG sequences 182
G.7 Rate-distortion improvement of the auxiliary view for the Microsoft sequences 185
G.8 Rate-distortion improvement of the auxiliary view for the MPEG sequences 186
G.9 Quality of the prediction used for encoding the auxiliary view of the Microsoft sequences 187
G.10 Quality of the prediction used for encoding the auxiliary view of the MPEG sequences 188
G.11 Prediction usage maps for the Breakdancers sequence 189
G.12 Prediction usage maps for the Book Arrival sequence 190
G.13 Final quality of the reconstruced view from sequences provided by Microsoft 192
G.14 Final quality of the reconstruced view from sequences provided by MPEG 193
G.15 Subjective comparison for the first frame of the Ballet sequence . . . 194
G.16 Subjective comparison for the first frame of the Book Arrival Sequence195

I.1 Main function diagram of the MMP encoder. 204
I.2 Pseudo-code notation 204

I.3 Optimization function for prediction 207
I.4 Optimization diagram for the hierarchical prediction 210
I.5 Optimization diagram for the analysis stage 213
I.6 Optimization diagram for the dictionary update 215

K.1 Airplane 218
K.2 Baboon 219
K.3 Balloon 219
K.4 Barb 220
K.5 Barb2 220
K.6 Cameraman 221
K.7 Couple 221
K.8 Goldhill 221
K.9 Lena 222
K.10 Lennagrey 222
K.11 Noisesquare 222
K.12 Peppers 223
K.13 Shapes 223
K.14 PP1205 224
K.15 PP1209 224
K.16 SCAN0002 224
K.17 SCAN0004 224
K.18 SCAN0006 224
K.19 Tsukuba 225
K.20 Teddy 225
K.21 First frames of the Ballet multiview sequence 226
K.22 First frames of the Breakdancers multiview sequence 227
K.23 First frames of selected views of the Book Arrival sequence 228
K.24 First frames of selected views of the Champagne Tower sequence . . . 228

Lista de Tabelas

2.1 Percentual de tempo economizado com o novo algoritmo MMP-FAST 12

B.1 Computational complexity for each block 67

B.2 Call of the search function for each block 68

B.3 Percentage of time saved with the new FAST algorithm 70

B.4 Simulation details for CAMERAMAN 73

D.1 State-of-the-art lossless compression algorithms results for smooth and compound images 107

D.2 MMP's lossless compression performance 113

D.3 Lossless compression results for several different prediction proposals 117

D.4 Lossless compression results for several different residue encoding techniques 122

D.5 Results for smooth image lossless compression 123

D.6 Results for compound and artificial image lossless compression 123

F.1 Experimental set-up for multiview images 164

G.1 Experimental set-up for multiview images joint coding 191

Lista de Símbolos

3DAV	3D Audio/Video, p. 3
AVC	Advanced Video Coding, p. 4
DCT	Discrete Cosine Transform, p. 3
DIBR	Depth Image Based Rendering, p. 137
EDP	Edge-Directed Prediction, p. 103
FIR	Finite Impulse Response, p. 80
GAP	Gradient-Adjusted Prediction, p. 99
GPU	Graphic Processing Unit, p. 74
HDTV	High Definition Television, p. 2
HEVC	High Efficiency Video Coding, p. 3
JBIG	Joint Bi-level Image Experts Group, p. 3
JPEG	Joint Photographic Expert Group, p. 3
LOCO-I	LOw COmplexity LOssless COmpression for Images, p. 96
LSP	Least-Square Prediction, p. 5
MED	Median Edge Detector, p. 96
MMP	Multidimensional Multiscale Parser, p. 3
MPEG	Motion Picture Expert Group, p. 3
MRF	Markov Random Fields, p. 76
MRP	Minimum-Rate Predictors, p. 103
MSE	Mean Square Error, p. 78

MVC	Multiview Video Coding, p. 3
NSHP	Non-Symmetric Half Plane, p. 76
RLS	Recursive Least Square, p. 90
SSIM	Structural SIMilarity, p. 152
SVD	Single Value Decomposition, p. 78
TSGD	Two-Sided Geometric Distribution, p. 97

Capítulo 1

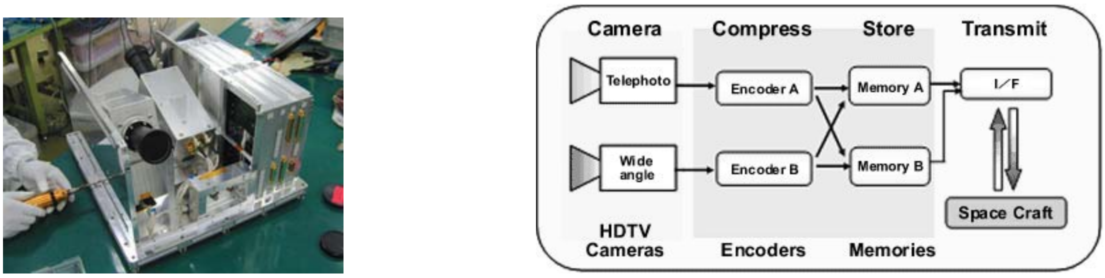
Introdução

1.1 História da codificação

A utilização de imagens permitiu ao homem expandir a sua visão. Com a chegada da era digital, difundiu-se o uso das imagens, e estas ficaram ao alcance de todas as pessoas, de tal modo que hoje podemos ver o funcionamento de órgãos dentro do corpo humano, podemos passear na lua e ver galáxias distantes. E estas imagens carregam consigo uma enorme quantidade de informação. Furacões podem ser previstos, doenças detectadas, novas reservas de metais podem ser achadas em planetas distantes. Porém esta revolução não só atingiu a comunidade científica. Hoje em dia, a maioria das pessoas carrega uma câmera digital, assiste a vídeos pela Internet, envia documentos via fax e imprime o seu conteúdo em impressoras. Por detrás de atividades corriqueiras e atividades extraordinárias, a compressão de imagens está presente.

No início, a compressão de imagens foi o fator viabilizador da digitalização das imagens. A capacidade de armazenamento era reduzida e o custo de transmissão era alto. Por exemplo, para digitalizarmos um segundo de vídeo sem compressão (usando o formato ITU-R 601), precisamos de mais de 20 MB, o que resulta em diversos gigabytes para apenas um único filme. Outro exemplo é a transmissão de imagens pela Internet, que sem compressão poderia levar horas, e hoje ocorre em poucos segundos.

Os primeiros algoritmos de compressão tiveram a dura tarefa de reduzir a quantidade de dados necessários para representar um conteúdo digital, para que este pudesse ser armazenado/transmitido. Hoje em dia temos muito mais espaço para armazenamento, os canais de transmissão ficaram mais robustos e com maior capacidade, os custos foram reduzidos e as máquinas ficaram mais potentes. Ao mesmo tempo ficamos mais críticos, queremos imagens mais nítidas, de maior definição, até mesmo imagens em três dimensões (como por exemplo, as imagens do solo lu-



(a) Câmera HDTV usada na sonda espacial Kaguya

(b) Esquema de captura e codificação de imagens usado na sonda espacial Kaguya

Figura 1.1: Exemplo de aplicação de compressão usado no sistema de captura de imagens 3D de alta definição da sonda espacial Kaguya (Selene). Duas câmeras capturaram imagens em alta definição do solo lunar para formar imagens em 3D. Devido à enorme quantidade de dados captados e limitado espaço de armazenamento, o algoritmo de compressão é essencial para viabilizar o mapeamento do solo lunar em alta definição, com resolução de até 50 metros.

nar obtidas através da sonda espacial japonesa Kaguya, Fig. 1.1). Queremos ver imagens a qualquer hora e em qualquer dispositivo, desde telas gigantes com ultra definição, até as pequenas telas dos celulares enquanto vamos de um lado a outro.

Desta forma, os algoritmos de compressão sempre serão necessários, pois mesmo com mais espaço e sendo mais barata a transmissão, a demanda aumentará por sinais mais complexos, com mais informação. Teremos a demanda também por imagens em diferentes tipos de terminais mais simples, como as telas de celulares, que apresentam uma capacidade de processamento bem mais limitada. Os algoritmos de compressão funcionam como um compromisso entre a capacidade de transmissão, armazenamento e a complexidade computacional para se obter a imagem. Algoritmos mais complexos computacionalmente irão produzir imagens mais compactas, algoritmos mais simples irão comprimir menos as imagens, porém poderão ser mais rápidos e computacionalmente mais eficientes.

O objetivo dos algoritmos de compressão é achar uma representação eficiente para a fonte a ser comprimida. No caso dos codificadores sem perdas, temos um limite teórico do quanto podemos comprimir uma imagem. Pela teoria da informação desenvolvida por Shannon [1], esse limite é a entropia da fonte. Porém a chave dos algoritmos de compressão sem perdas é saber como chegar nesse valor de forma eficiente. Muitos dos algoritmos estado-da-arte usam técnicas avançadas de predição, com a finalidade de reduzir a informação redundante presente na imagem, seguida de um codificador entrópico.

Já nos algoritmos de compressão com perdas, os algoritmos de compressão de maior sucesso são híbridos e baseados em três passos: transformada, quantização e codificação por entropia [2]. Os passos de transformada e codificação são reversíveis, porém o passo de quantização não o é, uma vez que ele introduz uma perda. A quantização irá mapear um intervalo possível de valores em um único valor, por isso é que temos uma incerteza associada ao valor real e a introdução de uma perda. Já o passo de codificação por entropia é geralmente o passo computacionalmente mais intenso. Ele é usualmente realizado de duas formas: baseado em codificadores aritméticos (computacionalmente mais complexos, porém geram um resultado mais eficiente, para aplicação com alto desempenho) ou em codificadores Huffman (mais rápidos, geralmente usam tabelas alocadas em memória, para aplicações mais simples).

Para incentivar o desenvolvimento de *softwares* e aplicativos baseados em compressão de imagens, padrões foram concebidos. Os padrões geralmente descrevem como deve funcionar o decodificador, para que diversos produtos diferentes possam comunicar entre si, usando uma linguagem comum. O primeiro padrão internacionalmente adotado para compressão de images foi o JPEG [3], sendo utilizado largamente na internet ou até mesmo em aplicações como impressoras e câmeras digitais. O objetivo do JPEG é a codificação de imagens em tom contínuo com mais de um bit de profundidade. Para imagens com apenas dois níveis (preto e branco), usadas na transmissão de fax, JBIG-1 [4] e JBIG-2 [5] são os padrões mais usados. Já o padrão JPEG2000 [6] passou a utilizar wavelets no lugar da transformada DCT, usada anteriormente pelo JPEG. No caso do vídeo, o padrão MPEG [7], também baseado em transformadas e quantização, é um dos padrões mais utilizados no mundo. Além disso, estamos assistindo nos dias de hoje à criação de novos padrões, como o padrão de codificação avançada de vídeo (H.264/AVC [8], H.264/MVC [9], H.264/HEVC [10]) ou o padrão para vídeos 3D (3DAV [11]).

A grande vantagem da compressão baseada em transformadas é a baixa complexidade computacional, um item quase essencial quando queremos transmissão em tempo real. Porém com o passar dos anos temos disponíveis máquinas cada vez mais potentes, o que nos leva a reavaliar este paradigma, e investir em novas abordagens, que consigam alcançar patamares de compressão até o momento não alcançados. Algoritmos como compressão fractal de imagens [12] já vêm obtendo algum relativo sucesso, provando que ainda existe muito o que se explorar quando falamos de compressão de imagens.

1.2 Motivações

Nos últimos anos, o codificador baseado em casamento de padrões recorrentes em multiescala, o MMP (*Multidimensional Multiscale Parser*), tem sido o principal assunto de muitas linhas de pesquisa [13–16]. O seu caráter universal abriu portas para que ele fosse utilizado em diversas áreas, iniciando com compressão de imagens compostas [17] (onde ele apresenta um desempenho estado-da-arte), e passando por áreas como compressão de imagens suaves [18], imagens estereoscópicas [19] e até eletrocardiogramas [20].

No que diz respeito a codificação de imagens suaves, o MMP surge como alternativa a um paradigma já bem estabelecido, a compressão usando transformada-quantização-codificação. A maneira inovadora com que o MMP atua, mesclando essas três operações em uma, abre um leque enorme de possibilidades a serem exploradas. Algumas das propriedades do MMP para codificar imagens suaves já vêm sendo exploradas há algum tempo ([21, 22]), e fizeram com que o MMP recentemente superasse os ganhos de compressão de codificadores estado-da-arte, como o JPEG2000 [6] e o H.264/AVC *Intra* [23].

A mudança de paradigma introduzida pelo MMP pode levar-nos a novos patamares de compressão, o que motiva a continuação do seu uso em diferentes áreas, além do aprimoramento da sua utilização nas áreas em que já atua. A proposta desta tese é ter o MMP como o algoritmo base de compressão de imagens e daí explorar técnicas novas, propondo novas soluções baseadas no codificador, que possam trazer uma nova perspectiva ao uso do MMP. Um exemplo claro de áreas onde o MMP ainda não apresentou nenhum resultado é a área de compressão sem perdas. O MMP tem uma capacidade inata de compressão sem perdas, porém o seu desempenho nesta área ainda não foi avaliado. Outra proposta interessante é o uso do MMP para a codificação do novo formato de imagens 3D, que incluem múltiplas vistas e seus respectivos mapas de profundidades.

Um grande desafio deste trabalho também será diminuir o custo computacional do algoritmo, que vem crescendo bastante com a introdução de novas técnicas adaptativas de codificação. Deficiência em áreas como a complexidade do algoritmo terão que ser combatidas, sem comprometer o desempenho taxa-distorção que o MMP tem registrado nos últimos anos. Um dos objetivos das técnicas desenvolvidas nesta tese deverá ser também o de manter a complexidade do MMP a níveis aceitáveis, e sempre que possível reduzir o tempo de codificação, para viabilizar o uso do MMP nas máquinas de hoje.

1.3 Organização da tese

Esta tese está organizada da seguinte forma. Os Capítulos 1 a 6 foram escritos em português, e têm por objetivo fornecer uma visão geral do trabalho realizado no âmbito desta tese. Os apêndices, por sua vez, foram escritos em inglês, e correspondem a uma versão mais detalhada do que será apresentado nos capítulos da tese.

No Capítulo 2, será apresentada uma revisão sobre o codificador MMP e suas evoluções. A complexidade computacional do algoritmo é formalmente analisada, e uma proposta de redução do custo computacional é feita. Simulações e resultados experimentais encontram-se no final deste capítulo. Maiores detalhes sobre os algoritmos baseados no paradigma do MMP e detalhes sobre as contribuições para a redução da complexidade computacional se encontram no Apêndice B.

O Capítulo 3 mostra os passos desenvolvidos para incoporar um método de predição adaptativo baseado no critério dos mínimos quadrados (LSP - *Least-square Prediction*) ao MMP. Adaptações para codificações em bloco e resultados experimentais são apresentados. Detalhes do método proposto e uma análise mais abrangente dos resultados encontram-se no Apêndice C.

Como o LSP foi uma técnica inicialmente idealizada para compressão sem perdas, e até o momento o MMP não foi avaliado neste sentido, no Capítulo 4, os conceitos usados na compressão sem perdas serão visitados, e um estudo do MMP atuando como um codificador sem perdas será realizado. Desta forma, poderemos ver o MMP atuando em todas as taxas de compressão, indo da compressão com perdas até a compressão sem perdas. Uma análise formal dos limites teóricos do MMP para compressão sem perdas e uma análise mais aprofundada do tema se encontram no Apêndice D.

O novo formato de imagens 3D é baseado na transmissão de textura e um correspondente mapa de profundidade. Este mapa apresenta características muito específicas [24], o que representa um novo desafio para codificadores de imagens. O Capítulo 5 explora a atuação do MMP na codificação de mapas de profundidade, e vai além, ao usarmos o MMP para codificar todos os elementos de uma imagem 3D. Iremos propor arquiteturas de codificadores 3D, e apresentar propostas para melhorias na codificação dos mapas de profundidade. Os Apêndices E, F e G apresentam os desafios deste novo formato de imagens 3D, mostram com mais detalhamento como foram abordados os problemas de codificação e quais as soluções propostas.

Por fim, o Capítulo 6 irá concluir esta tese e apresentar tópicos para a continuação deste trabalho de pesquisa.

Os Apêndices I, J e K complementam o trabalho com o pseudo-código dos algoritmos usados, a lista de publicações geradas por este trabalho e as imagens utilizadas nos testes, respectivamente.

1.4 Contribuições da tese

1.4.1 Contribuição para a codificação de imagens com perdas

Focando na codificação de imagens com perdas baseado no codificador MMP com predição, um método de aceleração do algoritmo foi proposto e publicado em [25]. Em seguida, uma adaptação não-trivial do método de predição baseado no critério dos mínimos quadrados foi realizada. O novo método de predição adaptativo foi adicionado ao modos de predição disponíveis, e dessa forma o desempenho taxa-distorção do MMP foi melhorado. O método foi publicado em [26] e em [27].

1.4.2 Contribuição para codificação de imagens sem perdas

A atuação do MMP como um codificador sem perdas foi explorada. Pela primeira vez, os limites teóricos do MMP como um codificador sem perdas foram formalmente desenvolvidos e o seu desempenho foi avaliado. Os métodos de predição do algoritmo foram adaptados para a codificação sem perdas e modificações no codificador entrópico para este caso foram avaliadas. Os resultados desta pesquisa foram submetidos para publicação em [28].

1.4.3 Contribuição para a codificação de imagens 3D

A flexibilidade do MMP mostrou-se particularmente eficaz na codificação do novo formato de imagens 3D, que usam textura em conjunto com mapas de profundidade/disparidade. Mapas de profundidade codificados com o MMP foram avaliados, onde critérios objetivos e subjetivos foram implementados e utilizados na avaliação dos artefatos. Em seguida, diversas arquiteturas baseadas no MMP para codificar imagens 3D foram propostas, onde uma alocação ótima de taxa entre a textura e os mapas de disparidade foi obtida. Neste âmbito, o desempenho das arquiteturas propostas com o MMP foi comparado com outros algoritmos usados para a codificação 3D. Além disso, mudanças no codificador foram propostas, para evitar artefatos junto às bordas dos objetos representados nos mapas de profundidade. Os resultados do MMP como codificador de mapas de profundidade foram publicados em [29, 30] e a proposta para um novo codificador de images 3D baseado em MMP está em preparação para submissão.

Capítulo 2

Codificador de imagens com casamento de padrões multiescala

2.1 Casamento aproximado de padrões usando multiescalas

Os algoritmos derivados do paradigma do MMP têm como base o casamento de padrões recorrentes. Esta técnica é apropriada para codificar elementos que se repetem frequentemente, usando um dicionário que contém uma lista dos elementos mais usados, $\mathcal{D} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_M\}$. Ao ser identificada a ocorrência de um padrão do dicionário na imagem a ser codificada, o seu índice é codificado no lugar do padrão usando um código de tamanho variável. Atingimos compressão quando os códigos mais curtos são usados para codificar padrões mais frequentes. Caso o padrão não exista no dicionário, ele será codificado de uma outra forma, menos eficiente, geralmente usando códigos de escape. Um exemplo de codificação sem perdas usando dicionário são os algoritmos Lempel-Ziv [31–41]. No caso de codificação com perdas usando o mesmo paradigma, os algoritmos são conhecidos como *Lossy* Lempel-Ziv [17, 42–44]. O MMP propõe usar blocos de outras dimensões para fazer o casamento, comparando blocos com diferentes escalas, uma idéia explorada também em codificadores fractais [12]. A vantagem de usarmos o casamento em múltiplas escalas deve-se ao fato de obtermos mais elementos para o dicionário através de contrações e expansões dos elementos atuais, como é mostrado na Figura 2.1.

O algoritmo MMP aplica os conceitos de casamento de padrões em multiescala da seguinte forma. A imagem de entrada será dividida em blocos que não se sobrepoem, e padrões de um dicionário em diferentes escalas serão usados para aproximar os blocos da imagem. Se nenhum padrão presente no dicionário aproxima o bloco da imagem de forma satisfatória, este é então iterativamente dividido, primeiro na vertical, em seguida na horizontal, até atingirmos uma escala onde temos um casa-

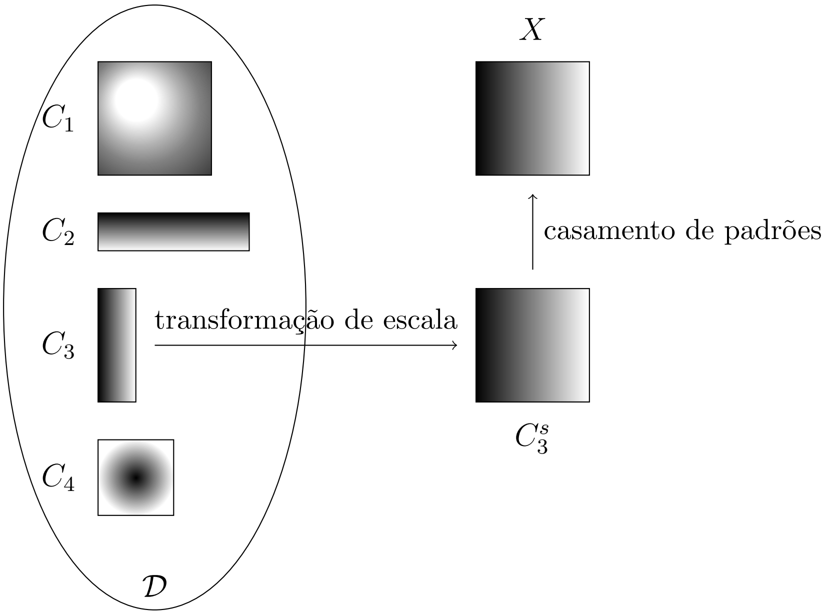


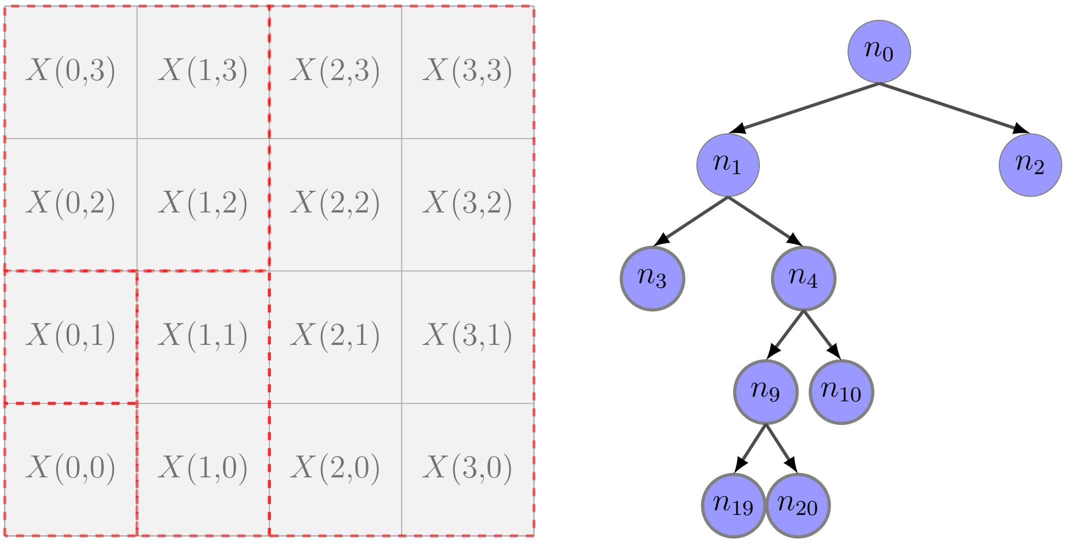
Figura 2.1: Exemplo de um casamento de padrões multiescala. A transformação de escala permite o casamento de elementos de escala diferente, como vemos neste exemplo, onde o elemento C_3 do dicionário $\mathcal{D}$ sofre uma mudança de escala para codificar o padrão de entrada X.

mento apropriado entre o bloco e um padrão presente no dicionário. A segmentação do bloco é sinalizada para o decodificador através de uma flag de segmentação. Este por sua vez usa as flags para repetir o processo de segmentação do codificador e atualizar os blocos de images com os índices transmitidos do dicionários. A Figura 2.2 mostra o processo de segmentação diádica, e a sua respectiva árvore de segmentação. Um codificador aritmético adaptativo é usado como codificador entrópico, para formar o fluxo binário da imagem codificada com o MMP.

Após a codificação do bloco, tanto o codificador quanto o decodificador irão atualizar o dicionário com o novo padrão formado pelas concatenações dos diversos subblocos, sendo que a atualização irá ocorrer em diversas escalas, através de expansões e contrações do novo bloco. Desta maneira, o dicionário cresce com valores da própria imagem, que provavelmente serão usados para codificar os blocos seguintes, devido às características de auto-similaridade das imagens naturais.

A segmentação do bloco e os índices escolhidos são determinados através de um algoritmo baseado num custo lagrangeano. A melhor segmentação do bloco é escolhida em um passo de otimização, que compara o custo da codificação de cada segmento do bloco e escolhe a segmentação que resulta no custo mínimo. Associamos a cada nó uma distorção

$$D(X, \mathbf{C}_i) = \sum_{x,y} (X(x,y) - \mathbf{C}_i(x,y))^2 \tag{2.1}$$



(a) Segmentação do bloco

(b) Árvore de segmentação

Figura 2.2: Exemplo de um bloco partido usando uma segmentação diádica, e sua correspondente árvore de segmentação

dada pela soma das diferenças ao quadrado entre o bloco original $\mathbf{X}$ e o elemento no dicionário $\mathbf{C}_i$, e a taxa

$$R(\mathbf{C}_i) = -\log_2(\Pr(i)) \tag{2.2}$$

dependente da probabilidade $\Pr(i)$ do símbolo i. O custo do nó n_i será

$$J(n_i) = D(X, \mathbf{C}_i) + \lambda R(\mathbf{C}_i) \tag{2.3}$$

onde λ é o parâmetro do custo lagrangeano.

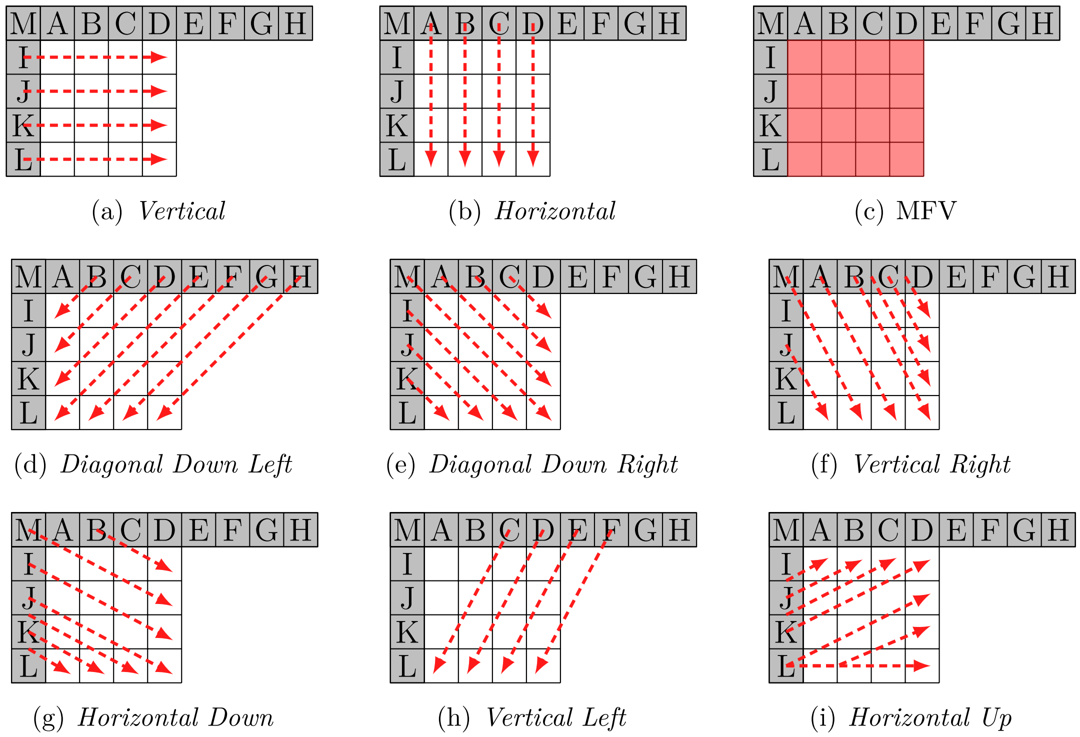
A decisão de segmentação do bloco é tomada com base na soma dos custos de cada nó, isto é, o bloco será dividido se

$$J(n_0) > J(n_1) + J(n_2) + \lambda R_{seg} \tag{2.4}$$

onde R_{seg} é a taxa necessára para transmitir a flag de segmentação.

Os resultados iniciais do MMP foram apresentados em [17], onde o excelente desempenho para imagens compostas se destaca. O MMP já foi aplicado com sucesso na codificação de imagens estéreo [19], eletrocardiogramas [20, 45, 46] e vídeo [47–50]. No entanto, o seu desempenho para imagens suaves ainda ficava bem abaixo dos codificadores estado-da-arte, como o JPEG2000 [6] ou até mesmo o H.264/AVC Intra [51, 52].

Algumas propostas já apresentadas visaram aumentar a eficiência do desempenho taxa-distorção do MMP para imagens suaves, através de um critério mais rigoroso de codificação entre blocos vizinhos, entre elas técnicas como o MMP-APM (*Adaptive*



(a) *Vertical* (b) *Horizontal* (c) MFV

(d) *Diagonal Down Left* (e) *Diagonal Down Right* (f) *Vertical Right*

(g) *Horizontal Down* (h) *Vertical Left* (i) *Horizontal Up*

Figura 2.3: Modos de predição Intra do H.264/AVC usados no MMP-Intra. Apenas o modo de predição DC foi substituído pelo modo de predição que usa o valor mais frequente (MFV)

Probability Model, [53]). Porém a técnica que apresentou os melhores resultados foi a desenvolvida em [54], que usa o conceito de predição, conhecido como MMP-Intra.

O MMP-Intra [15, 21, 54] usa os mesmos oito modos direcionais de predição usados no H.264/AVC, como mostra a Figura 2.3. Porém no caso do modo DC, um novo modo mais apropriado para codificar imagens suaves e compostas, o MFV (*Most Frequent Value*), é usado. A diferença entre o modo DC e o MFV é que no primeiro o valor predito é a média dos valores da fronteira, enquanto que o outro usa o valor que ocorreu mais vezes [21, 54]. Assim como no H.264/AVC, a predição pode usar blocos de tamanhos variados, limitados até um tamanho de 4 × 4. A grande vantagem de usarmos a predição é o fato do resíduo apresentar uma conjunto muito menor de valores diferentes, o que facilita o processo de adaptação do dicionário. No mesmo âmbito do MMP-Intra, métodos de adaptação do dicionário foram propostos e incoporados no algoritmo. Os detalhes encontram-se em [21].

Uma das grandes vantagens de algoritmos de compressão baseados no MMP é a flexibilidade que ele apresenta para codificar diversos tipos de imagens. No entanto, o MMP ainda apresenta alguns aspectos rígidos no seu algoritmo, como a segmentação dos blocos e os modos de predição. Em [22], a segmentação diádica do bloco foi substituída por uma segmentação flexível. Através do uso de *flags* diferenciadas, o codificador MMP irá indicar a direção de segmentação do bloco.

Com os últimos avanços no codificador MMP, o seu desempenho taxa-distorção passou a atingir patamares estado-da-arte para diversos tipos de imagens. No entanto, o ganho em taxa-distorção surgiu como contrapartida de um aumento significativo na complexidade computacional. Mais detalhes sobre as diversas técnicas aplicadas ao MMP se encontram no Apêndice B. Nas próximas seções iremos apresentar uma proposta que visa reduzir a complexidade computacional e diminuir o tempo de codificação do MMP.

2.2 Reduzindo a complexidade computacional

Embora na maioria dos casos o aumento de complexidade proveniente da predição é devido ao aumento de operações para o próprio cálculo da predição, no caso do MMP essa rotina não foi a principal causa do aumento da complexidade computacional. O método de otimização que irá escolher o melhor modo de predição é o responsável pelo aumento do tempo de codificação, uma vez que é necessário codificar o resíduo com o MMP para determinar o custo de usarmos uma determinada predição, e isso implica numa nova busca por um padrão do dicionário. A complexidade computacional do algoritmo original do MMP foi apresentada em [45], e no Apêndice B encontra-se o mesmo desenvolvimento para o algoritmo com predição e segmentação flexível.

Em métodos tradicionais de codificação de imagens baseados em transformadas, uma técnica comum para a redução do custo computacional é a escolha seletiva de apenas alguns modos de predição, baseado em algum critério, como por exemplo o método do gradiente proposto em [55]; porém no âmbito do MMP isso não iria reduzir drasticamente a complexidade, uma vez que ainda teríamos de efetuar as diversas buscas associadas ao cálculo do custo de codificação dos resíduos dos modos de predição remanescentes.

Uma maneira eficaz de evitar o cálculo do custo de codificação do resíduo usando o MMP é modificar o algoritmo de otimização, usando um critério diferente para a seleção do melhor modo de predição. Nesta implementação rápida, o modo de predição escolhido é aquele que gera o resíduo com uma energia mais baixa, ou seja, iremos apenas levar em consideração a distorção do bloco após a predição, e não mais a taxa associada com a codificação do resíduo via MMP. O algoritmo que incorpora a predição, a segmentação flexível e a escolha rápida do modo de predição passou a se chamar MMP-FAST.

Tabela 2.1: Percentual de tempo economizado com o novo algoritmo MMP-FAST.

Rate	0,4 bpp	0,7 bpp	1,1 bpp
GOLD	87%	88%	85%
LENA	86%	85%	81%
PP1205	87%	81%	84%
PP1209	86%	86%	82%

2.2.1 Resultados experimentais do MMP-FAST

As figuras 2.4(a) e 2.4(b) mostram o resultado do MMP-FAST para duas imagens suaves. Nota-se uma queda contante de 0,2 dB para o PSNR em todas as taxas. No entanto o desempenho taxa-distorção ainda está acima do H.264/AVC Intra para médias e altas taxas, e o ganho computacional deixou o codificador 7 vezes mais rápido.

Nas imagens compostas, a premissa de suavidade não se aplica, e as perdas devido ao modo de predição escolhido foram ainda maiores, chegando a 1 dB para a imagem PP1205 que contém somente texto, e 0,4 dB para a imagem composta PP1209. Mesmo com esta grande perda em termos de qualidade objetiva, ainda estamos acima dos codificadores usados como base de comparação (veja figuras 2.5(a) e 2.5(b)). Equivalente ao que aconteceu com as imagens suaves, também para imagens compostas o código ficou sete vezes mais rápido do que a versão anterior.

A Tabela 2.1 mostra os ganhos em tempo de codificação com o uso da nova técnica de decisão.

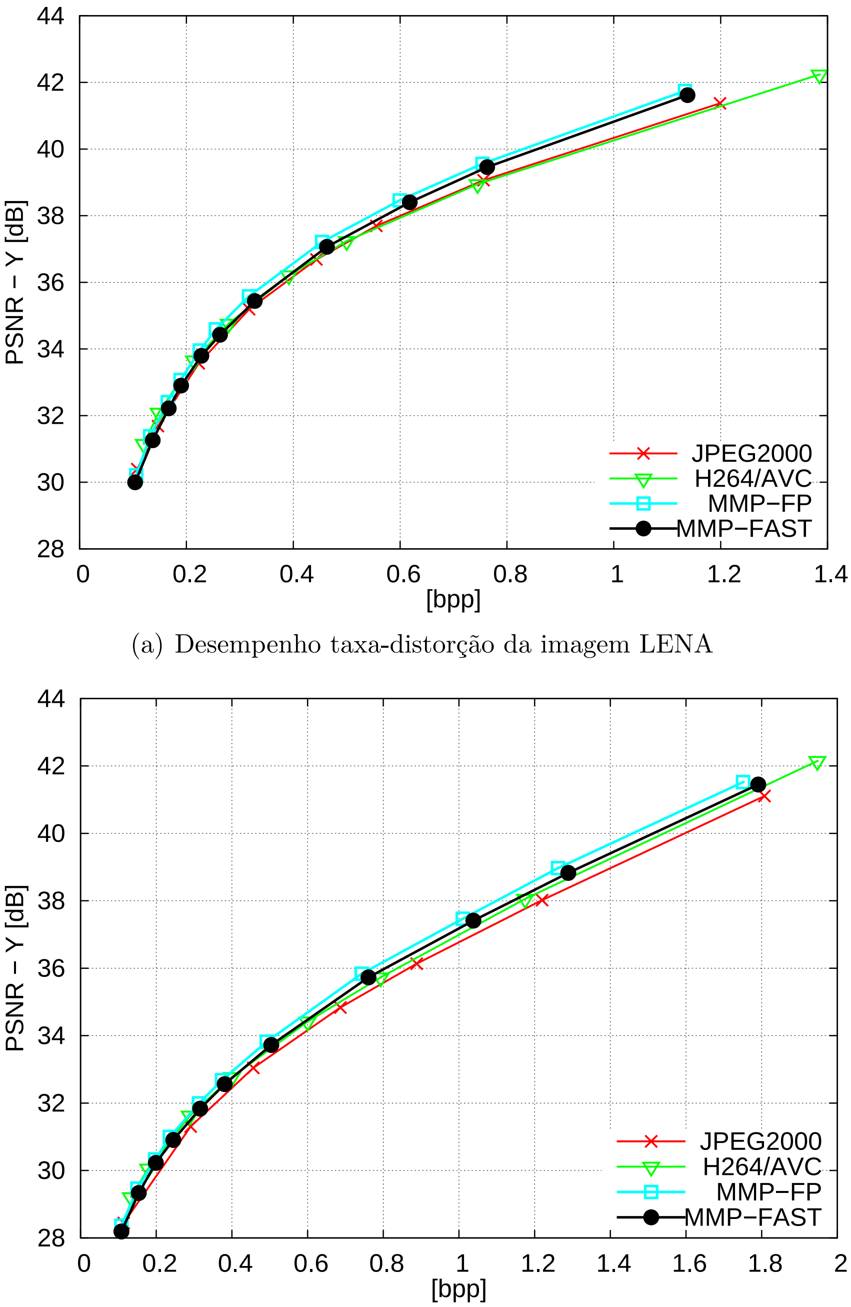
2.3 Conclusões

A grande adaptabilidade do MMP foi o fator que permitiu ao codificador atingir desempenho taxa-distorção ao nível do estado-da-arte. A flexibilidade e o largo campo de atuação que o MMP atinge torna-o uma excelente ferramenta de codificação. No entanto, o aumento de flexibilidade acarretou em um aumento de complexidade computacional, e a técnica de redução de complexidade aqui apresentada é apenas uma ferramenta que atua no sentido de diminuir o tempo de codificação. Novas técnicas para redução da complexidade computacional é um tópico de pesquisa importante para o MMP.

Outro tópico interessante a ser explorado é a paralelização do processo de codificação com o MMP. Todas as buscas em múltiplas escalas podem ser eficientemente paralelizadas, aumentando consideravelmente o desempenho computacional do codificador. Essa linha de evolução está de acordo com a linha de evolução dos novos processadores, e acompanha a tendência de desenvolvimento de *hardware* para aplicações de codificação de imagens, como por exemplo as aplicações utilizando GPU's (*Graphic Processing Unit*).

Devido a sua flexibilidade, o MMP pode atuar na compressão de qualquer tipo de imagem, sejam imagens naturais, ou ainda imagens artificiais, como por exemplo imagens geradas por computador ou mapas de profundidades, usados em imagens 3D. O uso do MMP para codificar imagens pode ser uma alternativa para codificadores atuais, baseados em transformadas.

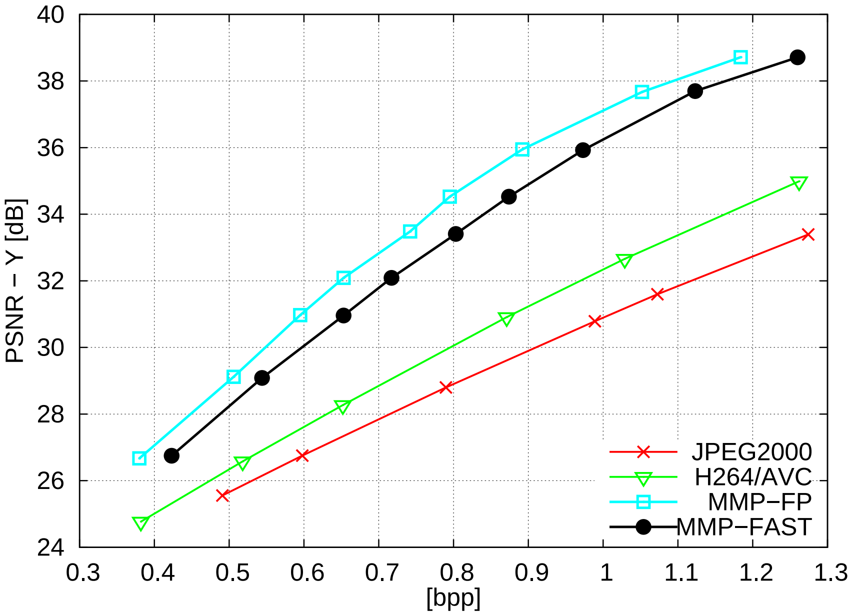
Como podemos ver, existem diversos tópicos interessantes a serem explorados no âmbito do codificador MMP. Os próximos capítulos desta tese irão abordar algumas vertentes aqui discutidas. Por ser uma abordagem nova, e uma mudança grande do paradigma de codificação de imagens, o MMP permite-nos explorar caminhos que conduzam a novos patamares na codificação de imagens.



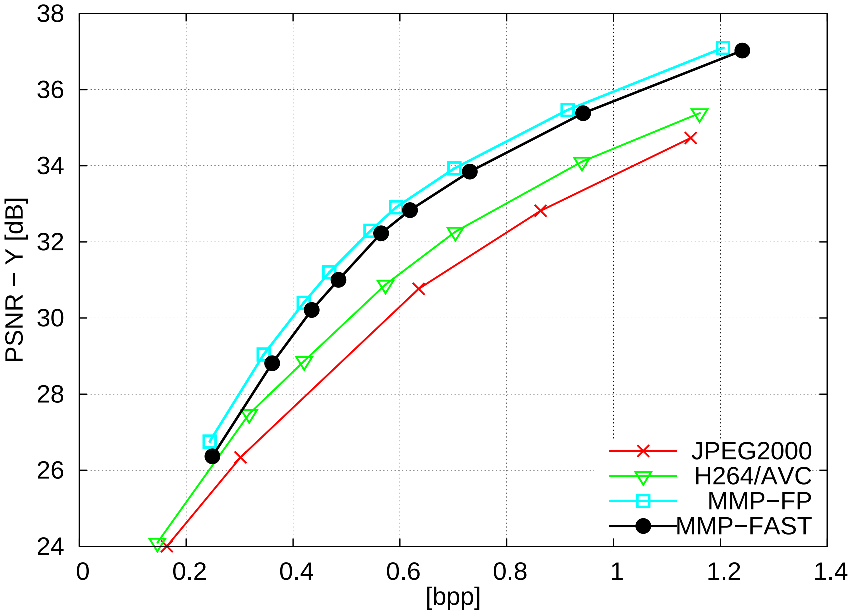
(a) Desempenho taxa-distorção da imagem LENA

(b) Desempenho taxa-distorção da imagem GOLD

Figura 2.4: Resultados experimentais de taxa-distorção do MMP-FAST para as imagens LENA e GOLD.



(a) Desempenho taxa-distorção da imagem PP1205



(b) Desempenho taxa-distorção da imagem PP1209

Figura 2.5: Resultados experimentais de taxa-distorção do MMP-FAST para as images PP1205 e PP1209.

Capítulo 3

MMP utilizando predição com critério de mínimos quadrados

3.1 Predição usando o critério dos mínimos quadrados

O processo de predição é análogo ao processo de inferência estatística. Este consiste em determinar algum parâmetro de uma distribuição baseado no conjunto de realizações disponíveis, por exemplo, determinar a média ou a variância de alguma distribuição, baseado em um conjunto de dados disponíveis. A validade deste método é baseada no fato que sendo observada uma certa quantidade de dados, a distribuição de interesse se encontrará provavelmente num subconjunto de distribuições descritas por esses dados. Ou seja, as observações podem revelar informações sobre a verdadeira distribuição na qual estamos interessados. Fazendo uma analogia com o processamento de imagens, se tomarmos um conjunto de realizações como sendo a vizinhança do pixel que estamos querendo prever, o processo de predição poderá usar os valores dos pixels vizinhos para estimar o valor do pixel atual.

Iremos então assumir que a lei de formação dos pixels vizinhos é a mesma lei que se aplica ao pixel de interesse. Dessa forma, podemos prever o valor de um pixel $X(\mathrm{g}(n))$ baseado na lei de formação obtida com uma vizinhança local M, também conhecida como "janela de treinamento". A função $\mathrm{g}(n)$ indica a coordenada do pixel na imagem, indexado por uma variável n. De acordo com LI [56], uma escolha adequada da janela de treinamento é uma área contendo dois retângulos de dimensões T e $T+1$, totalizando $M = 2T(T+1)$ elementos (veja a Figura 3.1).

Uma forma de estimarmos o valor do pixel é fazendo uma média ponderada de valores próximos. Geralmente as posições escolhidas são as N posições mais próximas, baseadas no modelo markoviano de ordem N. A predição é obtida então

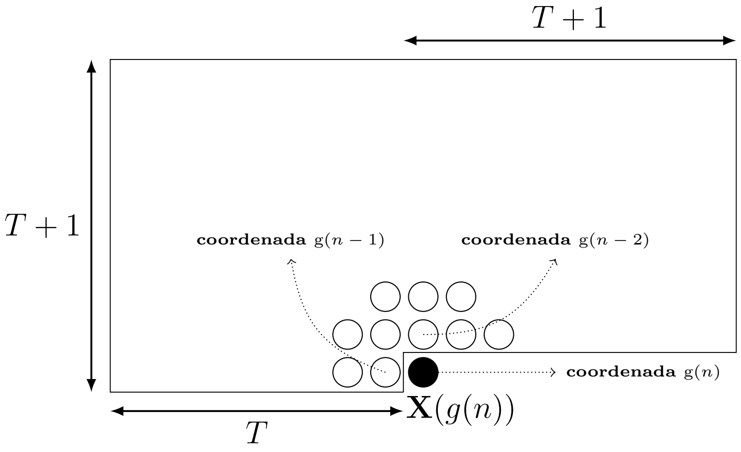


Figura 3.1: Vizinhança e janela de treinamento usado para a predição LSP. Os pixels usados para a predição situam-se numa região causal da imagem. Portanto, tanto o codificador quanto o decodificador realizam o mesmo treinamento e obtêm os mesmos coeficientes.

com a seguinte equação:

$$\hat{\mathbf{X}}(\mathrm{g}(n)) = \sum_{i=1}^{N} a_i \mathbf{X}(\mathrm{g}(n-i)) \tag{3.1}$$

onde os coeficientes a_i são os pesos de cada posição da predição.

Ao usarmos a janela de treinamento, estamos assumindo que os coeficientes usados para determinar o próximo pixel não variam dentro desta área, e podem ser usados para determinar a próxima posição de interesse. Para então determinarmos os melhores coeficientes, teremos que otimizá-los localmente usando a janela de treinamento causal. O critério de otimização usado é baseado no erro médio quadrático (MSE - *Mean Square Error*), dado por

$$\min(MSE) = \min\left\{\frac{1}{\#(M)} \sum_{\mathbf{X}(\mathrm{g}(n-k))\subset M} \left(\mathbf{X}(\mathrm{g}(n-k)) - \sum_{i=k-1}^{k-N} a_i \mathbf{X}(\mathrm{g}(n-i))\right)^2\right\} \tag{3.2}$$

onde o conjunto M denota os pixels dentro da janela de treinamento, e $\#(M)$ o número de elementos do conjunto M. Em notação matricial, temos

$$\min\{||\mathbf{y} - \mathbf{Ca}||_2\} \tag{3.3}$$

Onde a seqüência de treinamento será um vetor coluna $\mathbf{y} = [\mathbf{X}(\mathrm{g}(n-1))\ldots\mathbf{X}(\mathrm{g}(n-M))]^T$ de tamanho $M \times 1$ e a vizinhança de predição forma uma matriz de tamanho $M \times N$, onde

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

A escolha da norma L_2 como o critério de otimização se deve ao fato da simplicidade de implementação desta técnica e da maturidade com que esta técnica já foi empregada em diversos outros problemas. Os coeficientes ótimos são os que minimizam o erro, ou seja, a solução de $\min(||\mathbf{y} - \mathbf{Ca}||^2)$. Uma conhecida solução fechada para este problema é dado por

$$\mathbf{a} = (\mathbf{C}^T\mathbf{C})^{-1}(\mathbf{C}^T\mathbf{y}) \tag{3.4}$$

Aplicações para estimação por mínimos quadrados em processamento de sinais 1D assim como alguns algoritmos rápidos podem ser encontrados em [57]. Em [58] encontram-se diversas implementações rápidas de otimização usando mínimos quadrados (como fatorização LU e decomposição SVD) escritas em C padrão.

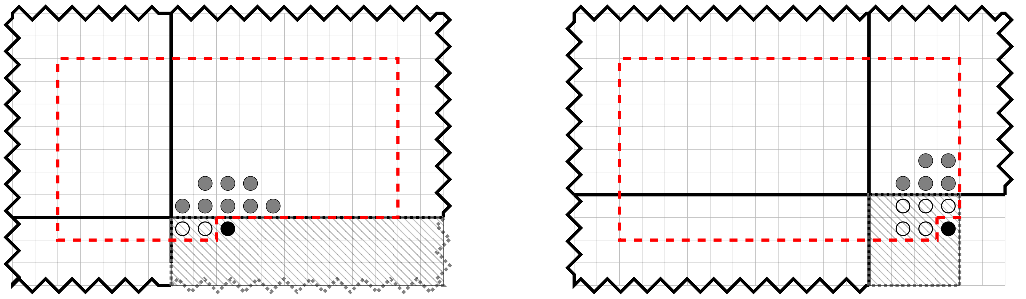
3.2 Incorporando o LSP ao MMP-FP

A partir da última versão do codificador MMP-FP, adicionamos o modo LSP ao conjunto de possíveis modos de predição a serem escolhidos pelo codificador. Em analogia com a seleção dos restantes modos, a escolha do melhor modo de predição é realizada através da análise do custo de codificação do modo de predição e de seu respectivo resíduo.

3.2.1 Adaptando o LSP para predição em blocos

O modo de predição LSP foi concebido para codificação sem perdas. Desta forma, os pixels da predição usados são os pixels mais próximos, o que no caso de uma varredura *raster* serão usados os pixels à esquerda ou acima da posição a ser predita. No caso da codificação de uma imagem em blocos, os pixels usados na janela de treinamento podem ainda não estar disponíveis, o que torna a implementação do LSP para blocos uma tarefa não-trivial. Em [59], apenas os valores de pixels pertencendo a blocos acima e à esquerda do bloco atual é que podem ser utilizados no cálculo da predição do pixel, mesmo para aqueles que estão nas posições mais afastadas desta vizinhança, o que acaba por apresentar uma perda da eficiência do preditor para posições distantes da borda do bloco.

Na implementação aqui proposta, flexibilizamos a restrição da vizinhança e es-



(a) LSP sem restrição

(b) LSP modificado devido à restrição da vizinhança

Figura 3.2: Implementação do LSP em blocos. As linhas pretas indicam as bordas dos blocos já codificados, enquanto que a área hachurada indica o bloco a ser predito com o LSP. Pixels representados pela cor cinza têm o seu valor reconstruído disponível para a adaptação do preditor. Pixels usados na adaptação que pertencem ao mesmo bloco usam o valor obtido com a predição LSP na sua respectiva posição, e não o valor reconstruído. O pixel preto representa a amosta atual.

colhemos sempre os pixels mais próximos para a predição. Nas posições que se encontram dentro do bloco, os valores dos pixels reconstruídos ainda não estão presentes, e para esses casos utilizamos o valor recém calculado na predição passada com o LSP, como mostra a Figura 3.2(a). Como ainda podemos ter casos onde pixels à direita não estão disponíveis, nestas situações a vizinhança e a janela de treinamento são alteradas conforme a Figura 3.2(b), para que a predição use somente posições já codificadas ou preditas. Nas bordas superior e esquerda da imagem, onde não há área suficiente para realizar o treinamento do LSP, o modo é desativado, e dessa forma apenas os modos do H.264/AVC são usados.

3.3 Resultados experimentais

O principal motivação para usar o LSP era melhorar o desempenho das curvas de taxa-distorção do algoritmo MMP para imagens suaves e com muita textura, sem provocar perdas no comportamento eficiente do MMP relacionado com a codificação de imagens não-suaves.

De maneira a explorar toda a capacidade de predição do modo LSP, um modelo de ordem alta foi utilizado, ou seja, uma vizinhança fixa de $N = 10$ elementos. O tamanho da janela de treinamento escolhido foi $T = 7$. Estudos empíricos realizados em [60] sugerem que janelas maiores do que 7 não melhoram o desempenho do método de predição LSP. Testes realizados no âmbito do codificador MMP também sugerem que a combinação ótima da ordem do preditor e o tamanho da janela de treinamento são dados por $N = 10$ e $T = 7$ (ver detalhes no Apêndice C).

As curvas de taxa distorção para imagens suaves podem ser visualizadas nas

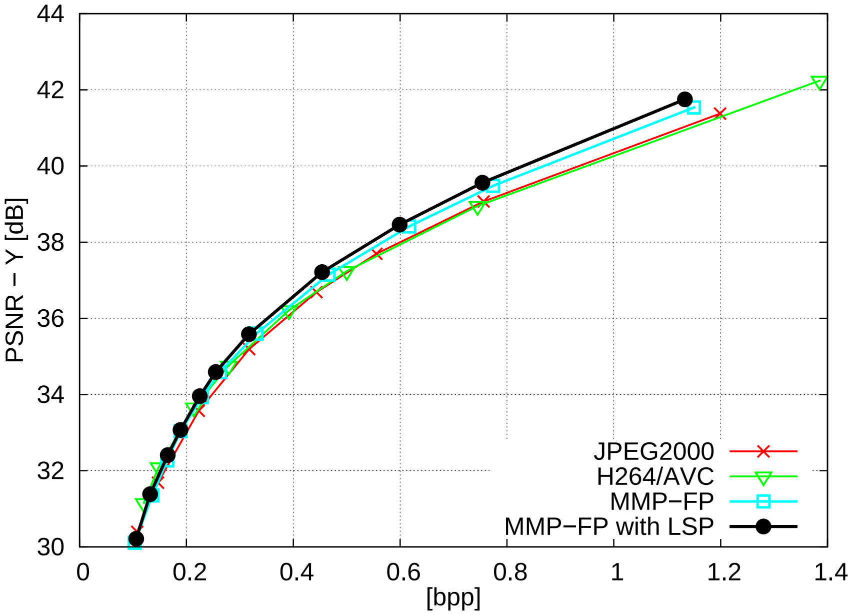
Figuras 3.3(a) e 3.3(b). Outro fato marcante, já mencionado anteriormente, é que o ganho do LSP depende da imagem a ser codificada. Para imagens suaves com textura complexa, como a imagem BARBARA, ganhos de PSNR de até 1,2 dB foram registrados, enquanto que os ganhos para a imagem Lena são menores, em torno de 0,25dB. Podemos também notar que o desempenho das curvas melhora com taxas médias e altas. Uma vez que o modo LSP depende diretamente da janela de treinamento utilizada na predição do pixel, taxas mais altas que contêm pixels reconstruídos mais precisos vão contribuir positivamente no treinamento. Para todas as imagens, podemos constatar o desempenho acima das curvas de taxa distorção dos algoritmos estado-da-arte usados para efeito de comparação, o JPEG2000 e o H.264/AVC Intra. Mais resultados se encontram no Apêndice C.

Para imagem compostas e com textos, como as imagens PP1205 e PP1209, as bordas dos elementos das imagens têm variações muito bruscas e ocorrem frequentemente, dificultando o processo de predição do LSP, tornando difícil o aprendizado de uma borda dentro da janela de treinamento. Pelo mesmo motivo, qualquer outro tipo de predição é ineficiente para este tipo de imagem, o que justifica o pior desempenho do MMP-Intra comparado ao MMP-FP para imagens compostas. Mesmo assim, a adição de mais um modo de predição não afetou o desempenho das curvas de taxa-distorção do algoritmo MMP para as imagens não-suaves, como podemos ver nas Figuras 3.4(a) e 3.4(b).

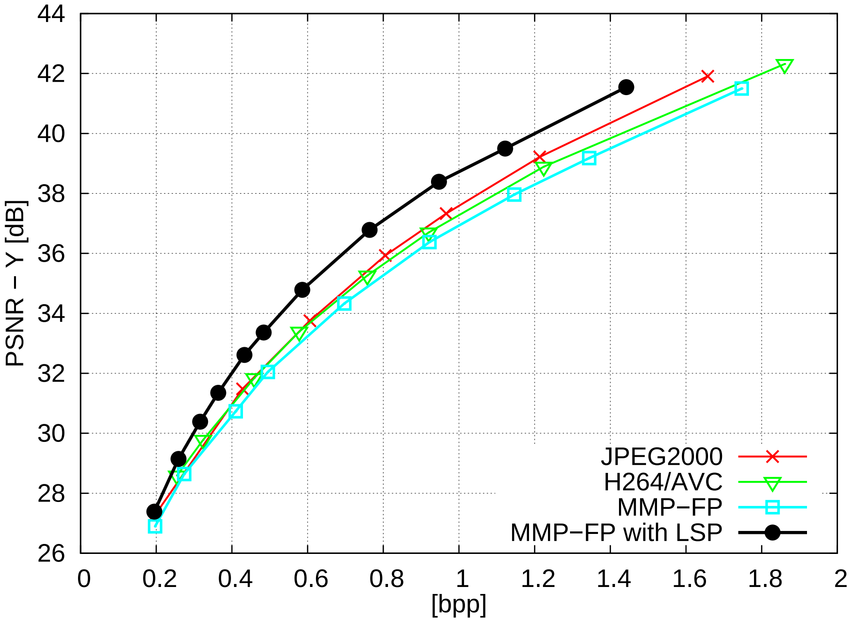
3.4 Conclusões

Neste capítulo um método de predição baseado no critério dos mínimos quadrados foi proposto. O novo modo de predição LSP é capaz de gerar boas predições para áreas com bordas de orientação arbitrária, o que acaba por resultar numa maior eficiência para o codificador MMP. Os ganhos em taxa-distorção chegam a 1 dB em algumas imagens, para taxas médias e altas, e não acarretaram perda alguma para imagens compostas.

Uma desvantagem do método é o aumento da complexidade computacional, devido à operação de inversão de matrizes necessária para a adaptação do preditor de cada pixel. Perspectivas de trabalhos futuros neste tema incluem implementações mais eficientes do método.

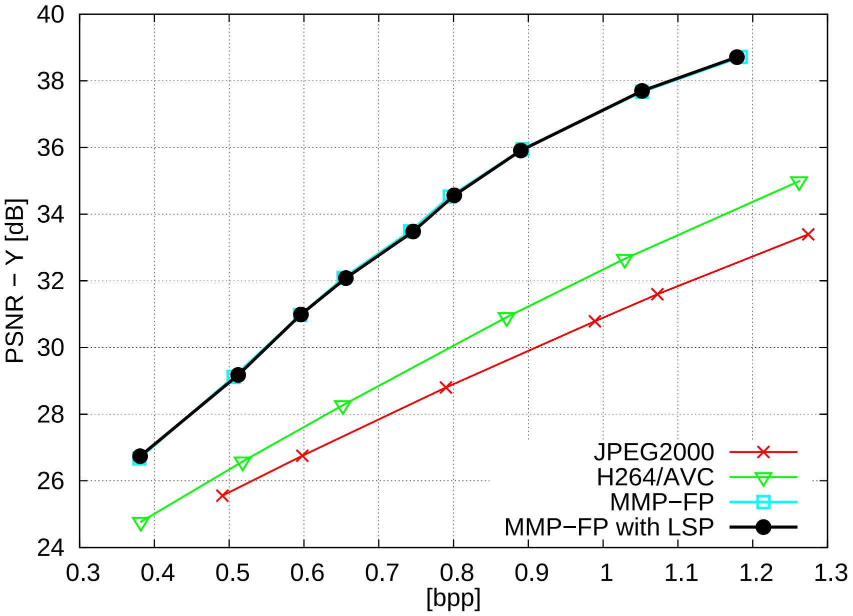


(a) Imagem LENA

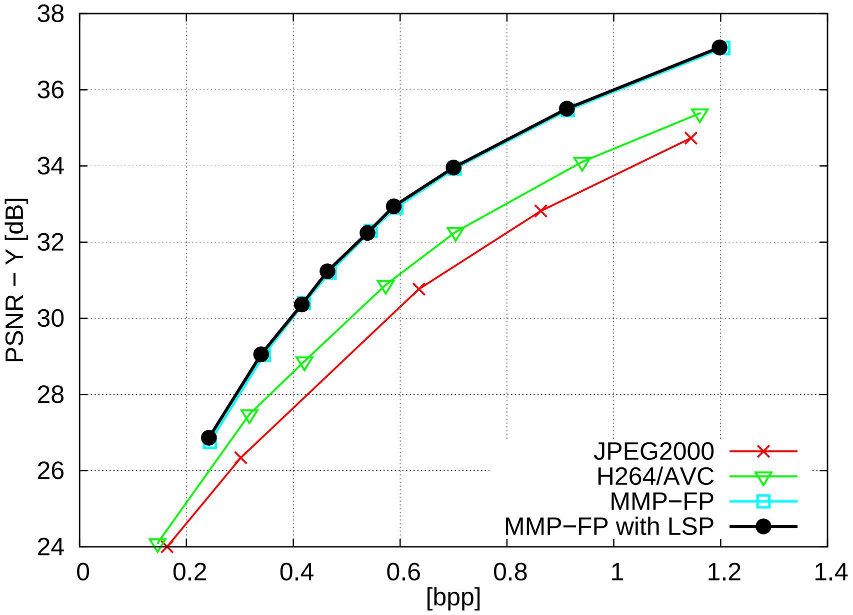


(b) Imagem BARBARA

Figura 3.3: Curvas de desempenho taxa-distorção para imagens suaves.



(a) Imagem PP1205



(b) Imagem PP1209

Figura 3.4: Curvas de desempenho taxa-distorção para imagens compostas.

Capítulo 4

Compressão sem perdas usando o MMP

4.1 Compressão de imagens sem perdas

Com a evolução tecnológica, os sinais discretos ganharam importância, dentro deles os sinais digitalizados como textos, imagens, áudio e vídeos. Os dispositivos de reprodução também evoluíram muito, a fidelidade exigida para esses sinais aumenta cada vez mais, o que nos torna cada vez menos tolerantes à baixa qualidade de reconstrução dos mesmos. A largura de banda para transmissão aumentou, a capacidade de armazenamento aumentou e ficou mais barata e acessível. Todos esses fatores fazem com que a importância dos algoritmos de compressão sem perdas volte a crescer relativamente aos algoritmos com perdas, para sinais onde a introdução de alguma perda não compromete a aplicação.

Aplicações como compressão de textos, imagens médicas ou ainda imagens de satélite, não toleram qualquer diferença entre o sinal original e o sinal reconstruído. Um erro de reconstrução numa radiografia pode mudar um diagnóstico, um artefato de compressão pode invalidar uma série de imagens de satélites que não poderão ser feitas novamente. Nestes casos a compressão sem perdas não é uma opção, é um item obrigatório.

A maioria dos algoritmos de compressão sem perdas realizam duas tarefas: o modelamento da imagem e a sua codificação. O objetivo de um codificador é através destas duas operações atingir a taxa máxima de compressão de uma fonte, ou seja o tamanho mínimo necessário para representar um sinal, cujo valor mínimo é dado pela entropia da fonte, também conhecido como limite de Shannon [61].

Desde a sua concepção, o MMP tem a capacidade inata de codificar uma imagem sem perdas. Para tanto, basta que o dicionário da escala 1×1 seja um dicionário completo. Dessa forma, se o casamento não ocorrer em escalas maiores, sempre poderemos transmitir o valor do pixel através de um bloco de escala 1×1. Os ganhos de codificação do MMP ocorrem quando temos um casamento em escalas maiores, e conseguimos codificar um bloco inteiro, ou seja diversos pixels em conjunto, com apenas um índice.

No Apêndice D, é obtido pela primeira vez um limite teórico para o desempenho do MMP atuando como um codificador de imagens sem perdas. Mostramos que o MMP é capaz de atingir a entropia de uma fonte estacionária, ergódica, sem memória com um alfabeto limitado. Um aspecto interessante deste resultado é o fato de explicitar a importância do passo de predição para a convergência do algoritmo.

4.2 Modificando o algoritmo MMP para codificação sem perdas

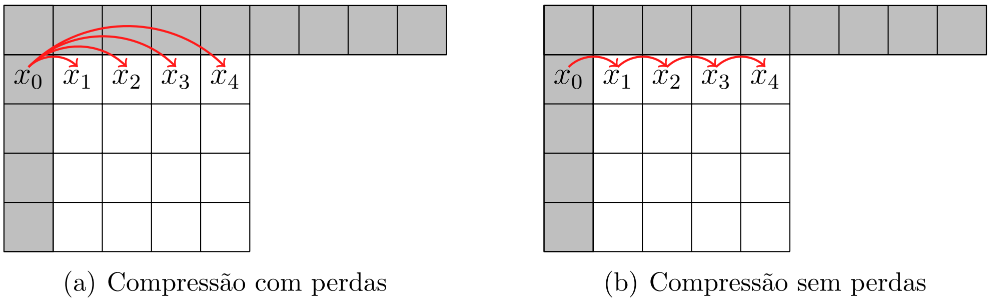
O algoritmo MMP irá escolher os índices do dicionário para representarem a imagem a ser codificada através de um algoritmo de otimização usando multiplicadores de Lagrange, baseado num critério voraz. O critério é dito voraz uma vez que a decisão de selecionar o melhor índice leva apenas em conta o bloco que está sendo codificado atualmente, e não considera o efeito dessa codificação para outros blocos que ainda serão codificados. Como o dicionário usado para codificar a imagem vai crescendo a medida que a imagem vai sendo codificada, diferentes valores vão sendo acrescentados ao dicionário. Estes múltiplos valores podem representar o mesmo elemento da imagem de diversas formas diferentes. Ou seja, a base formada pelas palavras presentes no dicionário do MMP será uma base *overcomplete* [62].

Dessa forma podemos ter também diversas formas de alcançar o ponto de distorção zero. Nesta caso, o algoritmo de otimização foi modificado para também levar em consideração a taxa necessária para chegarmos no ponto de distorção zero. Seguindo um critério voraz, é escolhido a segmentação e os índices que levam à codificação zero, porém com a menor taxa possível para o bloco. O algoritmo de otimização irá procurar todas as combinações possíveis de palavras-códigos, *flags* de segmentação e modos de predição que resultam em distorção zero, e dentro destas possibilidades, escolher a combinação que necessita de menos bits.

4.2.1 Predição em algoritmos sem perda

Uma técnica muito utilizada em compressão sem perdas é o uso de um passo de predição antes do codificador entrópico, com a finalidade de descorrelacionar as amostras, e dessa forma codificar apenas um resíduo da predição, contendo valores provavelmente próximos de zero. Porém, vale ressaltar que esta técnica é empregada de forma eficiente em imagens com alta correlação espacial, ou seja, imagens suaves, porém não consegue alcançar um bom desempenho para imagens compostas.

Os métodos de predição sem perdas, como o MED, o GAP ou o LSP, foram utilizados em conjunto com o MMP (veja Apêndice D), porém os melhores resultados foram atingidos ao usarmos uma adaptação para codificação sem perdas do esquema de predição atual, baseado nos modos de predição do H.264/AVC Intra e no LSP (ver detalhes no Capítulo 3). LEE *et al.* [63] descrevem a alteração dos modos de predição intra vertical e horizontal para codificação sem perdas, incorporada pelo padrão H.264/AVC [64]. Analogamente, os modos de predição do MMP também foram alterados para compressão sem perdas.



(a) Compressão com perdas

(b) Compressão sem perdas

Figura 4.1: Exemplo de alteração do modo de predição horizontal Intra para compressão sem perdas. Enquanto que no caso com perdas, a predição usa sempre o pixel reconstruído do bloco vizinho, no caso com perdas é usado para predição o pixel original mais próximo.

O método descrito usa para a predição o pixel mais próximo do pixel a ser predito, e não a vizinhança do bloco, que pode estar longe demais para pixels dentro do bloco. A Figura 4.1 ilustra o processo aplicado ao modo de predição horizontal. A mesma operação é aplicada ao modo vertical, devido à sua semelhança com o processo de predição DPCM (*Differential Pulse Code Modulation*), onde as diferenças são codificadas. O codificador MMP que usa os modos alterados será chamado de MMP-DPCM.

Os resultados apresentados em [64] mostram um ganho de desempenho na capacidade de compressão do codificador sem perdas quando alteramos os modos horizontais e verticais, sem aumento significativo da complexidade. No caso do MMP, os mesmos modos vertical e horizontal foram alterados de acordo. O mesmo princípio de usarmos os valores originais mais próximos foi aplicado ao modo LSP também. Ao invés de usarmos os valores recentemente preditos para posições de dentro do bloco, usamos os valores originais. Uma vez que teremos a reconstrução perfeita dos pixels, tanto o codificador quanto o decodificador podem realizar o mesmo treinamento, bastando apenas que o decodificador atualize os valores preditos com o resíduo recebido. Ou seja, é necessário o decodificador receber todos os resíduos de um bloco, para a partir deles obter a predição correta.

4.2.2 Restrição do histograma

Podemos refinar ainda mais a estimativa da probabilidade do erro de predição se condicionarmos esta probabilidade ao conhecimento da predição. Devido ao fato de termos apenas 256 possíveis valores para uma imagem em tons de cinza com 8 bits de profundidade, conhecida a predição, sabemos que o resíduo só pode estar numa faixa que varia entre $[-x_{\text{pred}}, 255 - x_{\text{pred}}]$. Logo, se condicionarmos a probabilidade do erro de predição ao valor da predição, iremos zerar os valores de resíduo que são improvavéis. Assim podemos re-escalonar a probabilidade de ocorrência do resíduo com um valor ligeiramente maior, levando em consideração somente os possíveis valores do resíduo. A Figura 4.2 mostra como fica a distribuição de probabilidade do resíduo condicionada ao valor da predição. Note porém que esta técnica é incompatível com o modo de predição DPCM, pois para a restrição do histograma precisamos da predição para obter o resíduo, enquanto que nos modos DPCM precisamos dos resíduos para obter a predição. Detalhes de implementação desta técnica no codificador MMP se encontram no Apêndice D.

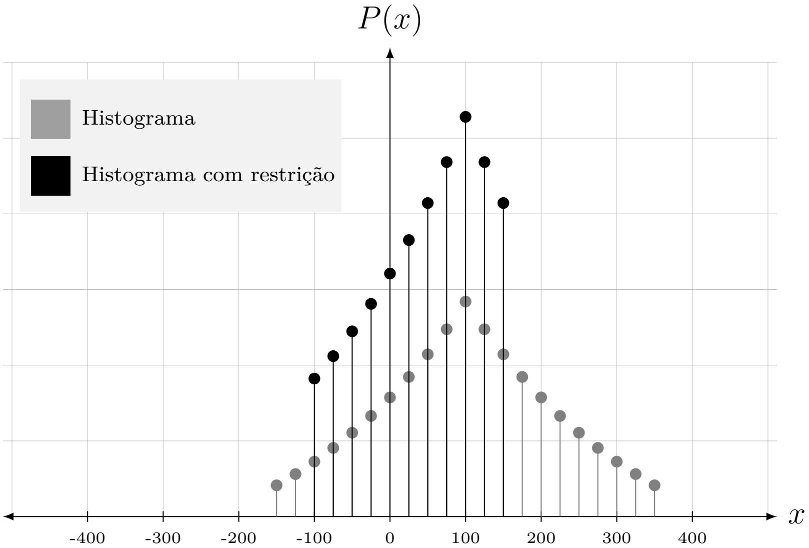
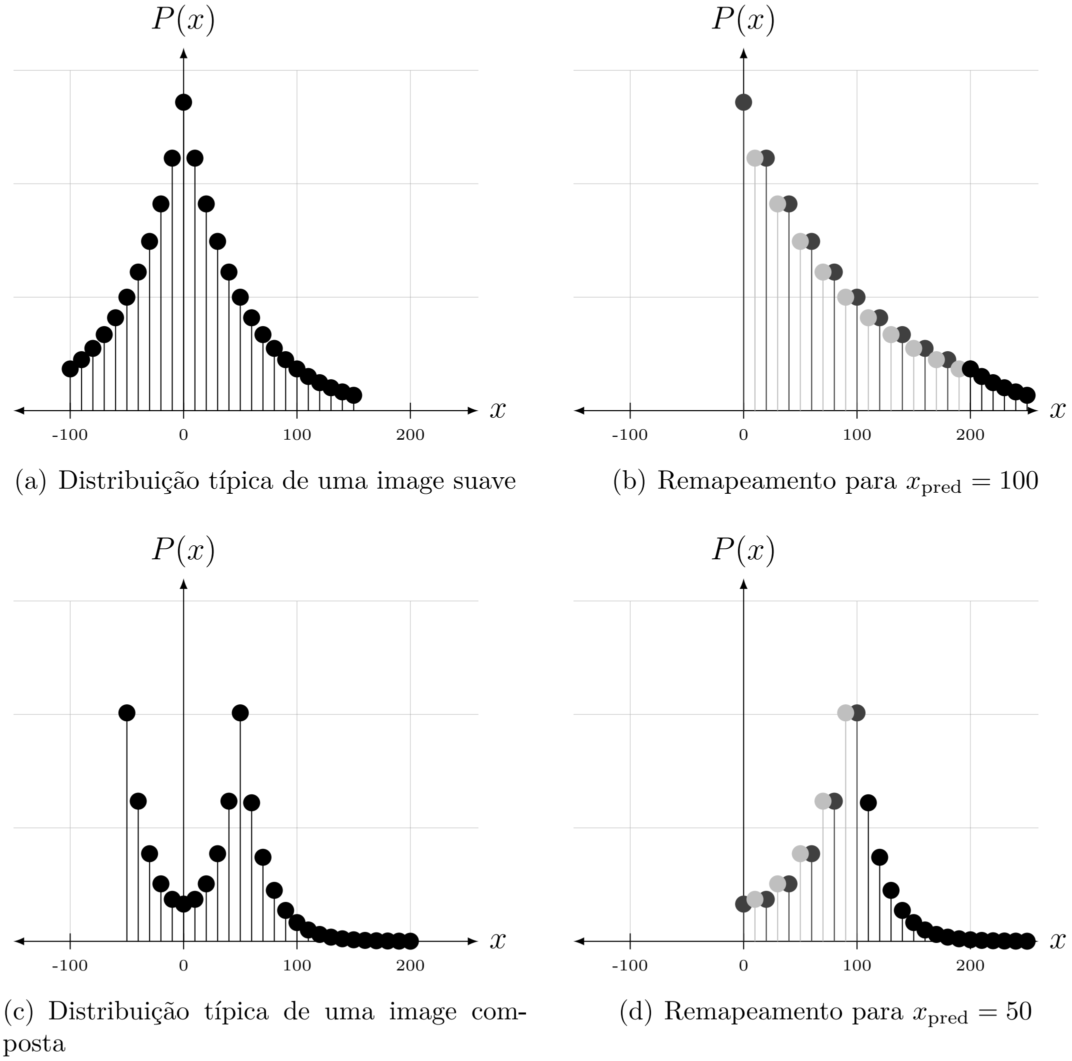


Figura 4.2: Restrição do histograma para o valor de predição $x_{\text{pred}} = 100$. O gráfico ilustra um exemplo de distribuição laplaciana, um modelo utilizado frequentemente para modelar a distribuição do resíduo da predição.

4.2.3 Remapeamento do erro de predição

Uma imagem de 8 bits pode assumir 256 valores diferentes. A diferença entre o pixel real e a sua predição resultam valores que variam entre -255 e 255, necessitando de 9 bits para serem representados. Porém como já mencionado, uma vez conhecida a predição, os possíveis valores de resíduo se encontram no intervalo entre $[-x_{pred}, 255 - x_{pred}]$, que contém 256 valores, e podem ser representados com 8 bits. Um procedimento comumente realizado por codificadores como o CALIC [65] e o JPEG-LS [66] é o remapeamento do resíduo. Uma revisão das diferentes funções de remapeamento podem ser encontrados em [67].



(a) Distribuição típica de uma image suave

(b) Remapeamento para $x_{\text{pred}} = 100$

(c) Distribuição típica de uma image composta

(d) Remapeamento para $x_{\text{pred}} = 50$

Figura 4.3: Exemplo de remapeamento do resíduo para dois tipos de distribuição. Para áreas suaves, onde o resíduo de predição terá uma característica laplaciana, como visto na Figura (a), o mapeamento não irá alterar o comportamento da função como mostra a Figura (b). Já no caso de uma distribuição bimodal como o da Figura (c), como é o caso de imagens compostas, o remapeamento irá atribuir altas probabilidades para transição acentuadas, como se pode ver na Figura (d).

O remapeamento do sinal residual irá juntar as probabilidades periféricas com a probabilidade central, como é mostrado na Figura 4.3 (mais detalhes se encontram no Apêndice D). O comportamento da distribuição residual, que possui valores altos próximos de zero, não será afetado com o remapeamento. Este procedimento é especialmente adequado para imagens compostas, pois irá atribuir altas probabilidades para transições acentuadas. Detalhes de implementação desta técnica no codificador MMP se encontram no Apêndice D.

4.2.4 Malha de realimentação para o erro de predição

Uma vez que o passo de predição não remove a redundância estatística da imagem por completo, os resíduos de predição ainda mantém uma correlação espacial entre si. Os valores dos resíduos de posições vizinhas podem então ser usados para melhorar a predição do pixel atual. Geralmente, a média dos resíduos vizinhos é usada como um fator de correção do valor predito, e um contexto é utilizado para acelerar a adaptação do fator de correção. Uma vez que o mesmo contexto volte a ocorrer, é possível reajustar o valor da predição através de uma malha de realimentação, que irá corrigir o valor predito retirando a média dos resíduos passados, pertencentes ao mesmo contexto. Esta técnica é aplicada em codificadores como o JPEG-LS [66] e o CALIC [65], e foi também implementada para o MMP. O contexto usado no caso do MMP foi a dimensão do bloco. Detalhes de implementação desta técnica no codificador MMP se encontram no Apêndice D.

4.3 Resultados experimentais

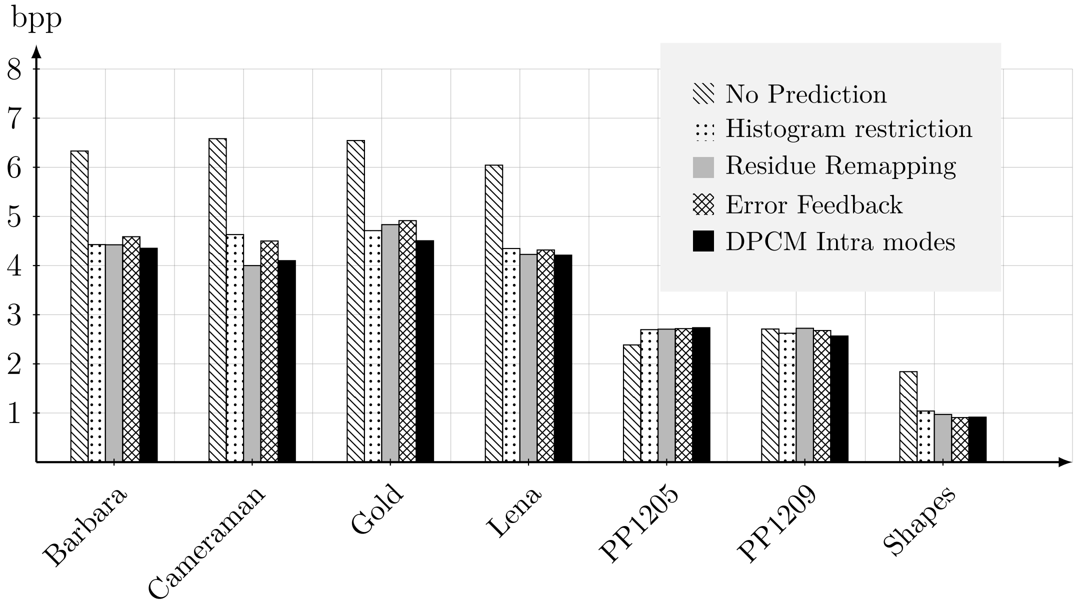


Figura 4.4: Resultados para diversas técnicas de compressão sem perdas. As imagens abrangem um espectro de imagens suaves e compostas, de diversos tamanhos. Os resultados mostram a eficiência de compressão das propostas descritas, avaliados em bits por pixel.

A Figura 4.4 mostra os resultados de compressão de diversas imagens usando as técnicas propostas anteriormente. Podemos notar que a predição é benéfica para imagens suaves e para a imagem geradas por computador, e que a alteração nos modos de predição ainda proporcionou ganhos ao MMP, atingindo as maiores taxas de compressão. Já no caso de imagens compostas, o MMP sem o passo de predição está entre os métodos que atingiu os melhores resultados. Porém os ganhos do método com predição foram maiores para as imagens suaves e computadorizadas, o que nos levou a adotar os modos de predição DPCM ao MMP, no lugar de outras técnicas de codificação do resíduo (note que os métodos mencionados são mutuamente exclusivos).

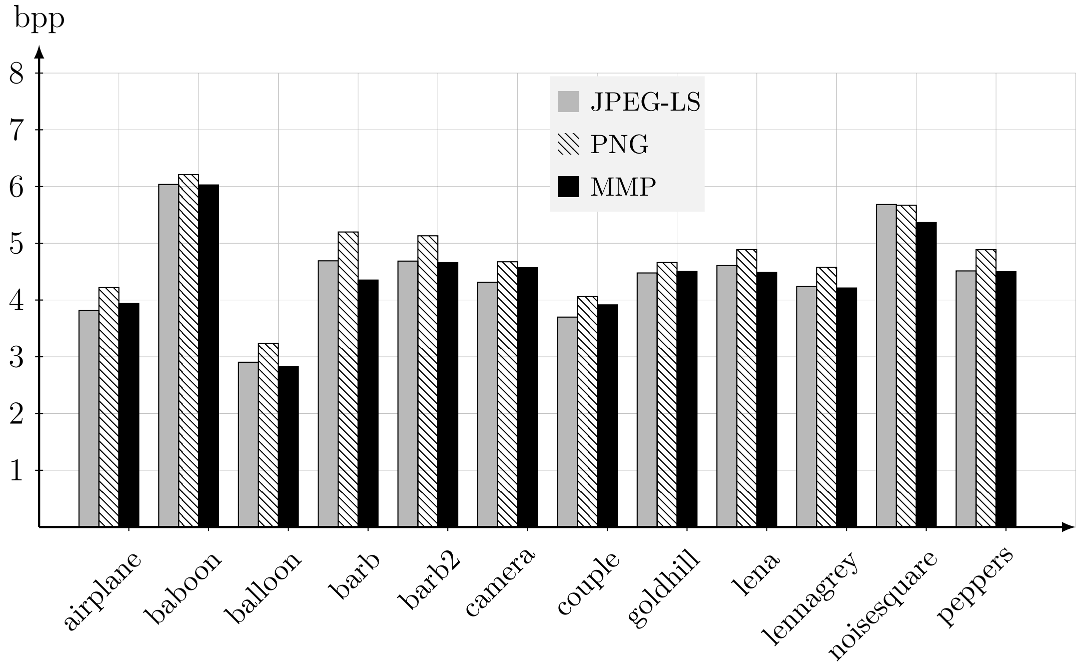


Figura 4.5: Comparação de resultados de compressão sem perdas para imagens suaves.

As Figuras 4.5 e 4.6 comparam o resultado do MMP usando os modos de predição com a adaptação DPCM para compressão sem perdas, com o JPEG-LS e o PNG. O JPEG-LS é o padrão de compressão sem perdas para imagens suaves, enquanto que o PNG, a semelhança do MMP, também utiliza um dicionário de padrões para codificação sem perdas. Note que o PNG apresenta bons resultados para imagens compostas, porém não consegue ser tão eficiente para imagens suaves. O contrário acontece com o JPEG-LS. O MMP conjuga o melhor dos dois algoritmos, gerando resultados competitivos para todos os tipos de imagens. Podemos então concluir que o método proposto de codificação sem perdas usando o MMP é eficaz para o uso com qualquer tipo de imagem, sem a necessidade de se alterar o algoritmo.

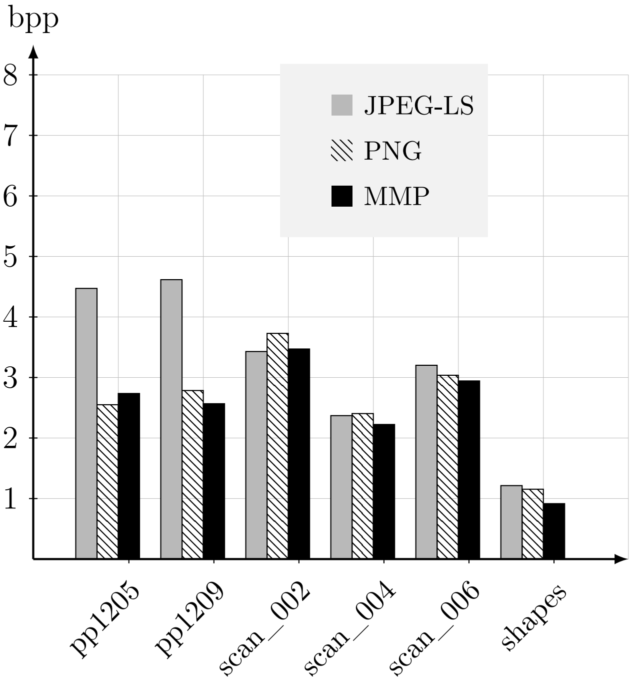


Figura 4.6: Comparação de resultados de compressão sem perdas para imagens compostas e geradas por computador.

4.4 Conclusões

Como vimos neste capítulo, o MMP-FP atuando na compressão de imagens sem perdas com o esquema de predição DPCM [64] foi o mais eficiente. Isto corrobora o resultado teórico obtido no Apêndice D, que indica que o desempenho do algoritmo deve melhorar se o passo de predição conseguir descorrelacionar os pixels da imagem.

No caso das imagens suaves, o MMP apresenta um desempenho mais eficiente do que o apresentado por outros codificadores baseados em casamento de padrões, como o formato PNG. Ao usar o passo de predição, o MMP consegue atingir taxas de compressão maiores, especialmente quando os pixels têm uma alta correlação espacial. Já no caso de imagens compostas, o passo de predição não consegue descorrelacionar os pixels vizinhos tão eficientemente como no caso das imagens suaves. Porém, a otimização da predição em conjunto com a codificação do resíduo irá fazer com que o algoritmo escolha modos de predição simples (geralmente o modo MFV), que não irão modificar as características da imagem, e os padrões recorrentes poderão ser codificados de forma eficiente com o uso do dicionário.

Pelos resultados aqui apresentados, e também pelos resultados de codificação com perdas apresentados no Capítulo 3, podemos concluir que o MMP apresenta resultados competitivos para a codificação de qualquer tipo de imagem, em qualquer taxa alvo desejada.

Capítulo 5

Compressão multivistas usando o MMP

5.1 O novo formato para imagens 3D: textura com mapas de profundidade

É cada dia mais evidente que chegamos na era do 3D. Novos filmes em formato 3D são lançados quase todas as semanas, novas televisões, jogos portáteis (Nintendo 3DS), tablets ou celulares *3D-ready* estão disponíveis nos mercados, os operadores de televisão usam a sua infra-estrutura atual para transmitir vídeo estereoscópico. A difusão de meios de criação, transmissão e consumo de material 3D, isto é, de toda a cadeia de operação do 3D, vem fomentar o interesse nesta área [68].

Ainda existem diversos desafios a serem superados para a aceitação em massa do 3D, não só nos cinemas, mas também na casa do usuário. Atualmente, a visualização de conteúdo 3D é na maioria dos casos feita com auxílio de óculos 3D. As vistas estereoscópicas são multiplexadas temporalmente, e os óculos 3D permitem a visualização de cada imagem apenas por um único olho, proporcionando uma visão binocular e criando a impressão de profundidade. No entanto, o efeito prolongado do uso dos óculos é causa comum de dores de cabeça e cansaço. Porém, os monitores auto-estereoscópicos oferecem uma solução para a visualização de vídeo 3D sem a necessidade de óculos, com a multiplexação espacial de diversas vistas e o uso de lentes difusoras. O uso de múltiplas vistas permite que o telespectador possa se movimentar livremente e ainda assim continuar com a impressão de profundidade. No entanto, ainda não está definido o número de vistas necessárias para o melhor efeito 3D. Outro agravante é o fato dos produtores de conteúdo não estarem propensos a aumentar ainda mais os custos de gravação, para capturarem mais vistas.

Motivados pela crescente demanda deste setor, o MPEG está finalizando a criação de um padrão para o novo formato de vídeos 3D, que é capaz de atender às necessidades tanto dos produtores de conteúdo quanto dos fabricantes de monitores. Com o auxílio de técnicas de visão computacional, o novo padrão será capaz de, com um número reduzido de vistas, sintetizar a quantidade de vistas necessárias ao telespectador, sem a necessidade de se capturar novas vistas. Dessa forma, a produção de conteúdo 3D fica independente do seu modo de visualização, e o novo padrão será eficaz tanto para os produtores quanto para os consumidores [69]. Para que o novo padrão seja ainda compatível com os padrões vigentes de multivistas [9] e vídeo mais profundidade [70, 71], o formato previsto deverá ser uma conjugação de múltiplas vistas com seus respectivos mapas de profundidade, também conhecido como multivistas mais profundidade. Uma revisão mais detalhada sobre a atividade de padronização em torno do novo formato de vídeo 3D pode ser encontrada no Apêndice E.

O uso de múltiplas vistas acarreta um aumento significativo na quantidade de dados para a codificação. O padrão de codificação de múltiplas vistas, o MVC, explora a correlação entre vistas, porém é conhecido que a sua taxa de compressão resultante cresce linearmente com o número de câmeras [72]. Através de algoritmos de síntese de vídeo, podem-se gerar vistas intermediárias, com o auxílio das vistas laterais e de seus respectivos mapas de profundidade. Os algoritmos de renderização de vistas virtuais usando a profundidade, também conhecidos como algoritmos DIBR (*Depth Image Based Rendering*), aplicam os conceitos de geometria projetiva para obter a textura de uma vista qualquer a partir das vistas adjacentes. Além disso, os mapas de profundidade requerem menos bits para serem codifados, sendo portanto uma solução eficiente para o problema crítico de aumento de taxa em sistemas multivistas.

Os mapas de profundidade são imagens que têm características bem distintas de imagens de textura. Eles apresentam geralmente superfícies suaves, sem textura qualquer, com bordas bem definidas [73]. Além da sua construção particular, os efeitos dos artefatos introduzidos na codificação são diversos, e estão particularmente relacionados com o software de síntese e de codificação usados no processo. Muitas propostas para codificação de mapas de profundidade foram feitas, desde adaptações de codificadores comuns [74–76] até a proposta de novos codificadores específicos para mapas de profundidade [77–81]. No entanto, a maioria delas têm dificuldade em preservar o elemento crítico dos mapas de profundidade: as bordas bem definidas dos objetos de uma cena. Os artefatos introduzidos em áreas de borda, isto é, com conteúdo de alta frequência, causa a mistura da textura de elementos no plano fontral com a textura do plano de fundo da imagem. Isso acarreta uma diluição das bordas dos objetos, e o aparecimento de pixels estranhos à textura de componentes da imagem, vindos de outras regiões da imagem [82].

Nas próximas seções, vamos propor o uso do MMP como um codificador de mapas de profundidade. O MMP já se mostrou eficaz para codificar imagens suaves e compostas, e suas capacidades de adaptação às características da imagem podem ser particularmente eficazes para a codificação de imagens com características tão específicas como os mapas de profundidade.

5.2 Usando o MMP para codificar mapas de profundidade

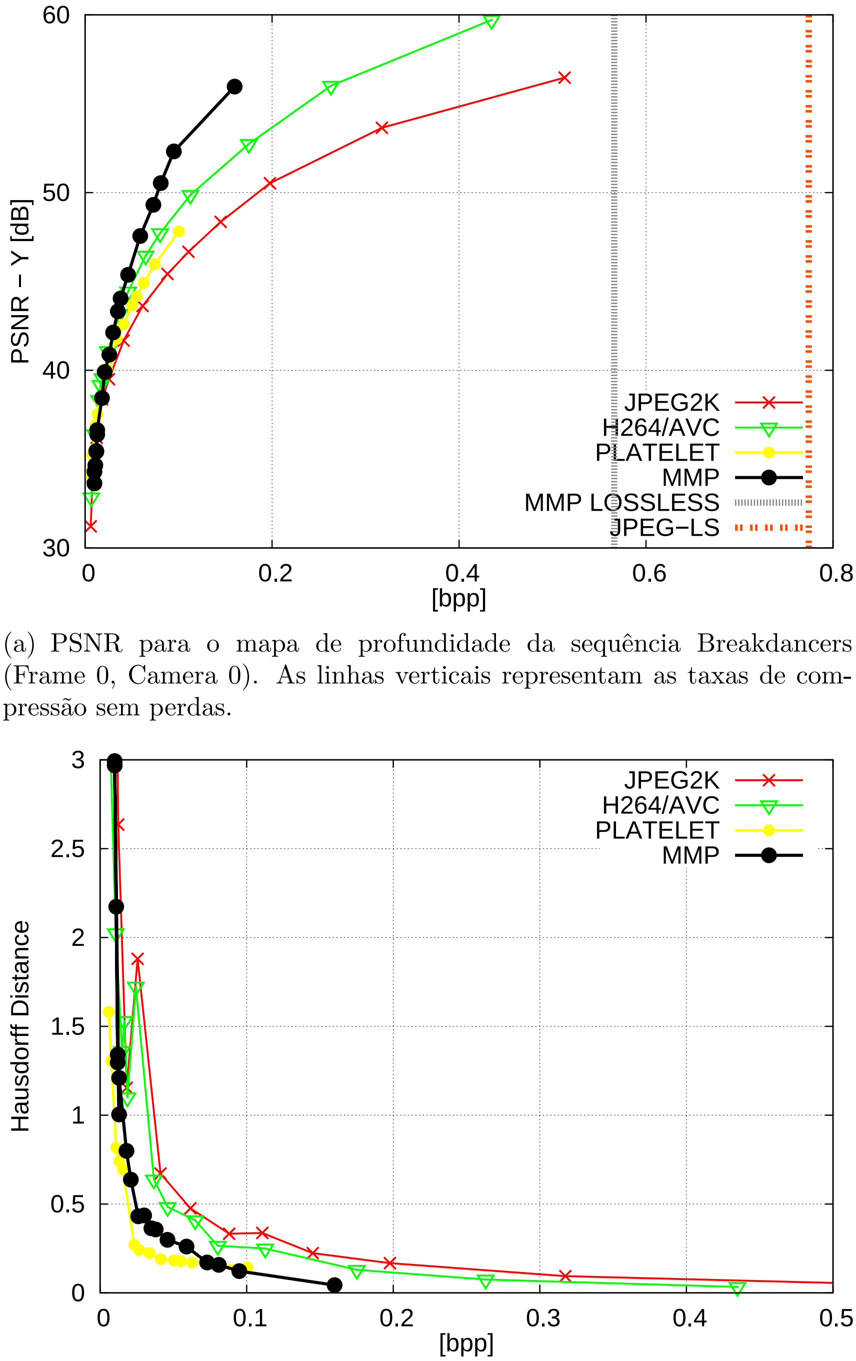
Como já foi mencionado anteriormente, as imagens de profundidade apresentam áreas de baixa frequência sem textura alguma e bordas muito bem definidas, resultante em áreas de alta frequência. Essa é uma característica vantajosa para o MMP, que consegue aprender os poucos padrões rapidamente e codificá-los eficientemente.

O MMP apresenta características semelhantes ao codificador de mapas de profundidade baseado em funções *platelet* apresentado em [78], conhecido como Platelet. Assim como este codificador, o MMP também utiliza uma segmentação flexível para codificar as bordas dos objetos por aproximações de blocos suaves. Porém ao aprender o novo padrão através da concatenação dos padrões suaves utilizados, o MMP é capaz de reutilizar esse padrão para codificar novas bordas, que venham a ocorrer durante a codificação. Desta forma o MMP evita a nova custosa segmentação da imagem, e portanto é mais eficiente que o Platelet, no sentido taxa-distorção. Outra vantagem do MMP em relação ao Platelet é o fato do MMP poder ser usado diretamente para a codificação das vistas, o que não é o caso quando falamos do Platelet.

5.2.1 Avaliação das vistas reconstruídas usando mapas de disparidade codificados

Uma análise objetiva da qualidade dos mapas de profundidade codificados deve ser feita. Como os mapas não serão visualizados, devemos ter cuidados ao utilizarmos medidas comumente usadas em análise de imagens suaves. Tais medidas podem não ser adequadas para avaliar os artefatos de codificação, uma vez que elas não levam em conta os efeitos dos artefatos no processo de síntese.

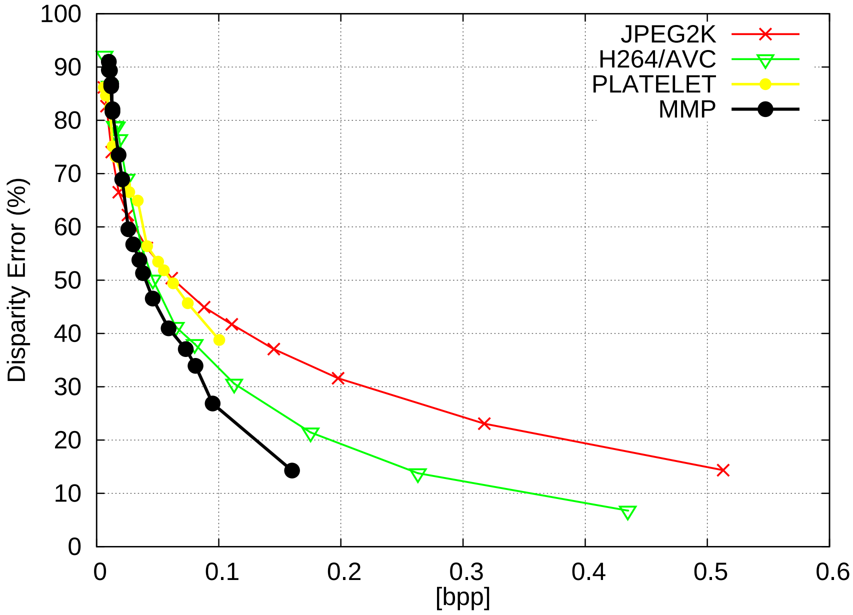
As Figuras 5.1 e 5.2 apresentam diversas medidas objetivas para avaliar mapas de profundidade, desde valores comumente utilizados em análise de imagens, como o PSNR, assim como medidas propostas em outras publicações relacionadas com a codificação de mapas de profundidade, como a distância de Hausdorff [78] (usada para medir distorção entre estruturas *mesh* 3D) ou a percentagem de erros da imagem [81]. Ainda sugerimos o uso do índice SSIM [83], que é uma medida qualitativa da



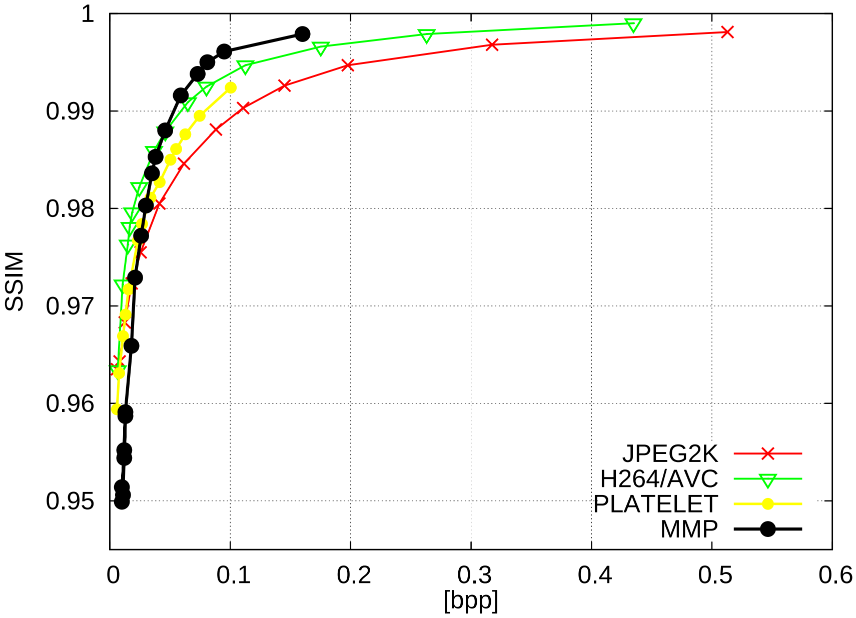
(a) PSNR para o mapa de profundidade da sequência Breakdancers (Frame 0, Camera 0). As linhas verticais representam as taxas de compressão sem perdas.

(b) Distância de Hausdorff para o mapa de profundidade da sequência Breakdancers (Frame 0, Camera 0)

Figura 5.1: PSNR e distância de Hausdorff, medidas qualitativas para mapas de profundidade da sequência Breakdancers, fornecidos pela Microsoft. O arranjo de câmeras para as sequências capturadas forma um meio arco e as câmeras não estão alinhadas.



(a) Percentagem de erros para o mapa de profundidade codificado da sequência Breakdancers (Frame 0, Camera 0)



(b) Índice SSIM para o mapa de profundidade codificado da sequência Breakdancers (Frame 0, Camera 0)

Figura 5.2: Percentagem de erros e SSIM, medidas qualitativas para mapas de profundidade da sequência Breakdancers, fornecidos pela Microsoft.

imagem, baseada na preservação da estrutura dos objetos da imagem. Em todas as medidas podemos ver a clara vantagem de usarmos o MMP, especialmente em taxas médias e altas. Para taxas baixas, o MMP não consegue adaptar o seu dicionário de maneira adequada, e apresenta problemas como um efeito de blocos demasiado. Mais resultados se encontram no Apêndice F.

Reconstrução de vistas usando textura codificada com o MMP

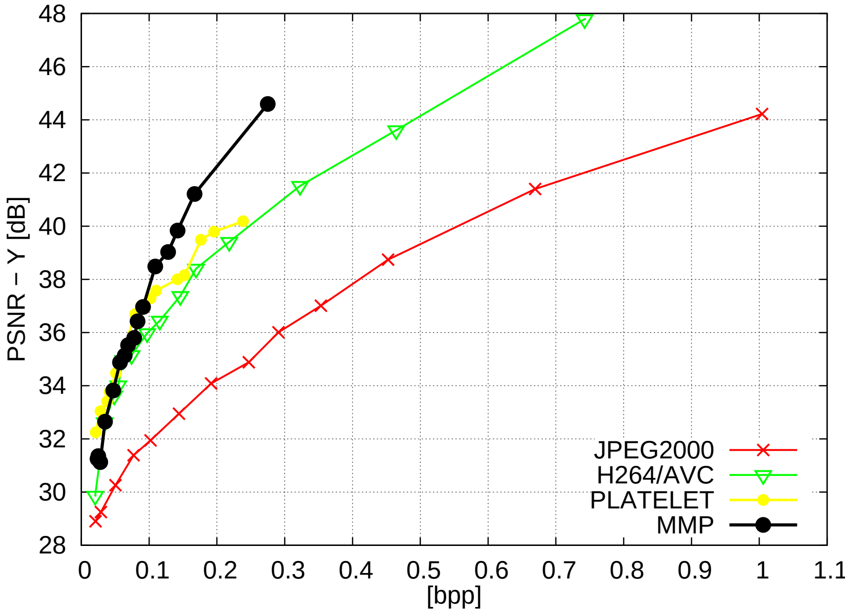
Apesar das medidas objetivas mostrarem a vantagem do uso do MMP, esta característica fica clara quando apresentamos o PSNR das vistas reconstruídas utilizando os mapas codificados, ao compararmos com a reconstrução usando textura e mapas de profundidade sem codificação.

A preservação das bordas gera imagens sintetizadas com maior qualidade objetiva e subjetiva. Já para taxas muito baixas, todas as propostas testadas não apresentam nenhuma solução viável para a reconstrução de vistas, já que todas as vistas apresentavam artefatos inaceitáveis. A Figura 5.3 mostra o desempenho taxa-distorção do MMP em duas situações: usando a textura original, ou usando a textura codificada com o H.264/MVC. Em ambos os casos, o MMP apresentou ganhos relativos aos outros codificadores testados, principalmente para taxas médias e altas. Mais detalhes e resultados se encontram no Apêndice F.

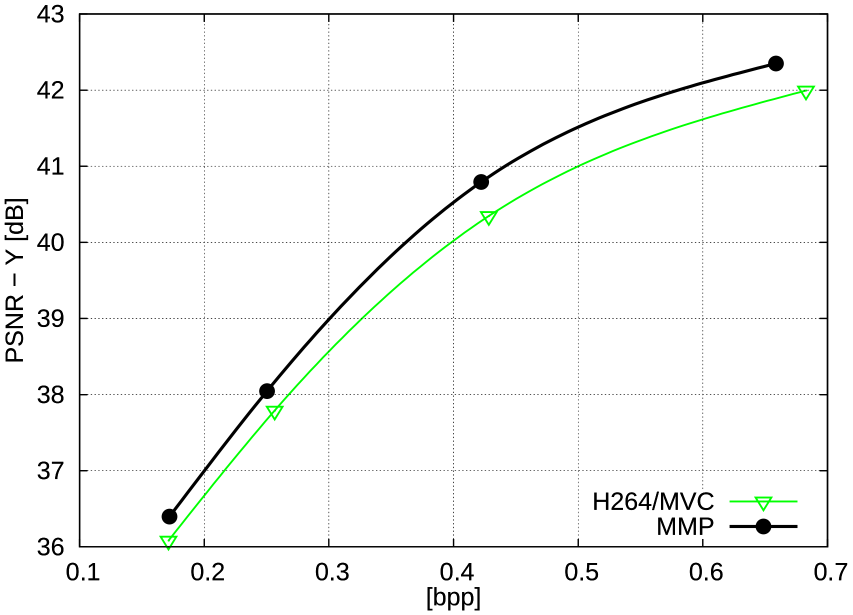
5.2.2 Restrição de bordas para codificação de mapas de disparidade

As bordas dos objetos presentes nos mapas de disparidade representam um desafio para a codificação. A sua preservação é fundamental para obtermos uma qualidade final de vistas reconstruídas aceitável. Diversas propostas foram feitas para a codificação de mapas de profundidade com uma atenção diferenciada para as bordas [76, 84, 85]. Aqui propomos uma alteração ao algoritmo do MMP para diminuir a distorção nessa região crítica dos mapas de disparidade, que iremos denominar MMP *edge-aware*. Adicionamos uma restrição ao algoritmo de otimização, onde não é permitido o uso de palavras-código que gerem uma distorção maior que um limar pré-definido, em regiões de borda marcadas por uma máscara. Mais detalhes sobre o método se encontram no Apêndice F.

A Figura 5.4 mostra os resultados obtidos com a proposta. O algoritmo proposto conseguiu eliminar em grande parte os artefatos da reconstrução, devido a uma melhor preservação das bordas dos objetos. No entanto, o desempenho taxa-distorção do algoritmo não foi satisfatório, uma vez que o gasto com os bits para a codificação das bordas foi demasiado. Uma característica interessante de se ressaltar nos resultados é o fato do algoritmo, em altas taxas, ter um comportamento semelhante ao

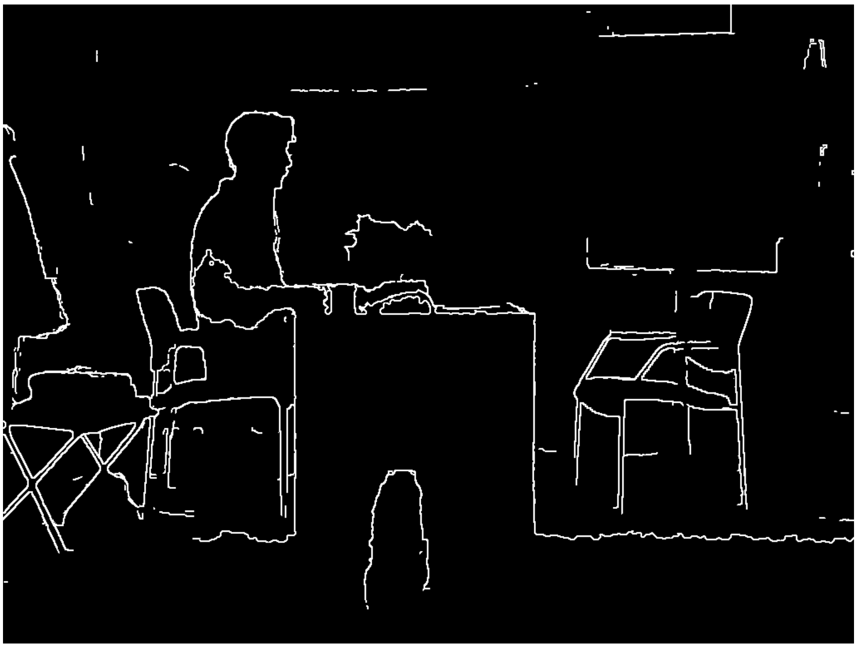


(a) Vista reconstruída da sequência Ballet (Frame 0, Camera 4), usando a textura sem codificação

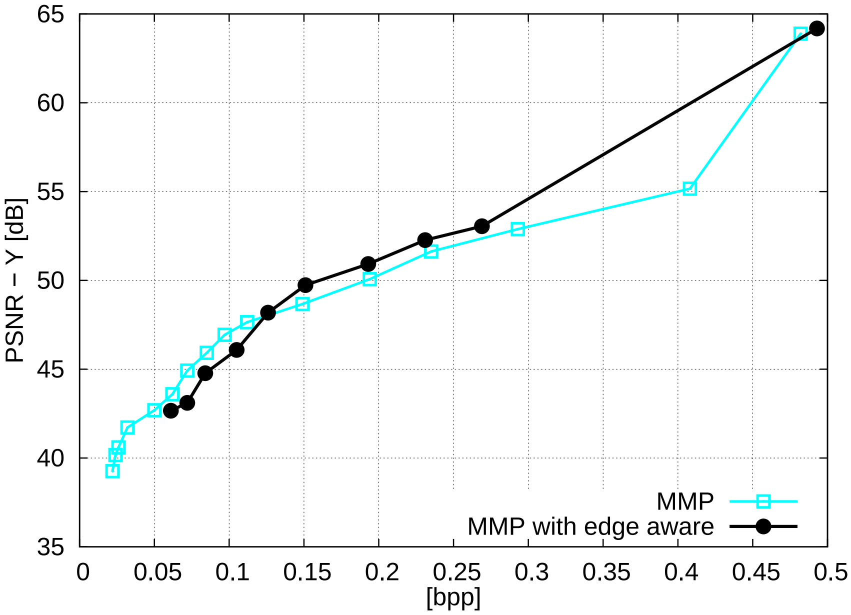


(b) Vista reconstruída da sequência Book Arrival (Frame 0, camera 9), usando a textura codificada com o H.264/MVC

Figura 5.3: Desempenho da vista reconstruída com mapas de profundidades codificados. O MMP proporciona ganhos objetivos e subjetivos (apresentados no Apêndice F), ao preservar a borda de mapas de profundidade codificados.



(a) Máscara usada para detectar bordas dos objetos da vista 10 da sequência Book Arrival, Frame 0



(b) Vista 9 reconstruída da sequência Book Arrival, Frame 0

Figura 5.4: Resultados do algoritmo MMP *edge-aware* para codificação de mapas de profundidade.

MMP original. Isso indica que o algoritmo original, uma vez que lhe seja garantido taxa suficiente, é capaz de preservar as bordas dos objetos com alta fidelidae, uma característica desejada para codificação de mapas de disparidade.

5.3 Usando o MMP para codificar conjuntamente textura e mapas de profundidade

Como vimos nas seções anteriores, o MMP é muito eficiente para a codificação dos mapas de profundidade. Além da síntese, os mapas de profundidade podem também ser usados para descorrelacionar múltiplas vistas na etapa de codificação da textura. Dessa forma, é realizada uma codificação conjunta de textura e profundidade. Algumas propostas [86, 87] foram feitas para o uso de vistas projetadas com o auxílio dos mapas de profundidade como predição para a codificação de imagens multivistas. Aqui também vamos propor uma solução conjunta de codificação, porém baseada no algoritmo MMP.

5.3.1 Alocação ótima de bits entre textura e profundidade

De forma semelhante à abordagem realizada pelo grupo MPEG, determinamos a relação ótima para o algoritmo MMP entre a taxa reservada para a codificação da textura e dos mapas de profundidade. A nossa análise foi feita codificando as vistas em separado e o resultado é uma relação entre os fatores λ do algoritmo de otimização da textura e da profundidade. Essa relação ótima é então usada para avaliar o desempenho do codificador de multivistas e respectivas profundidades, bem como compará-lo com outras propostas, como a referência do MPEG baseada no padrão H.264/MVC. Os resultados se encontram no Apêndice G.

5.3.2 Codificação conjunta de vistas e mapas de disparidade

A proposta de codificação conjunta das múltiplas texturas e respectivos mapas de disparidade baseada no MMP é descrita da seguinte maneira. Uma única vista e respectivo mapa de profundidade são codificados com o λ determinado pela alocação ótima de bits. Em seguida, com o auxílio dos parâmetros das câmeras, a vista de referência é projetada para a posição da vista auxiliar, e será usada como predição para a codificação da textura da vista auxiliar. Note-se que esta vista virtual poderá não resultar no resíduo mínimo, uma vez que diversos artefatos provocados por áreas de oclusão, diferença de iluminação, entre outros contribuem para o descasamento entre a vista projetada e a vista original.

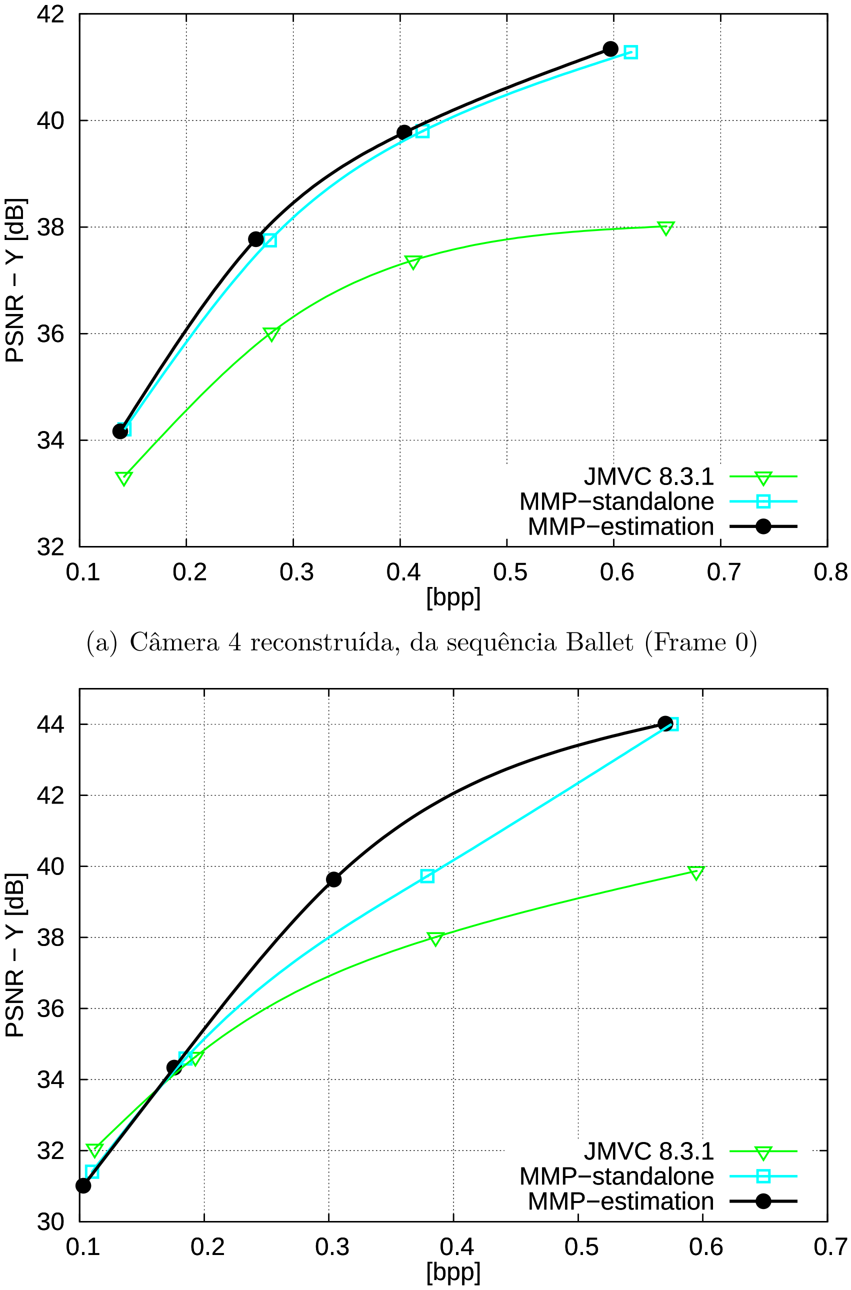
Para lidar com os problemas da síntese, usamos a vista projetada como mais um possível modo de predição disponível ao MMP. Portanto o MMP é capaz de alternar entre a vista projetada, ou os modos de predição intra já utilizados, e assim contornar os problemas da áreas de oclusão ou erros de síntese. A Figura 5.5 mostra o desempenho do algoritmo proposto de codificação conjunta baseado no MMP. O MMP supera em termos de qualidade objetiva o algoritmo usado como referência pelo MPEG, o JMVC, em até 4 dB.

5.4 Conclusões

O novo formato de imagens 3D apresenta muitos desafios. Entre eles, o uso do algoritmo de síntese para a geração de vistas virtuais, que apresenta diversos problemas por resolver, e ainda deverá ser aperfeiçoado. O uso dos mapas de profundidade proporciona uma independência entre produção e visualização de conteúdos 3D, sendo muito eficiente no que diz respeito à taxa necessária para a sua codificação. No entanto, a sua característica particular de mistura de áreas com altas e baixas frequências cria um desafio para os codificadores atuais.

Aqui apresentamos o MMP como uma alternativa para a codificação dos mapas de profundidade e propomos uma solução híbrida, usando o H.264/MVC para a codificação das vistas e o MMP para a codificação dos mapas. Devido à sua propriedade de preservação das bordas, o MMP se mostrou particularmente eficaz nesse cenário, permitindo um aumento no desempenho taxa-distorção das vistas sintetizadas, nomeadamente por diminuir artefatos nas bordas dos objetos. Além de mostrar a eficácia do MMP, o resultado ressalta a importância da preservação das bordas nos mapas de profundidade.

Como o MMP é flexível o suficiente para codificar textura e profundidade com o mesmo algoritmo, e em diversas taxas desejadas, propomos uma solução inteiramente baseada no MMP. Uma alocação ótima de taxa entre a textura e a profundidade foi obtida e, em seguida, codificamos uma vista de referência e seu respectivo mapa de profundidade. A vista então foi projetada com o auxílio de um algoritmo de síntese e adicionada ao *loop* de predição como mais um possível modo. Desta maneira foi possível melhorar o desempenho do MMP para a codificação de múltiplas vistas, possibilitando a descorrelação entre as vistas através da síntese. Os resultados apresentados mostram o eficiente desempenho taxa-distorção do algoritmo proposto, gerando vistas reconstruídas superiores à vistas geradas com o codificador de referência H.264/MVC. A extensão dos resultados de vistas para sequências de vídeo será um tópico de trabalho futuro.



(a) Câmera 4 reconstruída, da sequência Ballet (Frame 0)

(b) Câmera 38 reconstruída, da sequência Champagne Tower (Frame 0)

Figura 5.5: Comparação entre o desempenho do codificador conjunto multivistas e respectivas profundidades, baseado em MMP e baseado no padrão H.264/MVC, usado como referência pelo MPEG.

Capítulo 6

Conclusão

6.1 Conclusões discutidas por contribuição

6.1.1 Codificador de imagens com casamento de padrões multiescala

As técnicas recentes introduzidas ao MMP permitiram que o algoritmo atingisse um desempenho taxa-distorção estado-da-arte para diversos tipos de imagens. No entanto, mostra-se no Apêndice B que as novas técnicas que permitiram melhorar o desempenho do MMP acarretaram, em sua grande maioria, num aumento significativo da complexidade computacional. No Capítulo 2 propusemos o uso de um critério de decisão para a escolha do modo de predição que fosse baseado apenas na energia do bloco residual, e não no seu custo de codificação. Desta maneira, foi possível reduzir em até 80% o tempo de codificação de uma imagem. No entanto, perdas de até 0,25 dB para imagens suaves, ou ainda maiores para imagens compostas, foram reportadas. Mesmo assim, o desempenho taxa-distorção do MMP ainda ficou superior ao desempenho de outros codificadores, como o H.264/AVC Intra e o JPEG2000.

O problema da complexidade computacional para algoritmos que utilizam o paradigma de casamento de padrões é recorrente, e sempre deve-se buscar um compromisso entre o desempenho taxa-distorção e os ganhos computacionais. No entanto, com o desenvolvimento de novos *hardwares* e placas gráficas, o problema da complexidade computacional perde cada vez mais a sua importância. Esses novos equipamentos possuem uma elevada capacidade de processamento, e estão prontos para processar grandes quantidades de informação. Para tal, usam múltiplos processadores ou ainda placas dedicadas, baseadas também em computação paralela, como as GPU's (*Graphic Processing Unit*). Um grande desafio para o MMP neste momento é verificar as rotinas passíveis de paralelização e adaptar o algoritmo para processamento paralelo.

6.1.2 MMP utilizando predição com critério de mínimos quadrados

Com o intuito de flexibilizar ainda mais o passo de predição, adicionamos um modo de predição baseado no critério dos mínimos quadrados. Este modo de predição foi primeiramente concebido para compressão sem perdas, e no Capítulo 3 uma adaptação não-trivial do modo para predição de blocos no codificador MMP com perdas foi proposta. O novo modo é adaptativo e, diferentemente dos modos de predição originais usados pelo MMP, é capaz de se adaptar a bordas de objetos com qualquer direção e conseguir uma predição de elementos da imagem com maior fidelidade. Dessa forma, mostramos que o novo modo de predição é eficaz para imagens com conteúdo de alta frequência e também é um dos modos mais usados por imagens com este tipo de conteúdo. No caso de imagens compostas, onde a predição não consegue modelar corretamente o comportamento dos objetos, a adição de mais um modo de predição não veio afetar o desempenho taxa-distorção. Portanto, o novo modo de predição adotado pelo MMP foi eficaz em aumentar o desempenho taxa-distorção para todos os tipos de imagens, com ganhos de até 1 dB para imagens com muito conteúdo de alta frequência.

O método, no entanto, acarreta num aumento da complexidade computacional do algoritmo, não só por testarmos mais um modo de predição como pelo fato do treinamento exigir uma inversão de matrizes. Técnicas de redução da complexidade do treinamento são interessantes tópicos para trabalhos futuros.

6.1.3 Compressão sem perdas usando o MMP

O uso do modo de predição baseado no critério dos mínimos quadrados foi o fator motivante para usarmos o MMP para codificação de imagens sem perdas. Apesar do MMP ser naturalmente capaz de codificar sem perdas, para tal tendo apenas que ter um dicionário 1×1 completo, o seu desempenho para codificação de imagens sem perdas nunca havia sido analisado.

No Apêndice D foi derivado pela primeira vez um limite teórico sobre o desempenho do MMP como um codificador sem perdas. Foi mostrado que o MMP é capaz de atingir a entropia de fontes sem memória, ergódicas, geradas a partir de um alfabeto finito. Um ponto interessante da dedução é a evidência do passo de predição, o que nos levou a pesquisar soluções para melhorar ainda mais os passos de predição, quando usamos o MMP para codificação sem perdas.

No Capítulo 4, adaptamos para o MMP diversas técnicas comumente usadas em codificadores sem perdas, como alterações no modo de predição para usarmos sempre o vizinho mais próximo e o uso de contexto para a codificação do resíduo de predição. Com os diversos testes realizados, mostrou-se que o mais eficaz é otimizar o passo de predição, uma vez que a codificação do resíduo por casamento de padrões multiescala usando um dicionário adaptativo já era eficiente para a codificação do resíduo. Em seguida, o desempenho do algoritmo MMP de codificação sem perdas com as sugestões de melhoria da predição foi comparado ao desempenho de diversos codificadores considerados estado-da-arte para a codificação sem perdas. Mostrou-se que, no caso das imagens suaves, o MMP tem uma maior capacidade de compressão do que a de codificadores baseados em casamento de padrões, como o formato PNG, devido ao seu eficiente passo de predição. Já no caso de imagens compostas, onde o passo de predição não é eficaz, a otimização da predição em conjunto com a codificação do resíduo é responsável pelo excelente desempenho do MMP.

6.1.4 Compressão multivistas usando o MMP

No Capítulo 5 mostramos que o MMP é uma eficiente ferramenta para a codificação de mapas de profundidade, devido à sua característica de preservação das bordas dos objetos e elementos de alta frequência. Dessa forma, imagens sintetizadas com mapas de profundidade codificados com o MMP apresentaram menos artefatos, sendo objetivamente e subjetivamente superiores, no sentido taxa-distorção, às imagens onde o mapa de profundidade foi codificado com algoritmos baseados em transformadas, como o H.264/AVC Intra ou ainda o H.264/MVC. No entanto, ainda o desempenho taxa-distorção com o MMP, especialmente em baixas taxas, ficou aquém do desejado.

Baseado nos resultados obtidos com o MMP e na importância da codificação das bordas para o algoritmo de síntese, propomos uma modificação no MMP para não permitir distorções nas regiões identificadas como regiões de borda dos elementos no plano frontal. Os resultados não foram satisfatórios, por necessitarem de muitos bits para a codificação das bordas, mas mostraram uma propriedade interessante do MMP, a capacidade de preservação das bordas para taxas médias e altas, sem a necessidade de alterar o algoritmo.

Devido à sua capacidade de codificar textura e profundidade em diversas taxas possíveis com o mesmo algoritmo, propusemos uma solução de codificação de vídeo 3D inteiramente baseada no MMP. Em primeiro lugar codificamos a textura e a profundidade em separado, e a alocação ótima de taxa entre a textura e a profundidade foi obtida. Em seguida propusemos uma arquitetura de codificação conjunta, onde codificamos uma vista de referência e o seu respectivo mapa de profundidade. Usando o algoritmo de síntese de vistas padrão do MPEG, a vista de referência é projetada para a posição da vista auxiliar e usada como mais uma possibilidade para o modo de predição. Desta maneira o MMP se tornou mais eficiente para a codificação de múltiplas vistas, explorando a correlação entre vistas para codificar a vista auxiliar. As vistas virtuais sintetizadas entre as vistas do par estéreo codificado mostram o desempenho superior do MMP, contendo menos artefatos e com melhor qualidade objetiva. Os resultados apresentados levaram em consideração apenas um único quadro das sequências testadas e o algoritmo proposto serve apenas para a codificação de múltiplas imagens, e não sequências de vídeo. Uma extensão natural desta linha de pesquisa é a incorporação do vídeo e a exploração da redundância temporal em conjunto com a redundância entre vistas.

6.2 Perspectivas futuras para o MMP e tópicos em aberto

O MMP representa um paradigma alternativo de codificação e com constantes evoluções, o algoritmo foi capaz de atingir um desempenho taxa-distorção ao nível do estado-da-arte para diversos tipos de imagens em diversas taxas possíveis. Porém, será que é mesmo necessário pesquisarmos um novo codificador de imagens ou simplesmente deveríamos continuar a usar os algoritmos baseados em transformadas?

Com a evolução tecnológica e a ampla divulgação de imagens e vídeo digitais, o usuário passou a ter cada vez mais acesso a conteúdos codificados e, com o auxílio de terminais com alta capacidade e desempenho, passou também a ser crítico e a exigir mais qualidade. Porque precisamos de mais informação, mais imagens, em qualquer hora e em qualquer lugar, os codificadores de imagem e vídeo têm também que se atualizar. Porém, o que antes parecia ser suficiente, hoje em dia poderá não servir mais, e inclusive comprometer a funcionalidade de um sistema. O novo formato 3D é um exemplo desta abordagem. Artefatos conhecidos, gerados por codificadores baseados em transformada, que antes passavam desapercebidos, ao serem sujeitos a um algoritmo de síntese podem gerar vistas com uma qualidade inaceitável.

Portanto, o estudo de técnicas alternativas de codificação tem como objetivo ver o problema através de outros olhos, e dessa forma poder contribuir, seja na proposta de novas técnicas, ou até mesmo para a melhorias das técnicas vigentes. Os resultados apresentados nesta tese são um bom exemplo de como podemos encarar um problema usando outro ponto de vista, e de como isso pode nos levar a contribuir com novas técnicas ou exaltar características distintas de um novo tipo de imagem.

No entanto, a complexidade computacional do MMP pode ser uma barreira para o seu uso generalizado. Com as máquinas e técnicas atuais, o MMP ainda apresenta um desempenho computacional aquém do desejado. Porém, é certo também que as máquinas estão mais velozes e que cada vez mais temos disponível *hardware* com mais capacidade de memória e de processamento. Dessa forma, é crucial que sejam desenvolvidas técnicas para acelar o MMP, e um tópico em aberto é o desenvolvimento de *hardware* específico para algoritmos de casamento de padrões, como o MMP.

Por fim, acreditamos que o MMP poderá ser uma alternativa viável aos codificadores baseados em transformada, para a codificação de novas imagens, que misturam conteúdos como textura, objetos criados computacionalmente, textos informativos, informação de profundidade, etc. A mistura de diversos tipos de conteúdo com características distintas é um grande desafio para codificadores baseados em transformada, mas pode ser facilmente codificada com o MMP, que não assume qualquer característica da fonte que está codificando.

Apêndice A

Introduction

A.1 Image compression using multiscale recurrent pattern matching

In the last couple of years, the image coding algorithm based on multiscale recurrent pattern matching, the MMP (*Multidimensional Multiscale Parser*), has been the main subject of several research projects [13–16]. MMP's universal characteristic has enabled its use in several different areas, from lossy compression of compound images [17], smooth images [18], stereoscopic images [19] to ECG signals [20].

Regarding smooth image compression, MMP represents an alternative to the well-accepted image coding paradigm: the use of transform-quantization-coding. MMP innovative approach replaces the three mentioned operations, and opens the path for a new way of thinking image compression. Some contributions have already been done for smooth image compression using MMP [21, 22], which provided MMP with a state-of-the-art rate-distortion performance, outperforming encoders such as JPEG2000 [6] and H.264/AVC Intra [23].

A.2 Motivations

The latest developments of the MMP algorithm [21, 22] have come at a cost of increased encoding time. MMP's computational complexity has always been a concern [21, 45], and with the development of any new technique that is incorporated to the MMP algorithm, efforts to take the computational cost into consideration should also be made.

Despite the increased complexity, rate-distortion gains were reported when a more flexible segmentation technique [22] was introduced in the coding algorithm. Therefore, adaptive coding techniques have the potential to increase MMP's rate-distortion efficiency. For example, methods for relaxing the rigid prediction direction are able to provide rate-distortion gains and should be investigated.

The evolution of computational power allows the use of more sophisticated techniques for image compression. Also the wide availability and low-prices of storage, as well as the emergence of high resolution 3D displays creates new demands for image compression algorithms. What was previously accepted or even ignored, such as coding artifacts caused by quantization, may now impair the acceptance of new imaging systems.

The paradigm upon which MMP is based can provide new insights regarding image compression, and has the potential to achieve significant improvements in this area. With novel computational resources and new coding challenges, MMP might become a viable solution for image compression.

The aim of this thesis is to use MMP as the selected algorithm for image compression, and contribute with proposals for reducing the algorithm's complexity, enhancing its rate-distortion performance and also evaluating its use in new areas, such as lossless compression and 3D multiview images.

A.3 Thesis outline

This thesis is organized as follows. Chapters 1 through 6 are written in Portuguese, and provide an overview of the work done in the scope of this PhD. Thesis. All the appendices are written in English, and provide comprehensive details about the developed work.

Appendix B starts with a review of the MMP algorithm and its evolutions. An analysis of the algorithm coding cost is formally given, and a proposal for accelerating the encoding process along with the obtained results are presented.

The theory of least-squares predictors is presented in Appendix C. A new prediction mode based on least-square minimization is proposed for the MMP framework. Details on LSP block implementation and how this new prediction mode was incorporated into MMP are also given.

Appendix D deals with the usage of MMP as a lossless image encoder. For the first time, a performance bound for MMP lossless compression is derived, showing that MMP can asymptotically achieve the entropy rate of any stationary, ergodic, memoryless source with finite alphabet, and providing insights on how to enhance MMP lossless capability. Proposals for enhancing MMP's prediction and residue coding are presented, and the results are compared with selected state-of-the-art lossless image encoders.

Due to the novelty of coding 3D images with corresponding depth maps, Appendix E provides an overview of the technologies present in the 3D chain of operation. The view synthesis process is detailed and the standardization activity on 3D is also reviewed.

Appendix F focus on coding the depth information of multiview images. It presents the results of using MMP for coding depth maps, discussing metrics for depth map evaluation. A proposal for mitigating rendering artifacts by improving coding of depth map edges is also given.

Then, in Appendix G, the independent coding of texture and depth map using the MMP is investigated. The optimal bitrate allocation between texture and depth maps is found, and the advantages of using MMP for coding mutliview+depth images are discussed. Also, an architecture of a 3D encoder based on MMP that jointly encodes texture and depth is proposed, analyzing the advantage of using the warped frame as prediction.

Finally, Appendix H concludes this work.

A.4 Thesis contributions

A.4.1 Lossy image coding

One of the first contributions of this thesis regards lossy image coding. A method that reduced MMP's encoding time by 7 times, modifying the criteria for rate-distortion optimization was proposed, and then published in [25]. Next, a non-trivial adaptation for block prediction using least-squares optimization criteria was performed. The new prediction mode was then incorporated into the MMP prediction loop, increasing its rate-distortion performance. These results were published in [26] and [27].

A.4.2 Lossless image Coding

Since results for MMP acting as a lossless image encoder were never reported, this area was also investigated in this thesis. A performance bound for lossless compression was derived, showing that MMP algorithm can asymptotically achieve the entropy rate of any stationary, ergodic, memoryless source with a finite alphabet. This demonstration provided evidence on how to increase MMP's lossless compression performance, and some techniques were incorporated into MMP. The results of this research were submitted for publication in [28].

A.4.3 3D image coding

Due to MMP's flexibility, it was also applied for coding depth/disparity maps of multiview images, and proved to be very effective because of its high frequency preservation property. The first encoding results were published in [29]. Then in [30], a well-known DIBR (Depth Image Based Rendering) algorithm was used to evaluate virtual views synthesized with uncoded texture data and coded depth data using MMP. Objective and subjective results have shown the superiority of the algorithm. It produced synthesized views with less artifacts, especially at object's boundaries. A technique to improve the coding of depth maps by differentially coding object's edges was then proposed. However, coding results showed that the best option is to perform an optimal bitrate allocation between texture and depth maps, and to just use MMP without any modification. In light of these results, we proposed an architecture for multiview coding using depth maps coded with MMP and textures coded with MMP using the warped view as prediction. Results showed that MMP can outperform multiview encoders such as JMVC, also producing synthesized views with better quality than other compared methods. The outcome of this research is being prepared for publication [88].

Apêndice B

Multidimensional multiscale parser algorithm

SUMMARY: This appendix starts with a review of the MMP algorithm and its evolutions in Sections B.1 and B.2. In Section B.3, an analysis of the algorithm coding cost is formally given. A proposal for accelerating the encoding process along with the obtained results is presented in Section B.4. Section B.5 concludes this appendix discussing future perspectives for MMP-based algorithms.

B.1 Multiscale recurrent pattern matching

Instead of the traditional transform-quantize-encode procedure, MMP-based encoders exploit a paradigm based on dictionary search and recurrent pattern matching. This technique is suited for coding frequently recurring elements, where the list of most used elements is kept in a dictionary structure, $\mathcal{D} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_M\}$. Once a codeword $\mathbf{C}_i$ of the dictionary is found in the input signal, the dictionary index i is entropy encoded. Therefore, compression is achieved by attributing short length codes to the index of frequently occurring codewords. Elements that do not belong to the dictionary also do not have an associated index, so usually a escape symbol is used and the new pattern is coded in a less efficient manner. Examples of lossless compression using dictionary techniques are the Lempel-Ziv algorithms [31–41]. For lossy coding, several algorithms were proposed [17, 42–44], and some are known as lossy Lempel-Ziv algorithms. There is also an analogy between lossy pattern matching algorithms and vector quantization [89].

In the above mentioned algorithms, pattern matching is done between blocks of the same dimension. MMP proposes to waive this restriction by using blocks with different dimensions for the matching operation, a concept also exploited by fractal encoders [12]. A matching probability exists between blocks of different scale

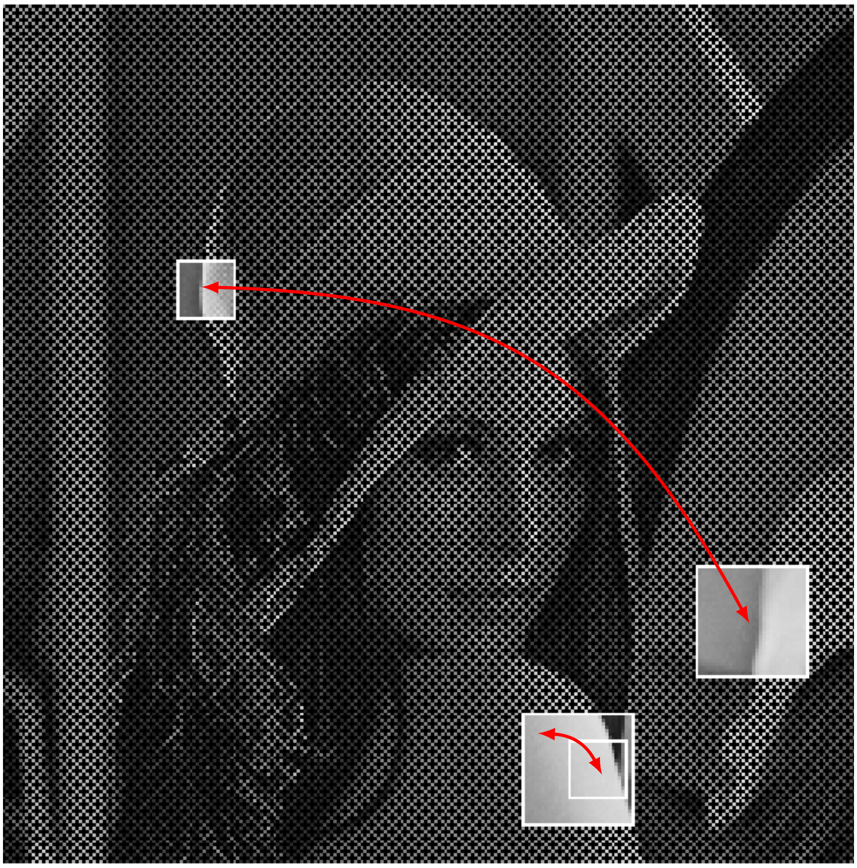


Figure B.1: Self similar portions of the Lena image, taken from [12]. Part of her shoulder is similar to the bigger overlapping block (notice the need for scale transformation), and portion of the reflection of her hat in the mirror is also similar to a smaller part of her hat.

due to the images' self-similarity property, that is, parts of the images that are approximately similar to each other, even at different scales. Figure B.1 highlights self-similar parts of the Lena image. The concept of multiscale pattern matching is illustrated in Figure B.2. Notice that a scale transformation is necessary, in order to do the matching between the input pattern and the codewords of the dictionary.

By using contractions and expansions of the dictionary codewords in pattern matching, we increase the probability of finding a match, due to the fact that more elements are tested. In [90], it is shown that multiscale pattern matching can outperform ordinary pattern matching, when used to lossy compress data from memoryless Gaussian sources. Nevertheless, the increased number of elements still needs to be efficiently encoded.

The MMP algorithm uses a block-based approach, dividing the image into non-overlapping blocks, usually of size 16 × 16. Blocks are encoded in raster scan order, and each block might be further segmented if no satisfactory match is found at that scale. Block segmentation is signalled by a segmentation flag, that has a function

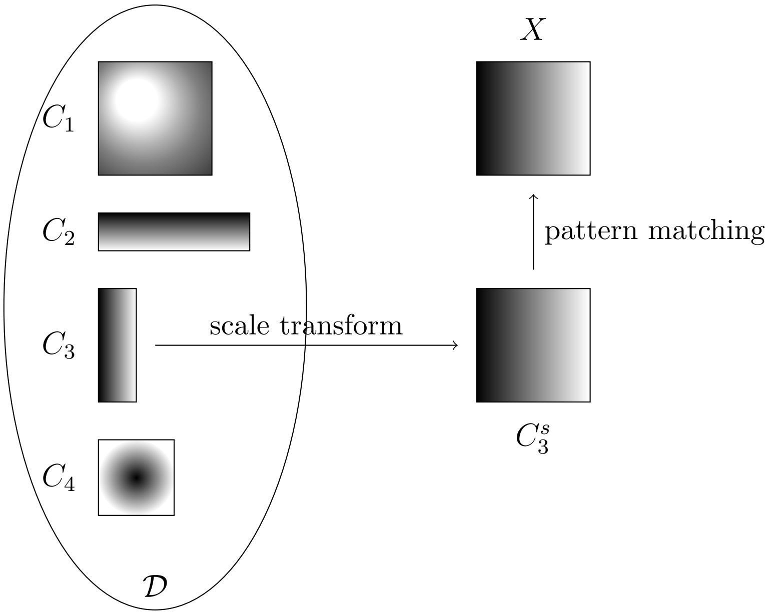
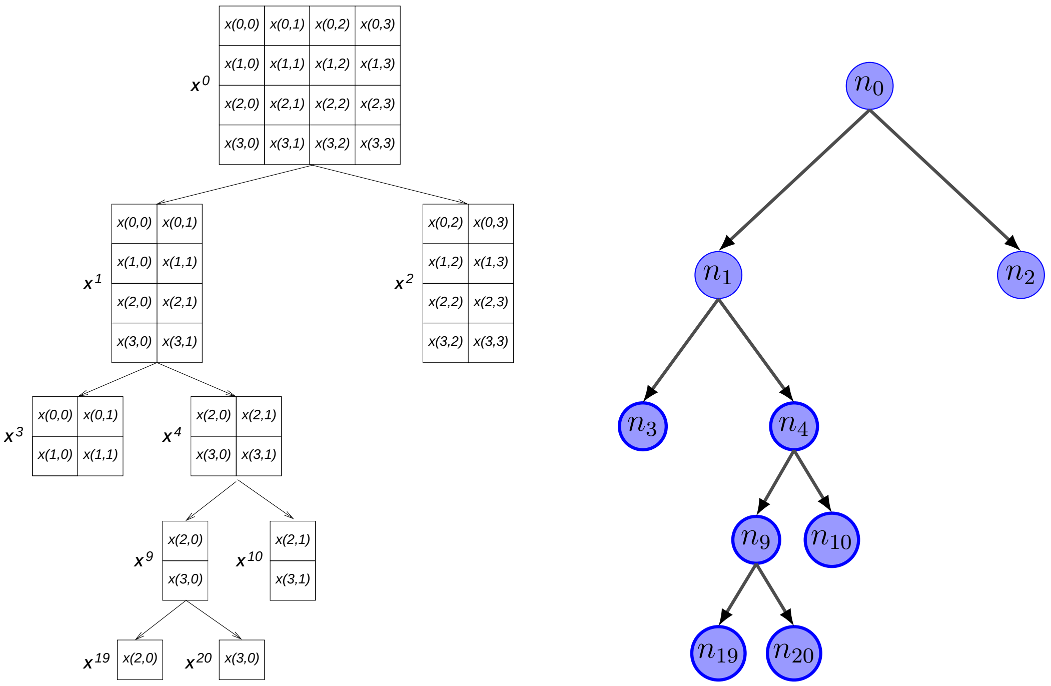


Figure B.2: Example of multiscale pattern matching. Notice that the use of scale transformation for each element C_i of the dictionary $\mathcal{D}$ enables the matching between the input block X and the corresponding codeword.

similar to an escape code. The input block will be divided, and the pattern matching operation can be recursively done for each segment. This operation goes on, until a satisfactory match is found for the segments, or the block can no longer be divided. In this case, the pixel value will be used for entropy coding.

In earlier versions, the block segmentation consisted in dividing the block in a quadtree structure [44]; later on, it was substituted by a dyadic segmentation [90], where the segments are divided in halves, alternatively in the vertical and horizontal directions. That is, a block of scale l, of dimensions $(2^{\lfloor \frac{l+1}{2} \rfloor}, 2^{\lfloor \frac{l}{2} \rfloor})$, is segmented into two blocks of dimensions $(2^{\lfloor \frac{l}{2} \rfloor}, 2^{\lfloor \frac{l-1}{2} \rfloor})$.

The resulting segmentation of the input block can be represented by a binary segmentation tree, where the nodes of the tree indicate a block segmentation, and the tree leaves represent the sub-blocks. On Figure B.3(a), an example of the dyadic block segmentation is depicted, and Figure B.3(b), shows its associated segmentation tree. The tree's nodes represent a block segmentation, and can be indicated on the bitstream by a '0' flag. The tree's leaves represent no segmentation of the block, and can also be signalized by a '1' flag followed by the dictionary index. The only exception is the 1 × 1 block, that cannot be further divided, and therefore it does not need any type of segmentation flag. For example, the MMP bitstream for the block presented in Figure B.3(a) would be `0,0,1,`i_3`,0,0,`i_{19}`,`i_{20}`,1,`i_{10}`,1,`i_2, where i_3,i_{19},i_{20},i_{10},i_2 are the indexes of the chosen codeword for each leaf node. Notice that no segmentation flag was sent for blocks 19 and 20, since their dimensions are 1×1, and they can no longer be segmented.



(a) Block Segmentation

(b) Segmentation Tree

Figure B.3: Example of dyadic block segmentation used by the MMP encoder and corresponding segmentation tree

Decoding the MMP bitstream is relatively simple. According to the received segmentation flags, the decoder will generate a segmentation tree similar to the one used by the encoder (and depicted in Figure B.3(b)). Once the segmentation tree is restored, the indexes are replaced with the respective codewords of the dictionary. Notice that there is no floating-point operation involved in the decoding process. Therefore, the decoder has low complexity and is appropriate for asymmetric applications.

The dictionary is then updated with the recently encoded patterns, inspired on the works of Lempel and Ziv [31], as shown in Figure B.4. Once the block is segmented, the dictionary indexes used to approximate each part of the block are concatenated and this new pattern is added to the dictionary. Also contractions and expansions of it are added to the dictionary, improving the dictionary adaptation. Note that by using recently encoded patterns, the dictionary adapts to the image statistics and no previous assumption is needed for its encoding.

The elements of the output bitstream are encoded using a context-based adaptive arithmetic encoder. The context used for the elements is the corresponding level in the segmentation tree, that is, the block's dimension. This information does not need to be explicitly sent to the decoder, since both encoder and decoder replicate the same segmentation structure. The use of the level information enhances the performance of the arithmetic encoder, since blocks of different sizes tend to have

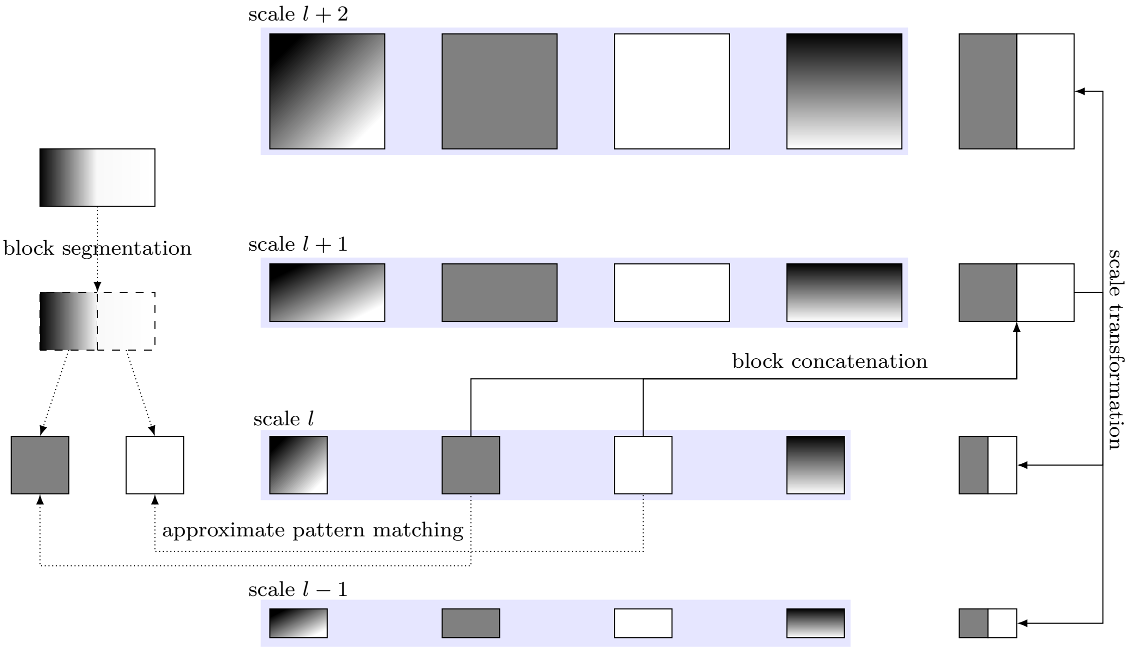


Figure B.4: The dictionary is updated with recently encoded patterns, that are concatenated and added also to different scales via scale transformation.

different probabilities of being segmented.

One way to help the adaptive arithmetic coder to reach the entropy of the dictionaries' indexes in a much faster way is to use the segmentation tree level of the blocks, that originated the codeword as a context for the adaptive arithmetic encoder [21]. This is similar to divide the dictionary in segments, where each segment only has elements that were formed by a block originally from a certain level $l_{\text{dic\_seg}}$. Each dictionary element is then encoded by this level information and the codeword index within this level. By separating the statistics of the codewords based on the level of origin, we allow a faster adaptation. This is so because the probabilities of each level of origin can be very different and thus the entropy for the dictionary segment may achieve values lower than $\lceil \log_2(N_P) \rceil$ (where N_P is the number of partitions) in the encoding of just few codewords.

In order to encode a block, one can choose many different combinations of segmentation flags and dictionary indexes. Coding decisions are made based on a Lagrangian cost optimization algorithm. One way to achieve a performance point on the convex hull of the operational rate-distortion curve is to use the segmentation tree structure with associated Lagrangian cost, and perform a minimization procedure similar to the one described in [91].

For each node of the segmentation tree, we associate a distortion given by

$$D(\mathbf{X}^l, \mathbf{C}_i^l) = \sum_{x,y} (\mathbf{X}^l(x,y) - \mathbf{C}_i^l(x,y))^2 \tag{B.1}$$

where $\mathbf{X}^l$ is the segment of scale l of the original block and $\mathbf{C}_i^l$ is the codeword chosen to represent that segment. The associated rate is given by

$$R(\mathbf{C}_i^l) = -\log_2(\Pr(i|\text{level})) \tag{B.2}$$

and it is dependent of the probability of the i-th codeword, $\Pr(i|\text{level})$, conditioned to the knowledge of the block level. The Lagrangian cost of a node n^l is then given by

$$J(n^l) = D(X^l, \mathbf{C}_i^l) + \lambda R(\mathbf{C}_i^l). \tag{B.3}$$

According to the node scale l, it can be further segmented, so the Lagrangian cost of each branch of this node (n_1^{l-1}, n_2^{l-1}) will be calculated. The tree will be segmented at the node if the sum of nodes' costs added to λ times the rate spent to send the segmentation flag is lower than the original Lagrangian cost, that is,

$$J(n^l) > J(n_1^{l-1}) + J(n_2^{l-1}) + \lambda R_{seg} \tag{B.4}$$

where R_{seg} is the rate necessary for transmitting the segmentation flag.

In this sense, the optimization function will actually test the full segmentation of the block and prune the nodes until the minimum cost in an RD sense is found [90]. This greedy procedure finds the minimum coding cost if the blocks are independent. However, due to the dictionary updating procedure, the encoding choices of a block affects the Lagrangian cost of future blocks. In [92], a rate-distortion optimization proposal was made, referred to as MMP-RDI, where the dictionary update procedure is taken into consideration when evaluating the coding cost. To this end, a “draft” dictionary was conceived, in order to estimate the rate necessary to send a codeword taking into consideration the new inserted codewords obtained from coding previous segments of the block. Marginal gains were reported at the cost of a higher computational complexity. More information about the MMP algorithm and considerations on its RD optimization can also be found in [90].

One of the main advantage of MMP encoders is their “universal" character. MMP can be efficiently used for a wide range of data signals, from voice and ECG [20, 45, 46] to stereoscopic images [19], still images [21, 90] and video [47–50].

In [17], it is presented results for MMP encoding of digital images that included images resulted from scanned documents or from computer graphics, usually presenting text and graphics (also referred to as compound images). For this sort of input, the MMP algorithm presented better results than most state-of-the-art encoders, such as JPEG2000 [6] or H.264/AVC Intra mode [51, 52]. Nevertheless, a gap still existed for smooth images. Some proposals aiming to bridge this gap have been proposed. In [93] and [53], a continuity criteria between blocks is included

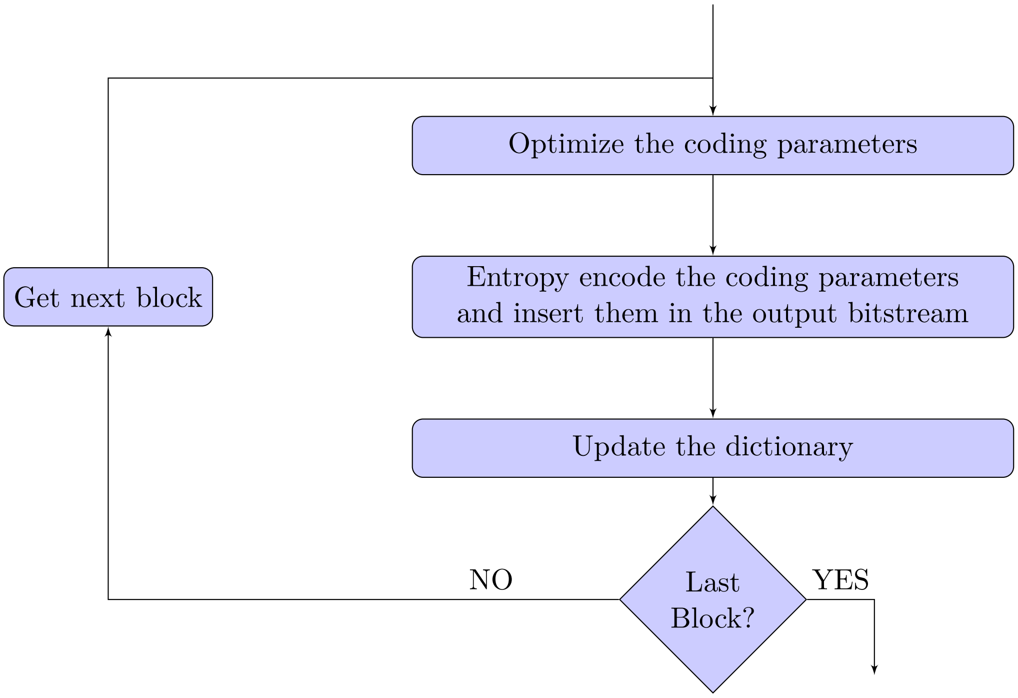

Optimize the coding parameters
Entropy encode the coding parameters and insert them in the output bitstream
Update the dictionary
Last Block?
NO
YES
Get next block

Figure B.5: Brief description of the MMP algorithm in pseudo-code.

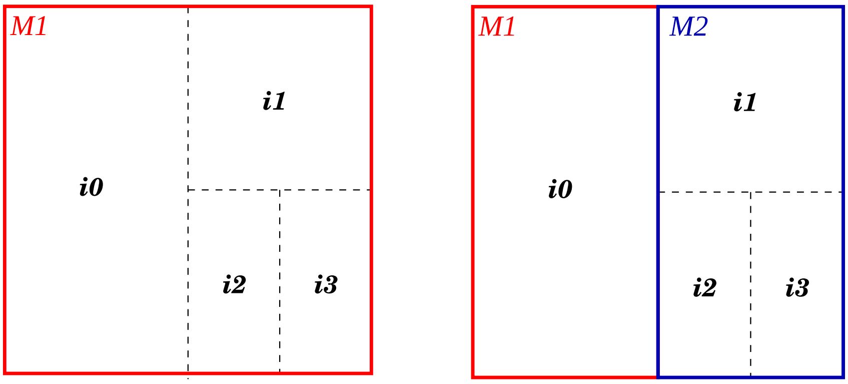
in the pattern matching procedure, giving rise to the MMP-APM (*Adaptive Probability Model*, [53, 93]) algorithm. It improved MMP's rate-distortion performance for smooth image coding, but still did not reach the rate-distortion performance presented by the state-of-the-art image encoder H.264/AVC Intra. Prediction was then added to the MMP framework in [54], enhancing its results for smooth images, and providing state-of-the-art rate-distortion performance.

A general pseudo-code of the MMP algorithm is provided in Figure B.5. A more detailed description can be found in Appendix I. This technique will be the base of all algorithms developed in this thesis, so it will be explained with further details in the next section.

B.2 MMP-Intra

Prediction was first used in an MMP framework in [94]. Based on the intra-prediction used in the H.264/AVC standard [51], the first prediction-based algorithm, the PC-MMP algorithm, used block prediction as the first encoding step, followed by the residual block being coded with a regular MMP algorithm. For this algorithm, prediction was applied only for a fixed blocks size of 16 × 16 (see Figure B.6(a)).

MMP-Intra or MMP-I [15, 21, 54], was the next prediction-based proposal. Similar to the H.264/AVC standard [8], it used adaptive block size prediction, with blocks



(a) Fixed Block Size Prediction (b) Hierarchical Prediction

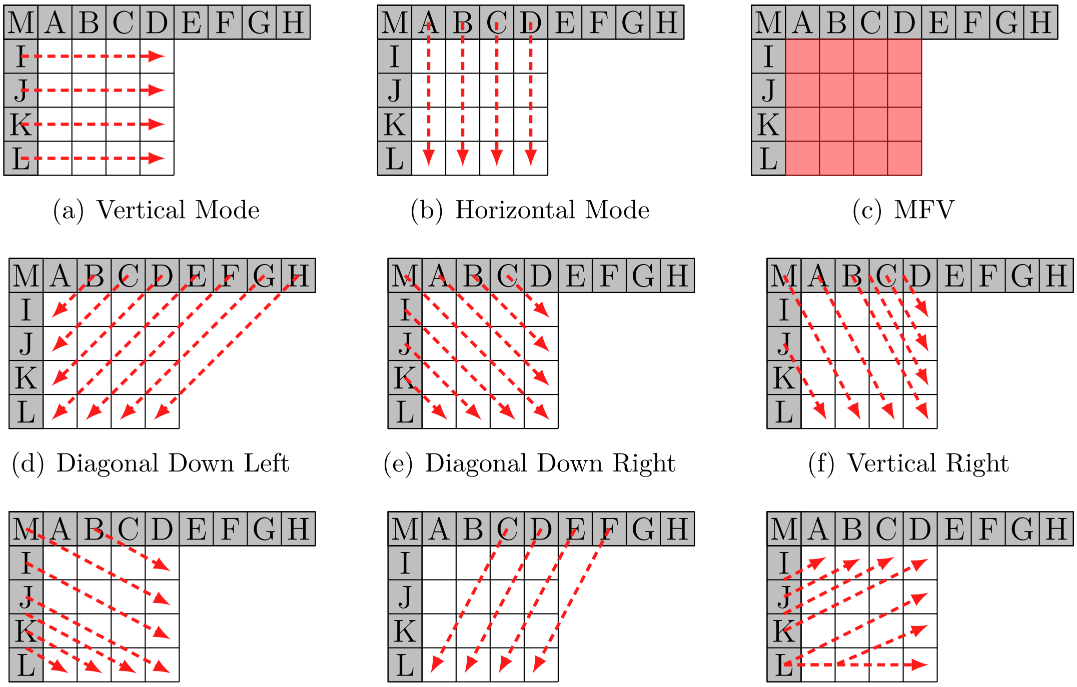
Figure B.6: Here we see two possible prediction usages. Either the whole block is predicted with one prediction mode (Fixed Block Size), or hierarchical prediction is applied (where the block and the prediction are both segmented)

of dimensions 16×16 down to 4×4 (see Figure B.6(b)). In order to implement this segmentation, the segmentation could assume three distinct values:

- The whole block is segmented (prediction and residue);
- Only the residue block is segmented, the prediction is not segmented. Notice that if the prediction mode hasn't been sent before, it will be sent after this flag;
- No block segmentation. Analogous to the previous case, if the prediction mode has not been sent yet, it will be inserted in the bitstream after this flag. In this case, also the codeword index will be sent. It is important to notice that a block that is not further segmented should be entirely contained in a prediction window, and a prediction window may contain more than one block.

The prediction modes used by the MMP-I algorithm are shown in Figure B.7. Due to the scanning order of the MMP algorithm's segment coding (top to bottom and left to right), the left and upper block neighbors are most of the times available. However, for modes that use the up-right neighborhood (for example, the diagonal mode), pixels not available are substituted by the closest available neighbor. MMP-I also uses the most frequent value (MFV) among neighboring pixels for the prediction, instead of the DC mode [21]. The MFV mode will choose one value that already exists in the picture's histogram, instead of using an average value that might not even exist in the image. Experiments have shown that for text and graphic images, the use of the MFV enhances the overall coding efficiency, and for smooth images this alternative mode has no effect on the performance [21].

When prediction is successful, it generates residue signals that have a distribution function concentrated in a reduced set of values around zero. The residue

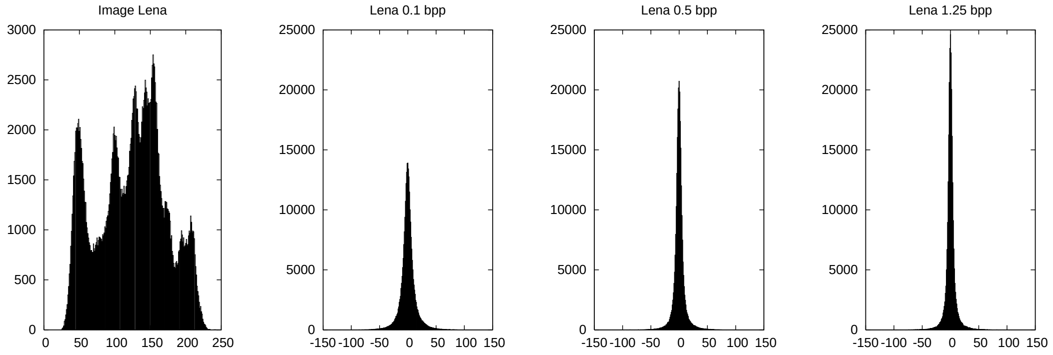


(a) Vertical Mode (b) Horizontal Mode (c) MFV

(d) Diagonal Down Left (e) Diagonal Down Right (f) Vertical Right

(g) Horizontal Down (h) Vertical Left (i) Horizontal Up

Figure B.7: Prediction modes used in MMP-INTRA algorithm.



(a) (b) (c) (d)

Figure B.8: Histogram plot for: a) original Lena Image; and for the prediction residues obtained from coding the Lena image with MMP-Intra with the following compression rates: b) 0.1 bpp; c) 0.5 bpp e d) 1.25 bpp.

statistical distribution is similar to a Generalized Gaussian distribution, and as it was demonstrated in [15], this distribution is favorable for the MMP residue encoding. This is so because more regular patterns are generated and the probability of using patterns already present in the dictionary increases. Figure B.8 shows the statistical distribution of Lena's residue coding. Notice that the use of prediction concentrates the values around zero, as desired. Also, as the bitrate increases, more modes are used and the prediction is enhanced, resulting in a better decorrelation, noticeable in the high-peaked distributions at higher bitrates.

The prediction mode is chosen based on the Lagrangian R-D cost function that determines the best trade-off between the prediction accuracy and the additional overhead introduced by the prediction data. For each prediction mode, the residue is encoded with the usual MMP encoder, and the cost of encoding that specific residue is added with the cost for sending the prediction mode is evaluated for all available modes, choosing the mode with the smallest cost.

Notice that the prediction step dramatically increases MMP's computational complexity, since the dictionary search has to be made for each mode, in order to determine the residue coding cost. A more detailed analysis on the prediction complexity burden, and also a proposal for reducing this computational cost, will be done in Section B.3.

B.2.1 MMP-II

A key factor for the performance of a pattern matching algorithm is the dictionary adaptation process. For a given stationary ergodic source, the performance of a pattern matching algorithm approaches the entropy of the source, given an infinite sequence [61]. But, for image encoding, we are far away from these conditions, so we have to employ techniques for a faster dictionary adaptation. A study on several techniques to enhance dictionary adaptation, and their use on MMP was made in [21]. The following techniques were introduced in the MMP-I algorithm, leading to the creation of the MMP-II algorithm [15, 21, 95].

Dictionary Redundancy Control: The introduction of new patterns will increase the dictionary approximation strength, but may also increase the number of bits necessary to encode an index of the dictionary. An approach to control the insertion of new elements in the dictionary is based on a distortion level. Once a codeword is present in the dictionary, dependent on the target distortion level for encoding, there is no need to insert any more new patterns similar to those patterns represented by that codeword, since in a rate-distortion sense the codeword on the dictionary is already a good enough approximation for the block to be encoded. This concept is illustrated in Figure B.9. The relationship between the minimum distance between codewords in the dictionary (also named minimum radius distortion) and the target rate-distortion point was heuristically determined. This rule can be found in [21] and is reproduced here for completeness.

$$d(\lambda) = \begin{cases} 5, & \text{if} \quad \lambda \leq 15; \\ 10, & \text{if} \quad 15 < \lambda \leq 50; \\ 20, & \text{otherwise.} \end{cases} \tag{B.5}$$

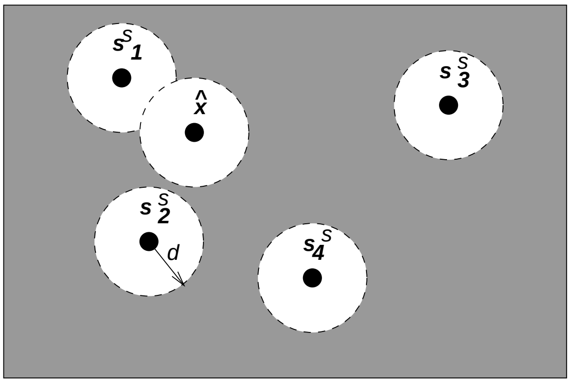


Figure B.9: Dictionary redundancy control technique. New patterns that fall into the multidimensional area defined by the parameter d will not be inserted into the dictionary.

Scale Restriction: Another technique to reduce the number of new elements that will be inserted in the dictionary is the restriction of scales, into which the new block will be introduced. Scale transformation between too far apart scales may destroy the blocks' structure, and be of no use for the image encoding. A parameter was added to the file header that indicates how many scales will be updated every time a new codeword is added to the dictionary.

Geometric transforms: In order to increase the dictionary adaptation, the encoder needs to introduce more useful patterns in the dictionary. One proposal for a faster dictionary growth is by adding not only the concatenated patterns and their scale-transformed relatives, but also to add blocks generated from geometric transforms, like rotated or symmetric blocks, and also displaced blocks. The use of this sort of blocks can be signalled in the header of the file, and improves lossy rate-distortion performance, as indicated in [19, 21].

Norm equalization of scaled blocks: In [21], a norm-equalization for upscaled transformed blocks was also proposed. Based on the theory of vector quantization for Generalized Gaussian sources [96, 97], the norm L^α of the residues modeled by a Generalized Gaussian of shape α will be constant. A norm equalization factor is used for scale transforms that increase the block dimension, such that the L^α norm of such blocks inserted in the dictionary is constant. The norm L^1 was used, as it resulted in the best compromise for all images and also had the advantage of being a simple implementation.

The techniques described in [21], not only enhance the rate-distortion performance, because of the reduction of the dictionaries indexes' entropy, but they also have a beneficial side-effect of reducing computational cost. Like Vector Quantization schemes, MMP's major computational burden rely on the dictionary search and any technique that reduces the number of elements in the dictionary will also reduce the encoding time. Moreover, some measures were taken in MMP-II to reduce the search time, such as early elimination of codewords by means of norm comparison [98].

In MMP's scheme, scale transforms add-up to the computational complexity, since scale transformation needs to be performed for each codeword comparison. In order to reduce encoding time, a multiple-codebook scheme was used, where several dictionaries with patterns from the same scale ($\mathcal{D}^{m,n}$) were stored instead of just one dictionary with patterns from different scales. By doing this, the transform operation is avoided during the dictionary search, being used only when adding a new pattern to the different dictionaries. Other operations can leverage on the same memory allocation concept, and use previously defined vectors in lookup tables to replace operations like logarithm and ssd (sum of squared differences). The use of lookup tables increases memory requirements; however these are tolerable in a standard personal computer (PC) based implementation. The initial block size was also taken into consideration. An analysis made in [21] showed that for low compression ratios, a significant reduction in computational cost was achieved, without any degradation in rate-distortion performance, by changing the initial block size from 16×16 to 8×8.

B.2.2 MMP-FP

One of MMP's main characteristic is its adaptive behavior. Nevertheless, MMP-II algorithm still have two rigid aspects in its encoding procedure: the prediction modes and the block segmentation. For the MMP-II encoder, a block is first segmented in the vertical direction, then in the horizontal direction. Results presented in [16] showed that rate-distortion gains could be achieved for some images, if the order of segmentation was inverted. So a new algorithm was conceived, where the segmentation direction is explicitly sent to the decoder, namely the MMP-FP algorithm (*Flexible Partition*, [22]).

The rigid dyadic block partitioning scheme previously used was relaxed, and now the block can be segmented in the vertical or horizontal directions, according to the best RD compromise. An additional flag is sent to indicate the direction of the segmentation. By doing this, the optimization algorithm now has to consider 25 possible segmentation patterns for a 16×16 block (eg. 16×1 or 2×8). Figure B.10 shows the possible block sizes.

The Lagrangian optimization algorithm was also modified in the MMP-FP, in order to choose the best segmentation option. The chosen segmentation orientation is the one that minimizes the Lagrangian cost (more details can be found in Appendix

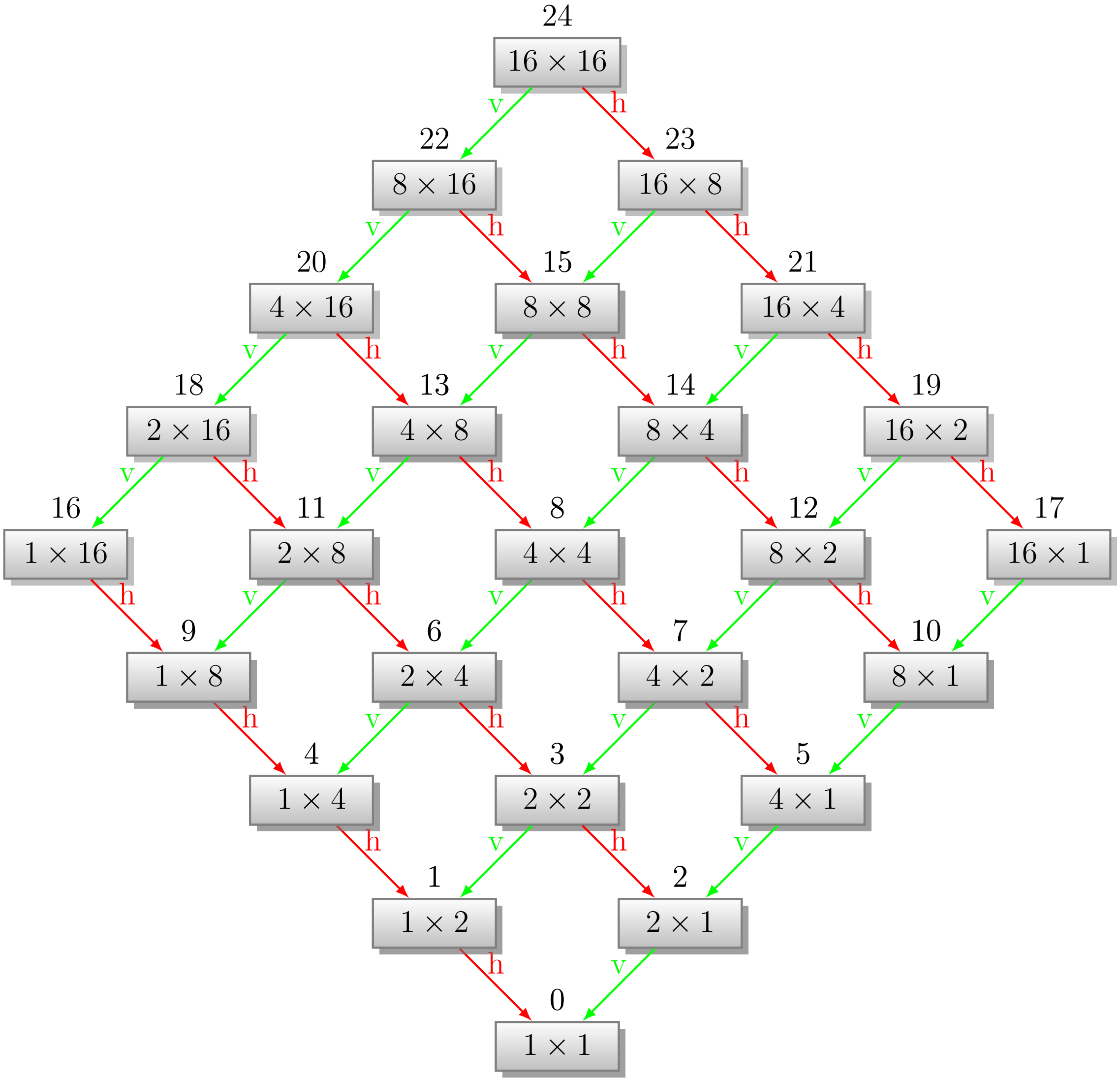
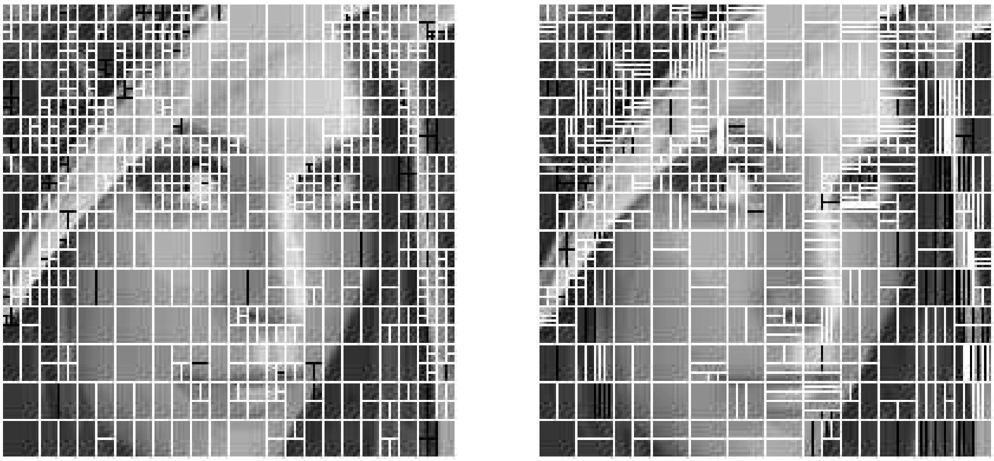


Figure B.10: Possible block sizes for the MMP-FP algorithm.

I). With more options for the block segmentation, computational complexity has also increased, because the algorithm must test several segmentation possibilities along with different prediction schemes for each block size.

Figure B.11 shows the effect of flexible segmentation. With the new segmentation scheme, thin blocks with a dominant orientation can be found, such as the horizontal blocks at Lena's lips. The new segmentation scheme allows prediction to choose thinner block segments. Neighbors that are closer to the position to be predicted are usually similar, due to the smoothness property of the image. By using thinner blocks, closer reconstructed values can be used for prediction, instead of far away pixels. By looking again at Lena's lips, we can notice that the horizontal orientation is then in accordance to the object's structure main orientation. Gains reported in [22] show the benefits of flexible segmentation, especially for middle to high rates.



(a) Original Segmentation

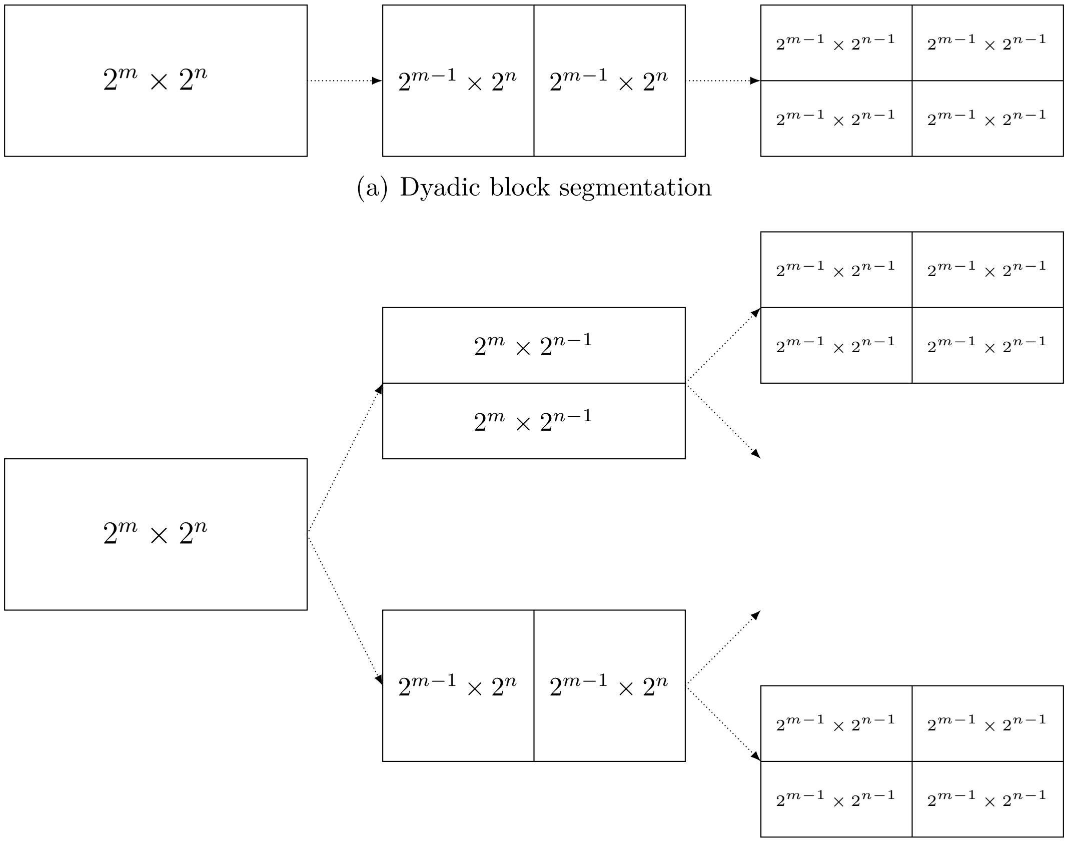
(b) Flexible Segmentation

Figure B.11: Flexible block segmentation for the Lena image. The block orientation is similar to the object's orientation in the image, such as the blocks around the lips and at Lena's hair.

B.3 MMP computational complexity analysis

Up to now, the focus of research on the MMP algorithm was improving the algorithm's rate-distortion performance, and little attention has been devoted to the algorithm's complexity. Since MMP-based encoders use an approximate pattern matching scheme, they have a complexity similar to that of standard Vector Quantization methods, that is traditionally higher than transform-based encoders. In addition, most of the improvements that lead to higher gains (new possible partitions, increased dictionary cardinality, etc. [21, 22]) also incurred in higher computational demands.

The biggest computational burden of the MMP algorithm relies on the estimation of the codeword index, similar to the search operation in a vector quantization operation. In [45], the number of multiplications required by the MMP algorithm was derived, based on a dyadic segmentation used by the R-D optimization algorithm. That is, when using dyadic segmentation, the full search VQ is performed $\log_2(N) + 1$ times for each block of size $N = 16 \times 16$. For the flexible segmentation scheme, the number of times the index search optimization is done is increased. Prediction also contributes to the repetition of the index search operation. Next, we determine the number of times the full search VQ operation is carried, in order to determine the computational complexity of the MMP-FP encoder.



(a) Dyadic block segmentation

(b) Flexible block segmentation

Figure B.12: Flexible block segmentation. The algorithm will inspect both segmentations possibilities, horizontal or vertical block division.

B.3.1 Formal derivation of MMP-FP's computational complexity

Here we will review the formal derivation for the computational complexity of the original MMP algorithm, as done in [45]. In a similar fashion, we will determine the computational complexity of the algorithm with flexible segmentation, and show the increase in computational complexity that incurs from adding the flexible segmentation. Still using the flexible segmentation, we will formally derive the computational complexity for the algorithm with hierarchical prediction, providing evidence on the increase of complexity achieved with the latest developments.

The computational complexity of an MMP-based algorithm required to find the best match is closely related to the number of searches the algorithm needs to perform. This operation is similar to a full search vector quantization, and its complexity is given by $(2^m \times 2^n)S$, where $(2^m, 2^n)$ is the block dimension, and S is the number of elements from the dictionary. For the MMP algorithm with dyadic block segmentation (see Figure B.12(a)), defining the computational complexity by the order of multiplications for coding one block, similar to [45], we obtain

$$\mathfrak{C}_{\mathrm{MMP}}(2^m, 2^n) = (2^m \times 2^n) \times S \times (m + n + 1) \tag{B.6}$$

The proof of equation (B.6) can be done by induction. The formula holds for blocks of size (1×1), since the elements of the dictionary will be tested only once, that is

$$\begin{aligned} \mathfrak{C}_{\mathrm{MMP}}(2^0, 2^0) &= (2^0 \times 2^0) \times S \times (0 + 0 + 1) \\ &= S \end{aligned} \tag{B.7}$$

Using the inductive hypothesis, the formula holds for blocks of dimension $(2^m, 2^n)$. Blocks of dimension $(2^{m+1}, 2^n)$ need to perform one vector search for the block without segmentation, then the block is divided horizontally. The total computational complexity of the block is the sum of the complexity for the block and its two offsprings after block segmentation. The complexity is then given by

$$\begin{aligned} \mathfrak{C}_{\mathrm{MMP}}(2^{m+1}, 2^n) &= (2^{m+1} \times 2^n) \times S + 2 \times \mathfrak{C}_{\mathrm{MMP}}(2^m, 2^n) \\ &= (2^{m+1} \times 2^n) \times S + 2 \times ((2^m \times 2^n) \times S \times (m + n + 1)) \\ &= (2^{m+1} \times 2^n) \times S + (2^{m+1} \times 2^n) \times S \times (m + n + 1) \\ &= (2^{m+1} \times 2^n) \times S \times (m + 1 + n + 1) \end{aligned} \tag{B.8}$$

The induction for the other coordinate is entirely analogous.

For the MMP algorithm with flexible block segmentation (MMP-FP, see Figure B.12(b)), both segmentations options are evaluated at each level, which increases the number of searches performed by the algorithm. The computational complexity for coding one block is then given by

$$\mathfrak{C}_{\mathrm{MMP-FP}}(2^m, 2^n) = \sum_{i=0}^{\max(m,n)} \sum_{j=0}^{i} \binom{i}{j} \times (2^m \times 2^n) \times S \times f(i, j) \tag{B.9}$$

where the function f is given by

$$f(i, j) \triangleq \begin{cases} 1 & \text{if } m - (i - j) \geq 0 \text{ and } n - j \geq 0, \\ 0 & \text{otherwise }, \end{cases} \tag{B.10}$$

Table B.1: Computational complexity for a block, given its size, for the MMP and the MMP-FP algorithms, considering only one element in the dictionary.

Block Size	1×1	2×1	1×2	2×2	4×1	1×4	4×2	2×4	4×4
MMP	1	2	NA	3	NA	NA	4	NA	5
MMP-FP	1	3	3	13	7	7	41	41	165

The formula clearly holds for a 1 × 1 block, since

$$\begin{aligned}\mathfrak{C}_{\text{MMP}-\text{FP}}(2^0, 2^0) &= \sum_{i=0}^{\max(0,0)} \sum_{j=0}^{i} \binom{0}{0} (2^0 \times 2^0) \times S \times \underbrace{f(0,0)}_{=1} \\ &= S\end{aligned} \tag{B.11}$$

Analogous to the deduction done for the dyadic block segmentation, we will also prove the computational complexity formula by induction. Assuming the formula holds for blocks of dimension $(2^m, 2^n)$. Blocks of dimension $(2^{m+1}, 2^n)$ need to perform one vector search for the block without segmentation, then horizontal and vertical segmentation is tested. The total computational complexity of the block is the sum of the complexity for the block, the complexity for its two offsprings after horizontal block segmentation and the complexity for its two offsprings after vertical block segmentation. The complexity is then given by

$$\begin{aligned}\mathfrak{C}_{\text{MMP}-\text{FP}}(2^{m+1}, 2^n) &= (2^{m+1} \times 2^n) \times S + 2 \times \mathfrak{C}_{\text{MMP}-\text{FP}}(2^m, 2^n) \\ &\quad +2 \times \mathfrak{C}_{\text{MMP}-\text{FP}}(2^{m+1}, 2^{n-1}) \\ &= (2^{m+1} \times 2^n)S \\ &\quad +2 \times \left(\sum_{i=0}^{\max(m,n)} \sum_{j=0}^{i} \binom{i}{j} (2^m \times 2^n) \times S \times f(i,j) \right) \\ &\quad +2 \times \left(\sum_{i=0}^{\max(m+1,n-1)} \sum_{j=0}^{i} \binom{i}{j} (2^{m+1} \times 2^{n-1}) \times S \times f(i,j) \right) \\ &= (2^{m+1} \times 2^n)S \\ &\quad + \sum_{i=0}^{\max(m,n)} \sum_{j=0}^{i} \binom{i}{j} (2^{m+1} \times 2^n) \times S \times f(i,j) \\ &\quad + \sum_{i=0}^{\max(m+1,n-1)} \sum_{j=0}^{i} \binom{i}{j} (2^{m+1} \times 2^n) \times S \times f(i,j) \\ &= \sum_{i=0}^{\max(m+1,n)} \sum_{j=0}^{i} \binom{i}{j} (2^{m+1} \times 2^n) \times S \times f(i,j)\end{aligned}$$

Again, the induction for the other coordinate is entirely analogous.

Table B.2: Number of calls of the function that performs the search for the best codeword, taken from the MMP, MMP-FP and MMP-FP+INTRA. Notice that code optimization was implemented in the MMP-FP and MMP-FP+INTRA algorithms, therefore resulting in less calls than calculated by Equations B.9 or B.12.

Block Size	1×1	2×2	4×4	8×8	16×16
MMP	1	3	5	7	9
MMP-FP	1	8	48	224	960
MMP-FP+INTRA	9	28	500	7604	112340

Table B.1 gives a numerical example of the computational complexity. Notice that, just by relaxing the dyadic block division, we have increased the computational complexity tremendously. It is important to stress that the value obtained with Equation B.9 is a pessimistic one, and does not reflect the common block sizes as in Figure B.10. Since the successive segmentations result in blocks with similar dimensions, complexity cost savings can be achieved by storing the best codeword per block dimension and reusing this result, once the codeword search for a block dimensions occurs again. Nevertheless, this incurs in higher memory requirements for the encoder as well.

If prediction is performed, and the residual block is encoded with dictionary codewords, the saving time due to the best codewords reuse cannot be used between predictions residues, since for each prediction the residue might be different. Thus, in case we have M prediction modes, and considering only prediction at the highest block dimension, the computational complexity from Equation B.9 would then become

$$\mathfrak{C}_{\text{MMP}-\text{FP}}(2^m, 2^n) = M \times \sum_{i=0}^{\max(m,n)} \sum_{j=0}^{i} \binom{i}{j} \times (2^m \times 2^n) \times S \times f(i,j) \qquad \text{(B.12)}$$

If we consider hierarchical prediction, for each block dimension, all the prediction modes will be tested. Then the block is divided horizontally and again all the prediction modes will be tested for each one of the two segments. The block is also divided vertically, and the same optimization procedure is done for each half. The formal derivation for the complexity of the hierarchical prediction will not be provided. However, Table B.2 shows some representative values of the computational complexity of the MMP algorithm, if we consider the use of the 9 prediction modes from H.264/AVC with hierarchical prediction and flexible segmentation. This measured values were taken from the MMP software with some implementation optimization, such as the storage of best values for reuse in recurrent block sizes. Even with the computational advantages, the computational complexity increase is noticeable.

B.4 MMP-FAST

The use of prediction does not impose a severe additional computational cost, since it implies only few extra additions. Nevertheless, the choice of the best prediction mode is made comparing the cost of encoding the residue obtained after each prediction, which means that the encoding process has to be repeated M times, one for each prediction mode. This drastically increases the computational cost in the case of MMP-FP prediction. This is so because, besides it being done hierarchically, the number of possible segmentations is greatly increased.

In transform-based methods that use prediction, a common approach towards reducing the computational cost is by selecting only a few modes according to certain criteria, like the gradient method proposed in [55]; however in the MMP framework this would still not avoid the repeated dictionary search for determining the residue's cost of the remaining prediction modes. For each remaining prediction mode, the costly dictionary search would still need to be performed, in order to determine the prediction mode with the lowest coding cost.

One way to avoid the calculation of the cost of encoding the residue of each prediction is to use a different criterion for selecting the best prediction mode. For our fast implementation, we decided to use the residue's energy, instead of the residue's coding cost, choosing the mode that results in the minimum energy. Residues with lower energy tend to be smoother, avoiding block segmentation and reducing bitrate, since less dictionary indexes are sent. Notice that not necessarily residues with the minimum energy will also have the lowest coding cost, since the coding cost depends on the patterns present in the dictionary. If we already have a high energy residue pattern in the dictionary, it could be more efficiently coded, since only one index would be sent and block segmentation would be avoided. However, choosing a lower energy pattern for the prediction's residue may result in a residual pattern that is not present in the dictionary. That would eventually lead to block segmentation and more indexes would be sent, possibly reducing MMP's rate-distortion performance.

With this simple algorithm modification, we avoid several dictionary searches, resulting in a significant reduction in computational complexity, although at a cost in the coding performance, as will be shown in Section B.4.1.

B.4.1 Experimental results

The original algorithm and the fast implementation were both tested under the same conditions. Gray scale smooth and compound images were used in the simulation, and they can be found in Appendix K.

Figures B.13(a) and B.13(b) show the rate-distortion performance for smooth images, while Figures B.14(a) and B.14(b) show the rate-distortion performance for

Table B.3: Percentage of time saved with the new FAST algorithm.

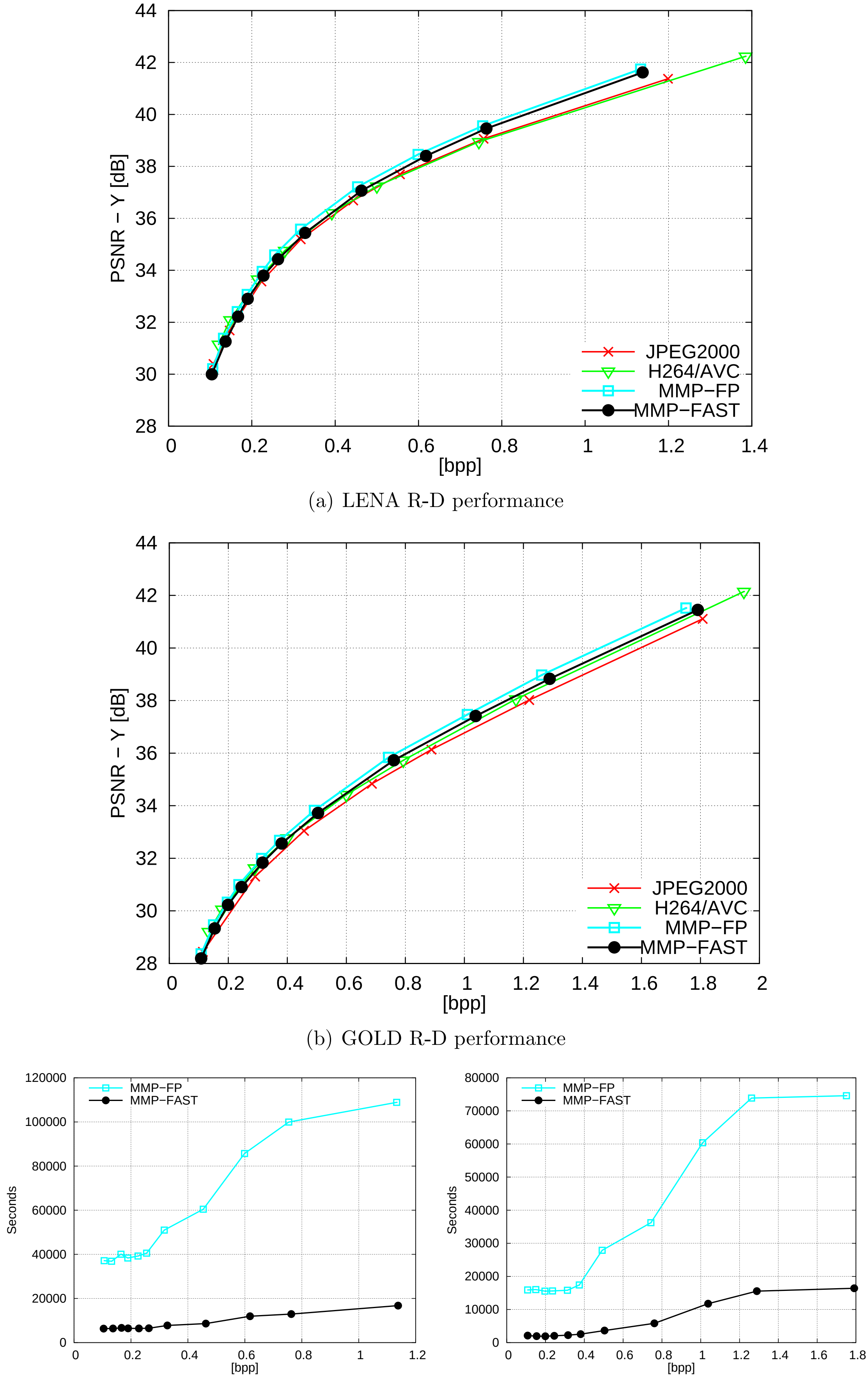
Rate	**0.4 bpp**	**0.7 bpp**	**1.1 bpp**
gold	87%	88%	85%
lena	86%	85%	81%
pp1205	87%	81%	84%
pp1209	86%	86%	82%

compound images. A small decrease in PSNR sense of about 0.2 dB can be seen in all available bit rates for smooth images. In the case of compound images, the rate-distortion penalty is higher, achieving a loss of 1 dB for the text only PP1205 image, and 0.4 dB for the compound image PP1209. Since the algorithm assumes smoothness of the residues for choosing the best prediction, higher losses are expected in images where smoothness criteria do not apply, such as compound images. Nevertheless, for all tested images MMP-FAST still outperforms the encoders used for comparison.

Despite the losses in rate-distortion performance, we have observed a considerable gain in encoding time. Figures B.13(c), B.13(d), B.14(c) and B.14(d) show a significant reduction in encoding time for all bitrates. Table B.3 shows the encoding time saved with the new decision criterion, reducing it in some cases more than seven times.

Table B.4 shows details of encoding image CAMERAMAN with $\lambda = 5$, that is, low distortion at the cost of a high rate. Notice that there has been a significant decrease in encoding time for the FAST version, and that also this algorithm produces a prediction with higher quality. Since the choice of prediction modes is based now on the distortion of the prediction result only, and not on the residue's coding cost, there is an increased tendency to use distortion modes that provide smaller residue. However, this does not translate to a more efficient coding. Instead of using high-energy codewords and avoiding segmentation, prediction will induce segmentation and the use of low-energy codewords. Therefore, more bits need to be spent in segmenting the block, and also for sending the dictionary's codeword indexes, as can be seen on the last three lines of Table B.4.

One of the beneficial side-effects of the FAST algorithm is that the final dictionary size is smaller, which leads to computational gains. Since the prediction favors the use of residues with smaller energy, the dictionary will be populated with patterns that have a similar characteristic. This constrained set of patterns is usually smaller than residue patterns that may appear in the image if the usual MMP algorithm is used. In figure B.15, the growth of all the dictionaries used by MMP is shown, when encoding CAMERAMEN image with high quality. Smaller dictionary



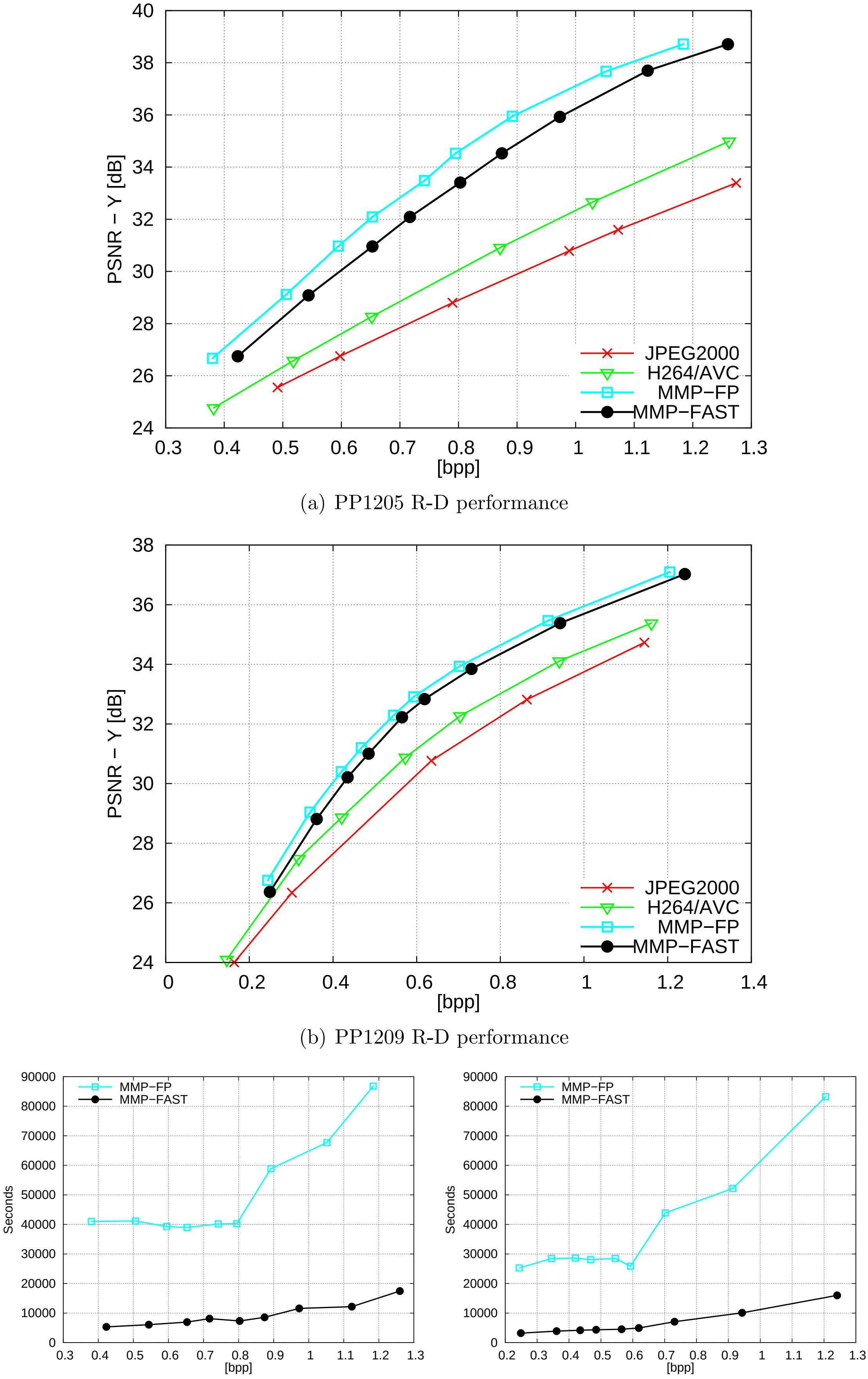
(a) LENA R-D performance

(b) GOLD R-D performance

(c) Lena coding time

(d) GOLD coding time

Figure B.13: Experimental results for images LENA and GOLD.



(a) PP1205 R-D performance

(b) PP1209 R-D performance

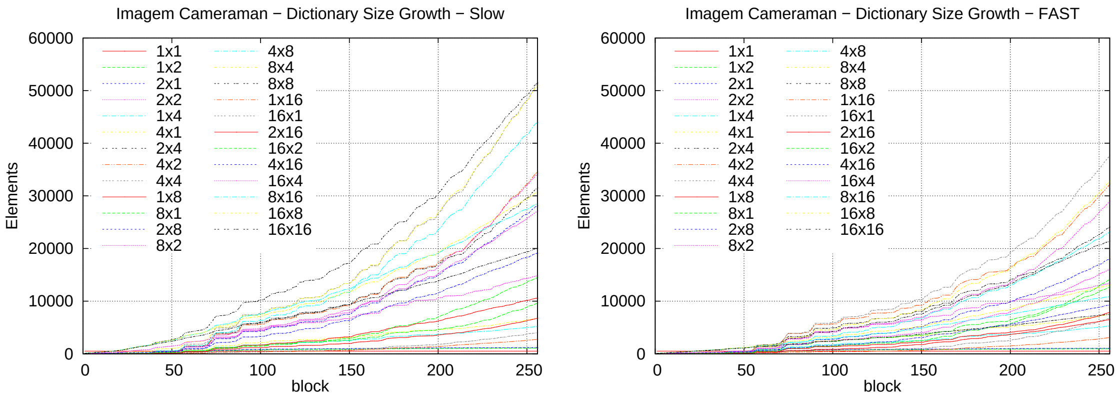
(c) PP1205 coding time

(d) PP1209 coding time

Figure B.14: Experimental results for images PP1205 and PP1209.

Table B.4: Simulation details for CAMERAMAN ($\lambda = 5$), running in an Intel(R) Xeon(R) CPU X5355 @ 2.66GHz, dual processor, each one with four cores, with 8GB RAM. For H.264/AVC and JPEG2000, time for encoding the same image is 0.232 and 0.04 seconds, respectively.

coding parameters	**MMP**	**MMP-FAST**
total rate	3.769 bpp	3.941 bpp
PSNR	42.369 dB	42.174 dB
PSNR (Prediction)	22.153 dB	23.769 dB
Times	6739.770 s	979.640 s
% rate with dictionary indexes	77.1%	78.6%
% rate with prediction mode	6.4%	4.5%
% rate with segmentation flags	16.5%	17.9%



(a) MMP

(b) MMP-FAST

Figure B.15: Dictionary growth for CAMERAMAN image, using $\lambda = 5$.

also implies in lower matching probability. Therefore, MMP-FAST will have to segment the image more often, in order to improve prediction and use the patterns that are in the dictionary, spending more bits with the segmentation and the dictionary indexes, as shown also in Table B.4.

B.5 Conclusions

The MMP algorithm has shown great potential for image encoding and the latest developments have provided MMP with a state-of-the-art rate-distortion performance for compound and smooth images. Nevertheless, these developments have focused only on enhancing the algorithm's performance and computational complexity has not been taken into consideration.

With the proposed fast implementation, we showed that we can significantly reduce the computational complexity and still profit from the good performance of the MMP algorithm. There are several other techniques, such as dictionary clean-up approaches and fast search methods, which can also be incorporated into the MMP framework. One promising area for encoding time reduction is the use of parallel processing and GPUs (*Graphic Processing Unit*). However, still a great effort in research needs to be done, in order to determine which routines in MMP are critical and can be parallelized.

One feature that provided enhancement in MMP's rate-distortion performance is the modification of the algorithm, in order to make it more adaptive, like for example the flexible segmentation scheme. One element of the MMP-FP algorithm that is still rigid is the prediction mode. The 9 prediction modes use a fixed neighborhood weighting function, and cannot adapt its prediction structure to variations inside the block. In Appendix C, we propose to add an adaptive prediction mode, in order to enhance MMP's performance even further.

Apêndice C

Least-squares prediction in MMP

SUMMARY: This appendix starts presenting the theory of least-square predictors in Section C.1. A new prediction mode based on least-square minimization is proposed for the MMP framework. Details on LSP block implementation and how this new prediction mode was incorporated into MMP are given in Sections C.2 and C.3, respectively. Results are presented in Section C.4 and Section C.5 concludes this topic along with some remarks.

C.1 Image modeling

Image modeling is the analytical representation of an image's intensity distribution and can be categorized into probabilistic or deterministic modeling. In probabilistic models, each image is treated as the realization of an event taken from a source, that is governed by some pre-defined probabilistic law, such as multivariate Gaussian models or discrete Markov Random Fields (MRF, [99]). For deterministic models, the image is considered as a two-dimensional data matrix, or simply a set of discrete samples that were taken from a continuous 2D function. Examples of global image representation models include deterministic 2D sinusoidal models, polynomial models [100] and AM-FM models [101]. Local model examples are 2D causal models [102] and the non-symmetric half plane (NSHP) models [103].

In [56], an efficient image model is characterized by being able to capture fundamental features of a natural image, such as edges. For example, H.264/AVC Intra prediction modes explicitly model the image formation process by using neighboring pixels considering 9 different directions for block prediction. If the neighbors used for prediction contain edge pixels, then the edge can be extended inside the block in any of the chosen directions. Nevertheless, such image model is still not capable of handling arbitrarily-oriented edges.

For efficient edge representation, the image model has to take into consideration two fundamental features of an edge:

1. sharp transitions of the image intensity field occur *across* the edge orientation;

2. the image intensity is almost homogenous *along* the edge orientation.

For digital images both properties hold, and are called "geometric constraints of edges" [56]. The first property is related to the sharpness of an image, an important visual feature. The second property indicates that estimations should be done along the edge orientation. Therefore, adaptation to an arbitrarily-oriented edge is a very desirable property for an image model. Next a technique based on Least-Square minimization for image prediction is presented. This technique has been successfully applied for lossless image compression [60] and video compression [104, 105].

C.1.1 Least-squares predictors

The Least-Squares Prediction (LSP) method determines each prediction pixel by adaptively filtering a selected neighborhood. The set of filter coefficients are determined based on training over a window containing reconstructed data [60]. The coefficients are selected according to an N*th* order Markovian model, and often the nearest pixels are used. Lets define an indicator function g(n), which provides the coordinate of a pixel indexed by n in the image. Then, the prediction can be described with the following equation

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

where g(n) is the position to be predicted using N neighbors at positions g$(n-i)$, and a_i are the weighting factors of each i-th position. Notice that the positions indicated by the function g$(n-i)$ are usually neighboring positions of the pixel to be predicted, as shown in Figure C.1, and are also known as *region of support of the predictor* or the *predictor mask* [102].

The coefficients used in LSP prediction are locally optimized using a causal training window in a least-square sense, also known as *analysis frame.* According to [56], a convenient choice of a training window is the double-rectangular window that contains $M = 2T(T+1)$ elements (see Figure C.1). The training sequence can be arranged in a $M \times 1$ column vector $\mathbf{y} = [\mathbf{X}(\mathrm{g}(n-1)) \ldots \mathbf{X}(\mathrm{g}(n-M))]^T$. With the prediction neighbors we form an $M \times N$ matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}(\mathrm{g}(n-1-1)) & \ldots & \mathbf{X}(\mathrm{g}(n-1-N)) \\ \vdots & & \vdots \\ \mathbf{X}(\mathrm{g}(n-M-1)) & \ldots & \mathbf{X}(\mathrm{g}(n-M-N)) \end{bmatrix} \tag{C.2}$$

The coefficients can be determined by LS optimization, minimizing the Mean

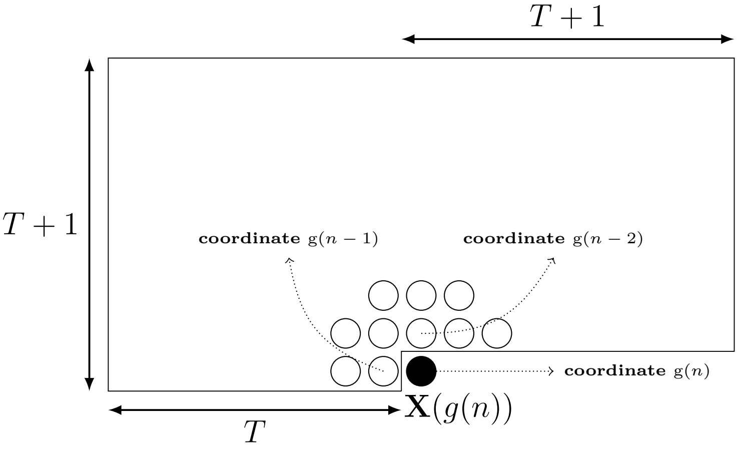


Figure C.1: Selected neighborhood for LSP prediction and training window. The pixels used for prediction and training are located in the causal part of the image, considering a raster scanning order. Therefore, reconstructed values are used, and precisely the same training can be done both in the encoder and in the decoder side.

Square Error (MSE) of the prediction coefficients in the training area M, that is

$$\min(MSE) = \min \left\{ \frac{1}{\#(M)} \sum_{\mathbf{X}(\mathrm{g}(n-k))\subset M} \left(\mathbf{X}(\mathrm{g}(n-k)) - \sum_{i=k-1}^{k-N} a_i \mathbf{X}(\mathrm{g}(n-i)) \right)^2 \right\} \tag{C.3}$$

where $\#(M)$ denotes the number of elements in the set M. Equation C.3 can also be stated in matrix notation

$$\min \{||\mathbf{y} - \mathbf{Ca}||_2\} \tag{C.4}$$

A well-known closed form solution for this problem is [106]:

$$\mathbf{a} = (\mathbf{C}^T\mathbf{C})^{-1}(\mathbf{C}^T\mathbf{y}) \tag{C.5}$$

Least-squares estimation of 1D signals as well as fast implementations can be found in [57]. In [58], several fast implementations for mean-square optimization, such as Singular Value Decomposition (SVD) or LU decomposition are presented in standard C code. LSP is well suited for predicting arbitrarily oriented edges, due to its edge-directed property [60], where the edge pixels play a dominant role in the LS optimization process.

Edge directed property

Pixels inside the training window can be classified into edge pixels (pixels that belong to an edge) or non-edge pixels (pixels from smooth areas). In case the training area contains only non-edge pixels, the matrix C will be singular, and an infinite number

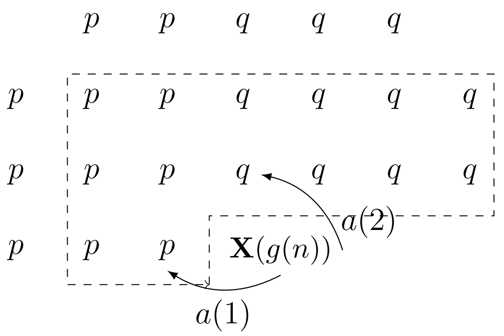


Figure C.2: Edge directed property. This example shows how LSP is capable of determining the best neighbors combination, that is, the optimal coefficents value $a(1)$ and $a(2)$, according to the edge present in the training area.

of solutions for the predictor's coefficients are possible. For such case, the set of solutions can be found in the hyperplane $\sum_{i=1}^{N} a(i) = 1$, and the solution $a(i) = 1/N$ can be chosen with no performance loss. If the training window has enough edge pixels, then matrix C will be non-singular and only one possible solution exists. In this case, LS prediction will successfully identify the edge direction and also correctly predict the new pixel.

The optimal solution will correctly determine the coefficients values, without the need to explicitly determine the edge's orientation. The following simple example illustrates the edge directed property of the LSP solution. Figure C.2 shows the pixel to be estimated, which is situated in a part of an image with only two pixel values, p and q.

The pixel to be predicted, $\mathbf{X}(\mathrm{g}(n))$, is found along a sharp vertical edge ($|p - q| >> 0$). For simplicity, a second-order predictor will be considered ($N = 2$), and the training window has only 12 elements ($T = 2$). According to the chosen predictor, the estimated value will be an weighted average of the left and upper neighbor, in this case $\hat{\mathbf{X}}(\mathrm{g}(n)) = a(1)\mathbf{X}(\mathrm{g}(n-1)) + a(2)\mathbf{X}(\mathrm{g}(n-2)) = a(1)p + a(2)q$. The following matrices are obtained:

$$C = \begin{bmatrix} p & p \\ \vdots & \vdots \\ p & p \\ p & q \\ p & q \\ q & q \\ \vdots & \vdots \\ q & q \end{bmatrix}_{12\times 2} \qquad y = \begin{bmatrix} p \\ \vdots \\ p \\ q \\ \vdots \\ q \end{bmatrix}_{12\times 1} \tag{C.6}$$

The optimal solution will then be given by:

$$\begin{aligned}\mathbf{a} &= (C^TC)^{-1}(C^T\mathbf{y}) \\ &= \left(\begin{bmatrix} 8p^2+4q^2 & 6p^2+2pq+4q^2 \\ 6p^2+2pq+4q^2 & 6p^2+6q^2 \end{bmatrix}\right)^{-1} \begin{bmatrix} 6p^2+2pq+4q^2 \\ 6p^2+6q^2 \end{bmatrix}\end{aligned} \tag{C.7}$$

replacing $\alpha = 8p^2+4q^2$,$\beta = 6p^2+2pq+4q^2$ and $\gamma = 6p^2+6q^2$,

$$\begin{aligned}\begin{bmatrix} a(1) \\ a(2) \end{bmatrix} &= \begin{bmatrix} \frac{c}{\alpha\gamma-\beta^2} & \frac{-\beta}{\alpha\gamma-\beta^2} \\ \frac{-\beta}{\alpha\gamma-\beta^2} & \frac{\alpha}{\alpha\gamma-\beta^2} \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix} \\ &= \begin{bmatrix} \frac{\gamma\beta-\beta\gamma}{\alpha\gamma-\beta^2} \\ \frac{\alpha\gamma-\beta^2}{\alpha\gamma-\beta^2} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}\end{aligned} \tag{C.8}$$

which shows that the optimal solution is aligned with the upper neighbor, corresponding to the vertical edge orientation. A similar closed-form solution can be found for horizontal edges. Experiments in [60] show that LSP predictors are able to adapt to edges with arbitrary orientation.

Filtering interpretation

The same results can be derived from traditional linear prediction theory [57, 102, 107]. The LSP problem can be stated as a linear relationship between two stochastic process $\hat{X}(k)$ and $X(k)$, and the linear model is also referred to as Wiener Filter. The estimation of the present realization of a process is a filtered version of N past samples, that is, the output of an N-th order FIR filter (see Figure C.3). Therefore,

$$\hat{\mathbf{X}}(k) = \sum_{i=1}^{N} a_i \mathbf{X}(k-i) \tag{C.9}$$

where a_i are known as the Wiener filter coefficients.

The filter coefficients can be obtained by minimizing an objective function, given by the mean square error $\xi(k) = E[e^2(k)] = E[(X(k) - \hat{X}(k))^2]$, where the error is defined by the difference between the actual pixel value $X(k)$ and the value obtained after filtering, in vector notation $\hat{X}(k) = \vec{a}^T(k)\vec{X}(k)$. A reasonable consideration for the image's local characteristics is to consider the image formation model as a stationary Gaussian source. In this case, the MMSE prediction for a stationary Gaussian process can be determined by the second-order statistics (that is, the

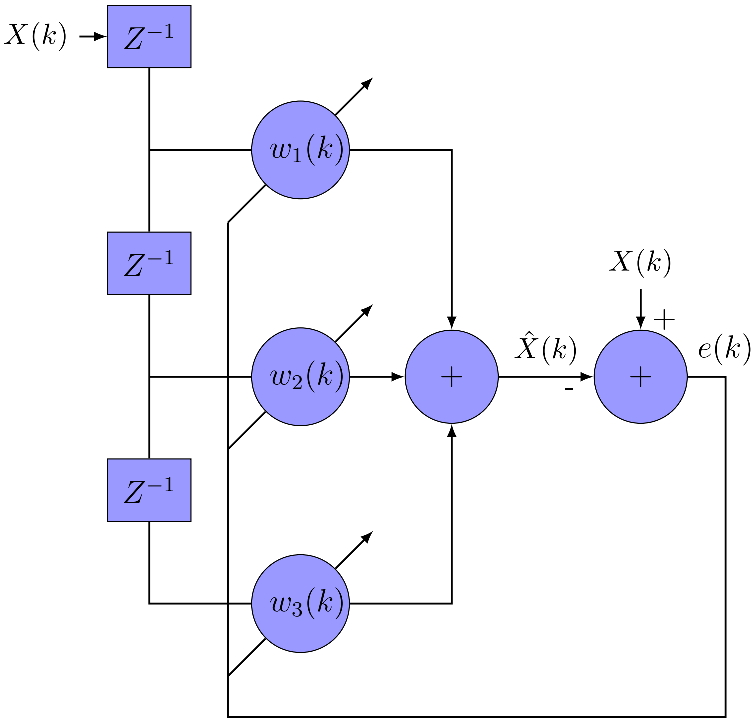


Figure C.3: Wiener filter structure. The input source $X(k)$ is considered an stochastic process, where the k-th realization is linearly related to the N-past realizations. The error between estimation $\hat{X}(k)$ and the present value $X(k)$ is used to tune the filter coefficients.

process' covariance matrix), and is given by

$$\vec{a} = \mathbf{R}_{XX}^{-1}\vec{r}_X \tag{C.10}$$

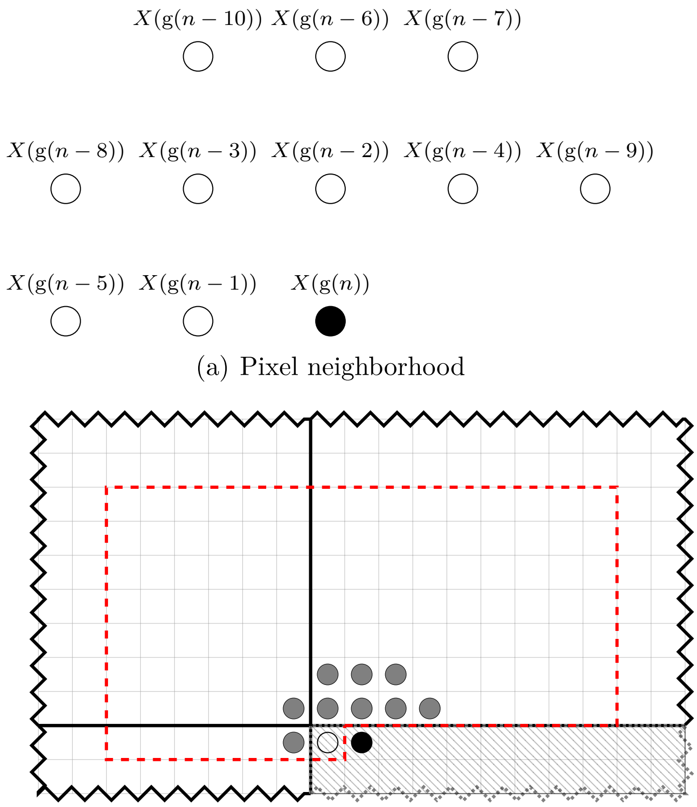
However, the image source often violates the assumption of stationary Gaussian process and the statistics of the image formation vary inside an image from region to region. A practical solution for the non-stationarity behavior is to instantaneously estimate the local statistics $(\vec{r}_X, \mathbf{R}_{XX})$ inside a localized window M. In this sense, the statistics can be calculated as

$$\mathbf{R}_{XX} = \frac{1}{M}\mathbf{C}^T\mathbf{C}, \quad \vec{r}_X = \frac{1}{M}\mathbf{C}^T\vec{y} \tag{C.11}$$

By substituting Equation C.11 in Equation C.10, we obtain again Equation C.5. More details on Wiener Filter and filter theory can be found in [57] and [107].

C.2 Block implementation of LSP predictor

In [59], a block prediction formulation using LSP was proposed. In its lossy approach only a set of previously decoded pixels, on the top and to the left of the current block, are used as the neighborhood for predicting all the pixels inside the block.



(a) Pixel neighborhood

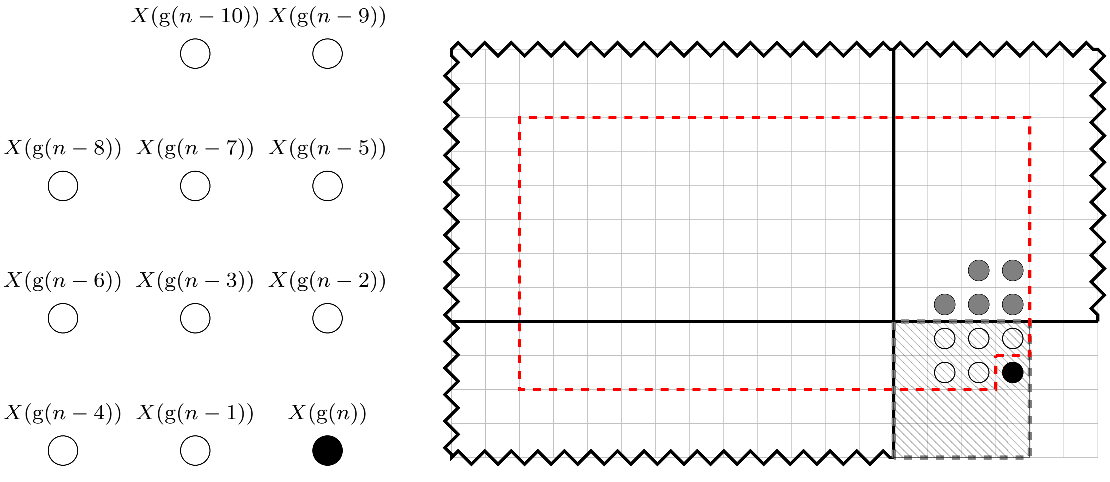
(b) Training area

Figure C.4: LSP block prediction implementation. The thick lines indicate block boundaries, and the gray pixels are positions where the reconstructed pixel value is available. The black dot corresponds to the pixel to be predicted. Notice that the neighbor to the left belongs to the same block where prediction is being applied. In this case, the value used for this pixel will be the predicted value from previous LSP adaptation, and not the reconstructed value.

This is so even for pixels that are far away from these prediction borders, what decreases prediction effectiveness.

The above restriction on the neighborhood is used due to the fact that, since the encoding is block-based, when encoding all the pixels in a block only pixels from the previous block are available for use in the predictor. In our approach, we decided to waive such a restriction. The closest neighbors in a pixel-by-pixel basis are always chosen, even for positions that have not been previously decoded. In such cases, we replace the unavailable reconstructed pixel value by its predicted value, previously determined with LSP. Figure C.4(a) shows the neighborhood chosen for prediction, and Figure C.4(b) illustrates the pixels that will be selected for training.

A limiting factor of the proposed method is that it performs the training using pixels that are to the right of the predicted position, as shown in Figures C.4. However, for block prediction, these pixels may not be available for training, since some of them may belong to a block that still needs to be encoded. Therefore, for such elements we replace the training window and the available neighborhood by



(a) Pixel neighborhood (b) Training area

Figure C.5: LSP block prediction implementation for causal areas. Here, some pixels to the right have not been reconstructed or predicted, since they belong to a block that is still going to be coded. Therefore, training window and predictor support area are changed in order to use only pixels that have either been reconstructed or predicted.

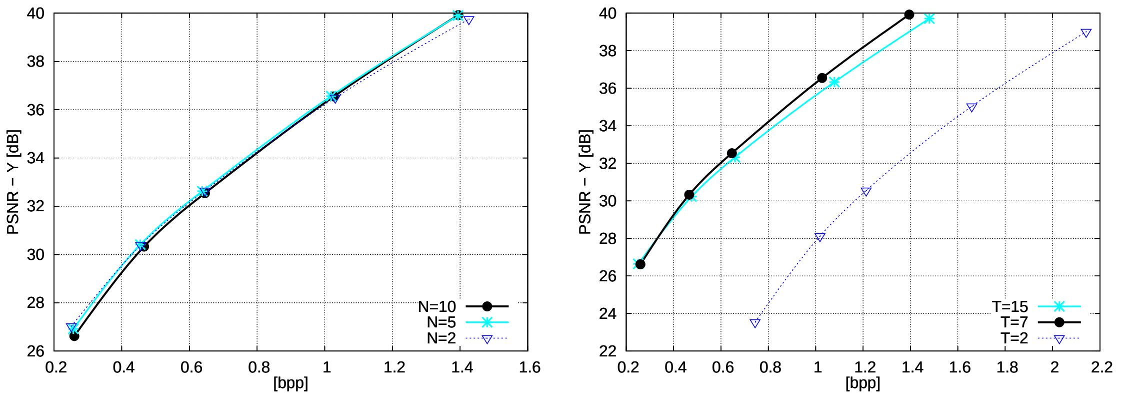
one that has only causal elements. Figure C.5 shows the modified training window and the neighborhood used in such cases; notice that elements on the right are not used for prediction. It is important to note that in both cases (Figures C.4 and C.5), we still need to use the predicted pixels instead of the decoded pixels in order to predict pixels inside a block.

On the image's borders, where there are no available pixels for LSP to perform the training (leftmost and uppermost blocks), the LSP mode is deactivated, and only the H.264/AVC based prediction modes can be used.

C.3 Incorporating LSP predictor into the MMP encoder

The first step was to optimize the parameters of the LSP predictor, that is, to choose the optimal predictor order (N) and the optimal training window (T, as seen in Figure C.1). Experiments were conducted with several images, and Figure C.6 show results obtained for one of the tested images (CAMERAMAN).

The predictor support area has to be large enough in order to successfully capture the image characteristics. Figure C.6(a) compares models with different order, showing their rate-distortion performance. In order to fully exploit the prediction capability of the LSP mode, a model order equal to $N = 10$ was chosen. After choosing the predictor order, the optimal training window size was determined. According to empirical studies conducted in [60] and as can be seen in Figure C.6(b),



(a) Using fixed T=7

(b) Using fixed N=10

Figure C.6: LSP parameter optimization. The MMP encoder uses only the LSP prediction, and the rate-distortion performance for several target rates is obtained while varying the LSP parameters. First, the predictor order was optimized based on the rate-distortion performance. Then with the chosen neighborhood size, the best training window size was determined.

windows larger than $T = 7$ do not further improve the prediction performance. Therefore, the rest of the results presented here will use a predictor model order of $N = 10$ and a training window size set to $T = 7$.

Since the MMP-FP algorithm already uses prediction in its block encoding loop, the new prediction mode can either replace all the 9 available modes, or it can be added to existing H.264/AVC intra prediction modes, the choice among them being made through rate-distortion optimization. Both scenarios were tested, and the results can be seen in Figure C.7. Due to its causal training window, LSP might not be able to capture the image local statistics, which may impair the predictor's performance. However, when LSP is added to the prediction modes set, gains of up to 0.25 dB are achieved for cameraman image. Notice that no extra overhead is needed for signaling the coefficients, since the decoder simply repeats the same training process, using the reconstructed data.

The residue's distribution can be effectively related to a generalized Gaussian distribution [21], described by two parameters: α (shape parameter, describes the decaying rate of the distribution) and β (scale parameter, describes the standard deviation of the distribution). Known distributions such as uniform, Gaussian or Laplacian distributions can be described by generalized Gaussian with $\alpha = \infty$, $\alpha = 2$ and $\alpha = 1$, respectively. In the case of the Barbara image, the parameters for the residue's distribution, using only the regular H.264/AVC prediction modes, are $\alpha = 0.6890$ and $\beta = 17.3640$. The effectiveness of adding LSP mode can be measured by the new GG parameters of the residue distribution, using LSP and the H.264/AVC intra modes. For the Barbara image the parameters assume the new values $\alpha = 0.6777$ and $\beta = 13.5646$, resulting in a more peaked function with less

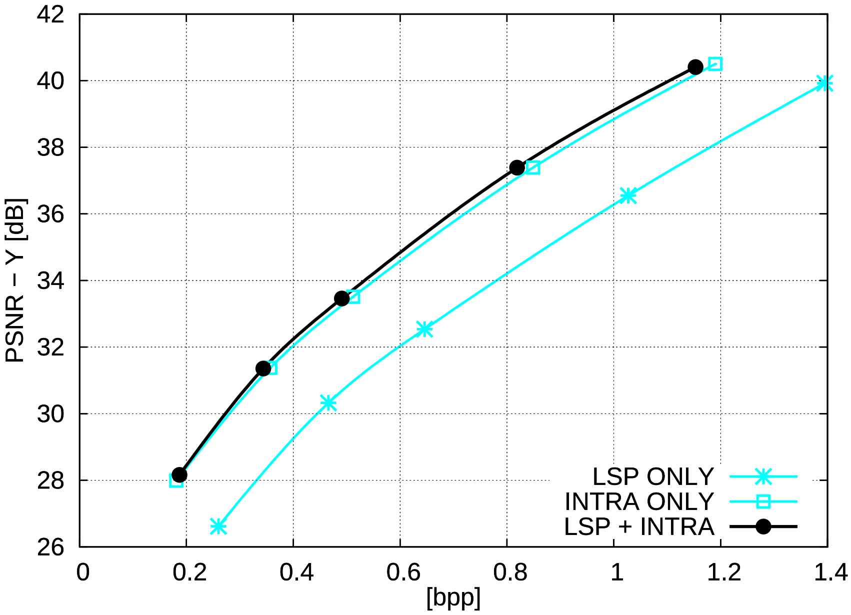


Figure C.7: Results for LSP prediction with optimal parameters (T=7,N=10), for CAMERAMAN image. The LSP prediction alone presents limitations due to the causal training. However, adding the LSP prediction mode to the predictors set increases the rate-distortion performance of MMP.

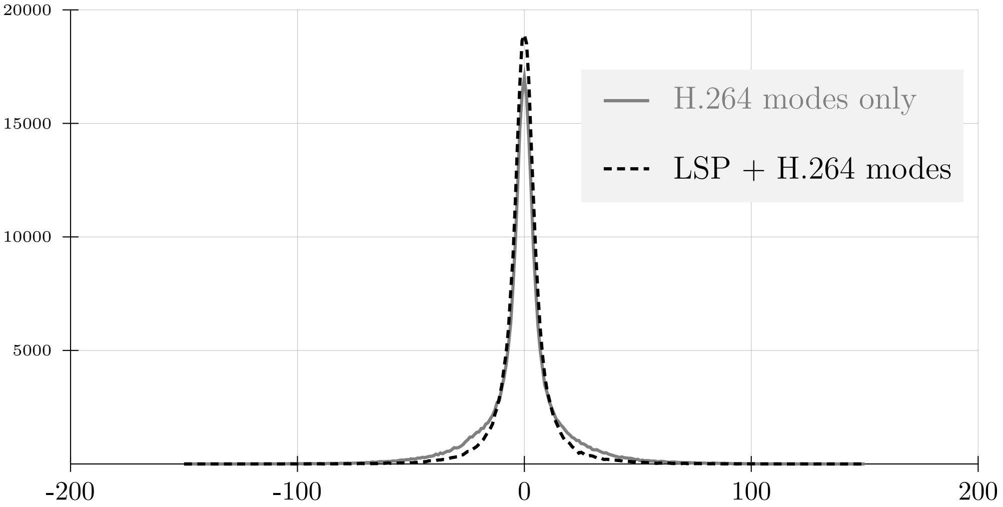


Figure C.8: Residue probability distribution for the Barbara image. Notice that the use of LSP reduces the distribution variance and increases the central peak, that is, prediction results in more values around zero, increasing encoding rate-distortion efficiency.

deviation, as shown in Figure C.8.

As in most pattern-matching based encoders, such as VQ, MMP's complexity is mainly affected by the search of each optimum index. Therefore, the impact of prediction process on the complexity of MMP-FP is negligible. However, when using LSP prediction a matrix inversion operation needs to be performed for each pixel, which has a non-negligible impact on the computational complexity. Fast implementations of LSP have been proposed [60]; however, since the main focus here is the algorithm's compression performance, fast implementations will be left out of the scope of this work. Also, the results were produced without the fast version of the algorithm, proposed in Appendix B, since the main objective here is to evaluate the rate-distortion performance, and not the computational complexity.

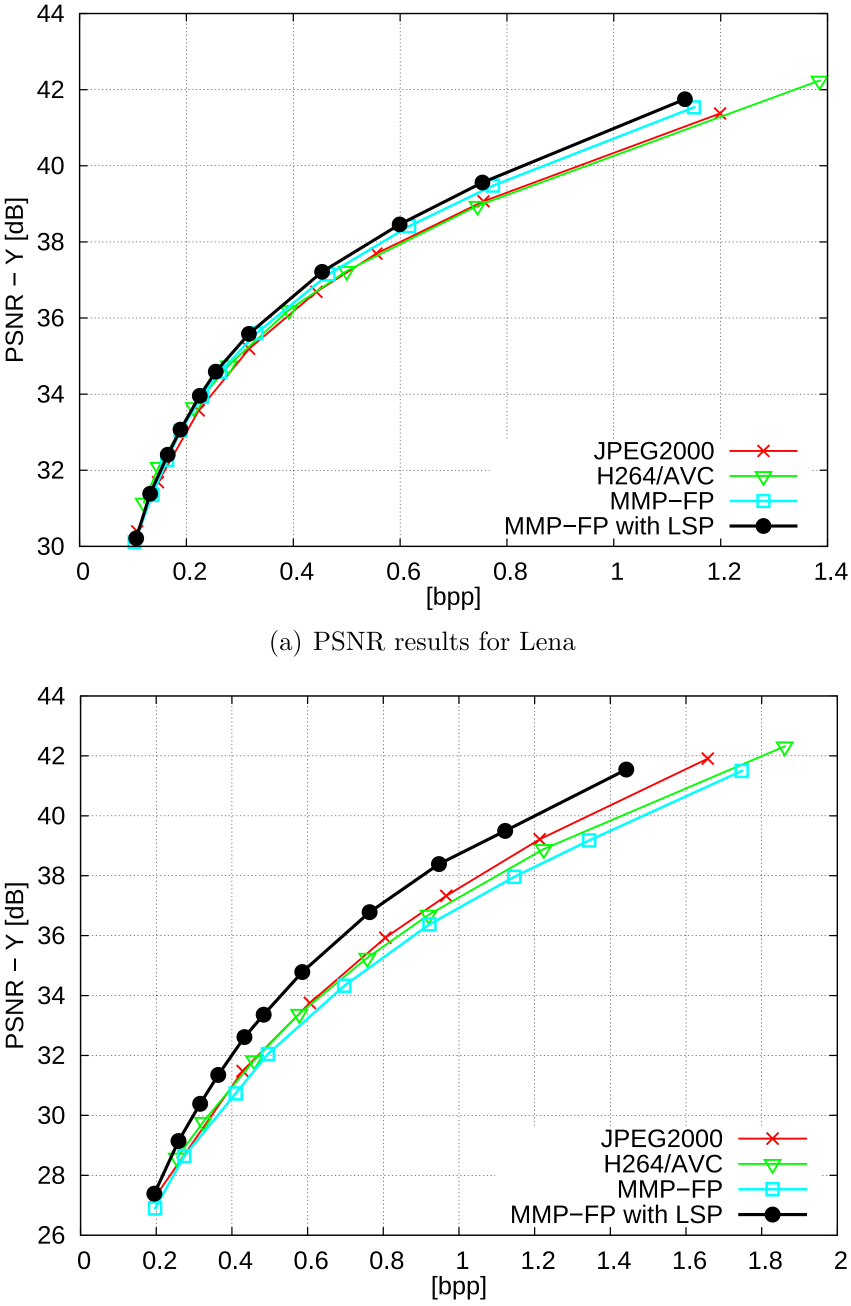
C.4 Experimental results for MMP-LSP

The main motivation for using LSP was to improve the efficiency of the MMP encoder for smooth, textured images, without degrading its performance for non-smooth images. In this section we present some experimental results that justify the use of LSP as an additional prediction mode in MMP-FP. Results of two state-of-the-art image encoding algorithms, JPEG2000 [6] and H.264/AVC *High* profile intra encoder [51], are also presented.

The RD curves for different test images are shown in Figures C.9(a), C.9(b), C.10(a) and C.10(b). One may notice that the gain achieved by LSP prediction depends on the input image. For smooth images with complex texture, like Barbara, gains up to 1.2 dB are achieved, while for other pictures, like Lena, they amount to about 0.25dB. We can also notice that the performance increase is higher for middle to high rates. Since LSP is strongly dependent on the training accuracy, it performs better for those rates, due to the fact that the pixels used in the optimization process are more accurate. In all cases we may observe the PSNR advantage of MMP over the transform-based state-of-the-art methods, JPEG2000 and H.264/AVC.

For text and compound images, like images PP1205 and PP1209, the edges have very steep variations and occur very often, making it difficult for LSP (or any other prediction method) to "learn" an edge from the training window data. Nevertheless, adding the extra prediction mode did not affect the algorithm's rate-distortion performance, as we can see in Figures C.10(a) and C.10(b). There are no notable differences between the MMP curves with or without the LSP prediction mode.

Figure C.11 shows the usage of the available prediction modes in the encoding of the Barbara image, for the original MMP-FP and MMP-FP with LSP, while Figure C.12 shows a similar evaluation for the Lena image. For Barbara image, one may



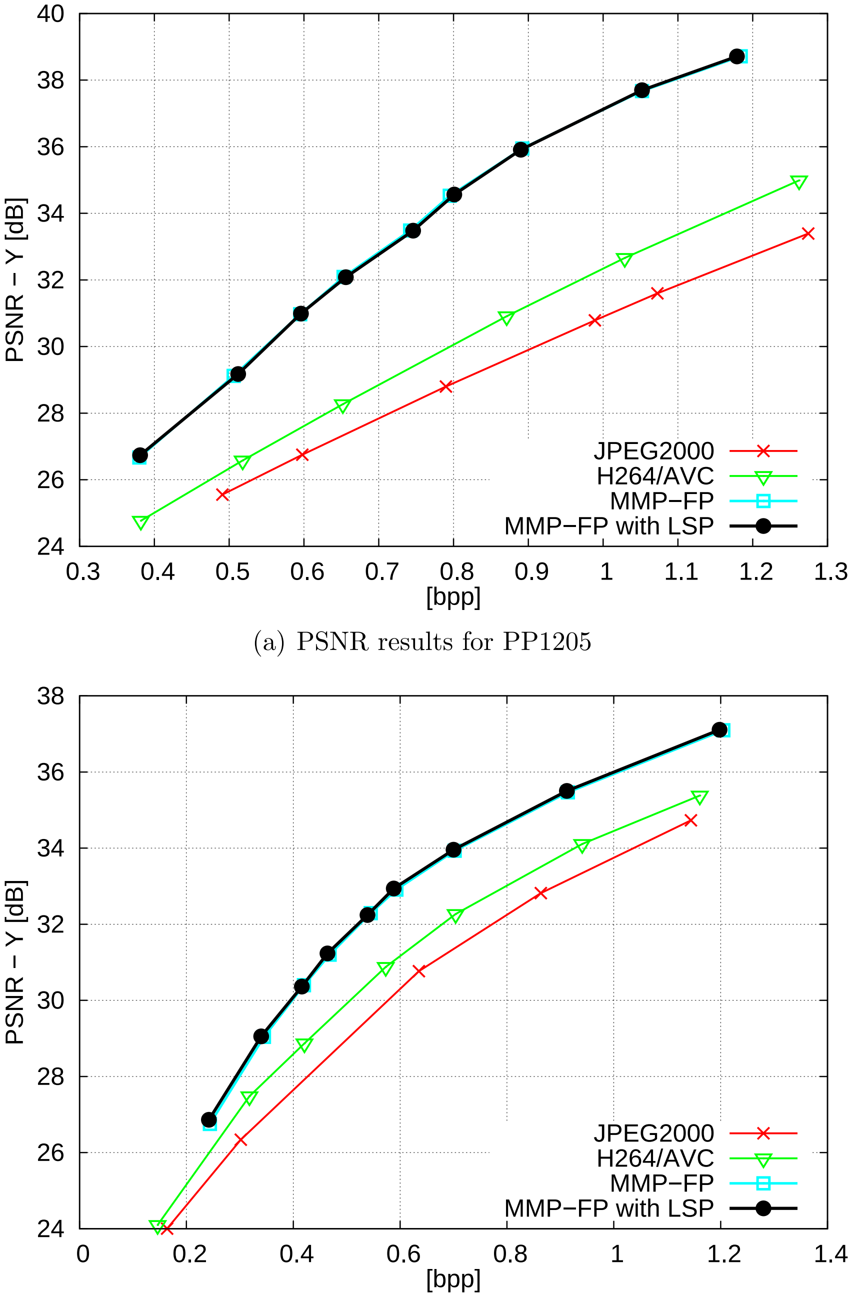
(a) PSNR results for Lena

(b) PSNR results for Barbara

Figure C.9: Rate distortion curves for smooth images.

observe the high adoption of the LSP mode in comparison with the other modes (more than 50%). For images that do not present so many high frequency areas, such as the Lena image, the LSP prediction is used less often. Nevertheless, LSP is still one the most used prediction modes, as seen in Figure C.12.

For the Barbara image, the LSP mode was chosen mainly in areas with strong edges, specially in the woman's clothes, as depicted in Figure C.13. In such areas, the presence of strong edges in the training window contribute for the correct prediction of the pixel during the LS optimization process.



(a) PSNR results for PP1205

(b) PSNR results for PP1209

Figure C.10: Rate distortion curves for compound images.

Figures C.14(a) to C.14(d) show regions with high frequency detail of the coded Barbara image. Ringing artifacts of the JPEG encoder affect the high frequency pattern. In the case of H.264/AVC encoder and MMP-FP, the prediction modes used cannot correctly capture the orientation of the edge, and artifacts at block boundaries are noticeable. The best outcome is the image coded using the LSP predictor, that is able to successfully adapt to the edge's orientation and reproduce the edge with higher fidelity.

For the Lena image, the LSP prediction mode is adopted at edge areas, for

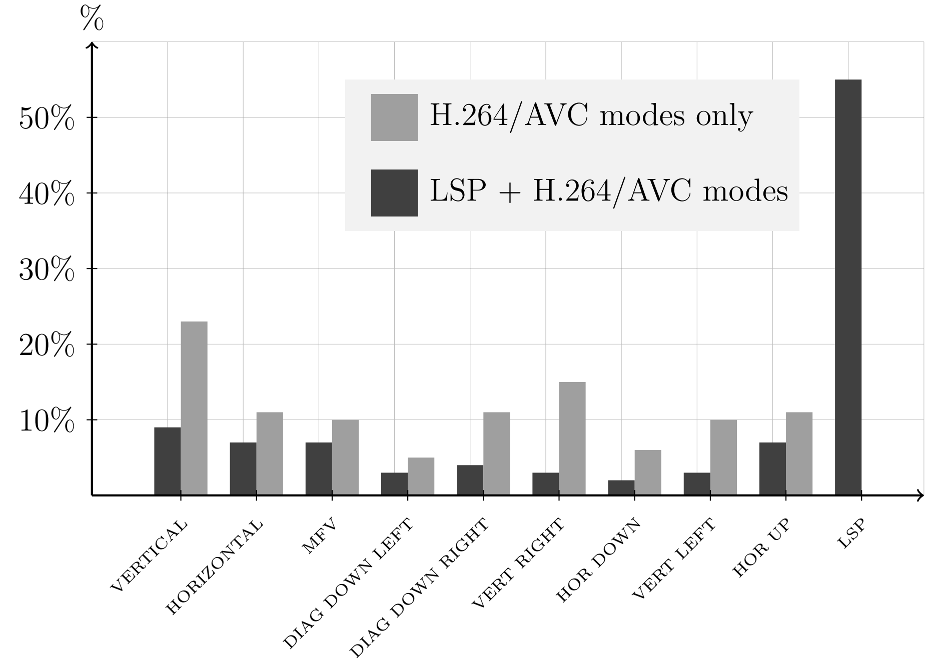


Figure C.11: Percentages of choice for each prediction mode for image Barbara. When using the LSP mode, the use of the other modes is noticeably reduced.

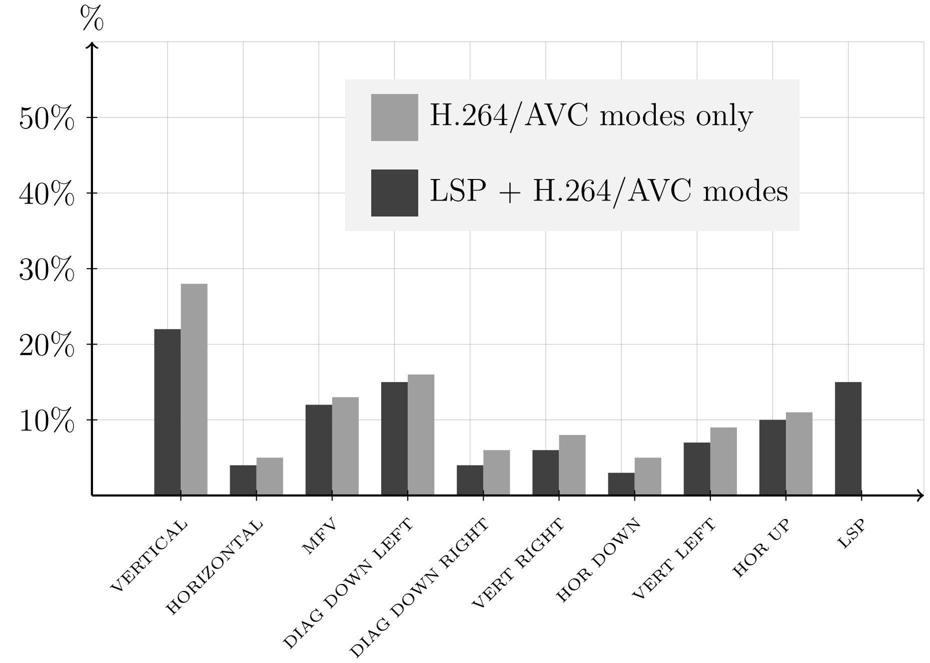


Figure C.12: Percentages of choice for each prediction mode for image Lena. Since Lena image has more smooth areas, the LSP is not adopted as often as it was for the Barbara image. Nevertheless, it is still one of the most adopted modes, and a noticeable reduction on the direction modes can be seen.

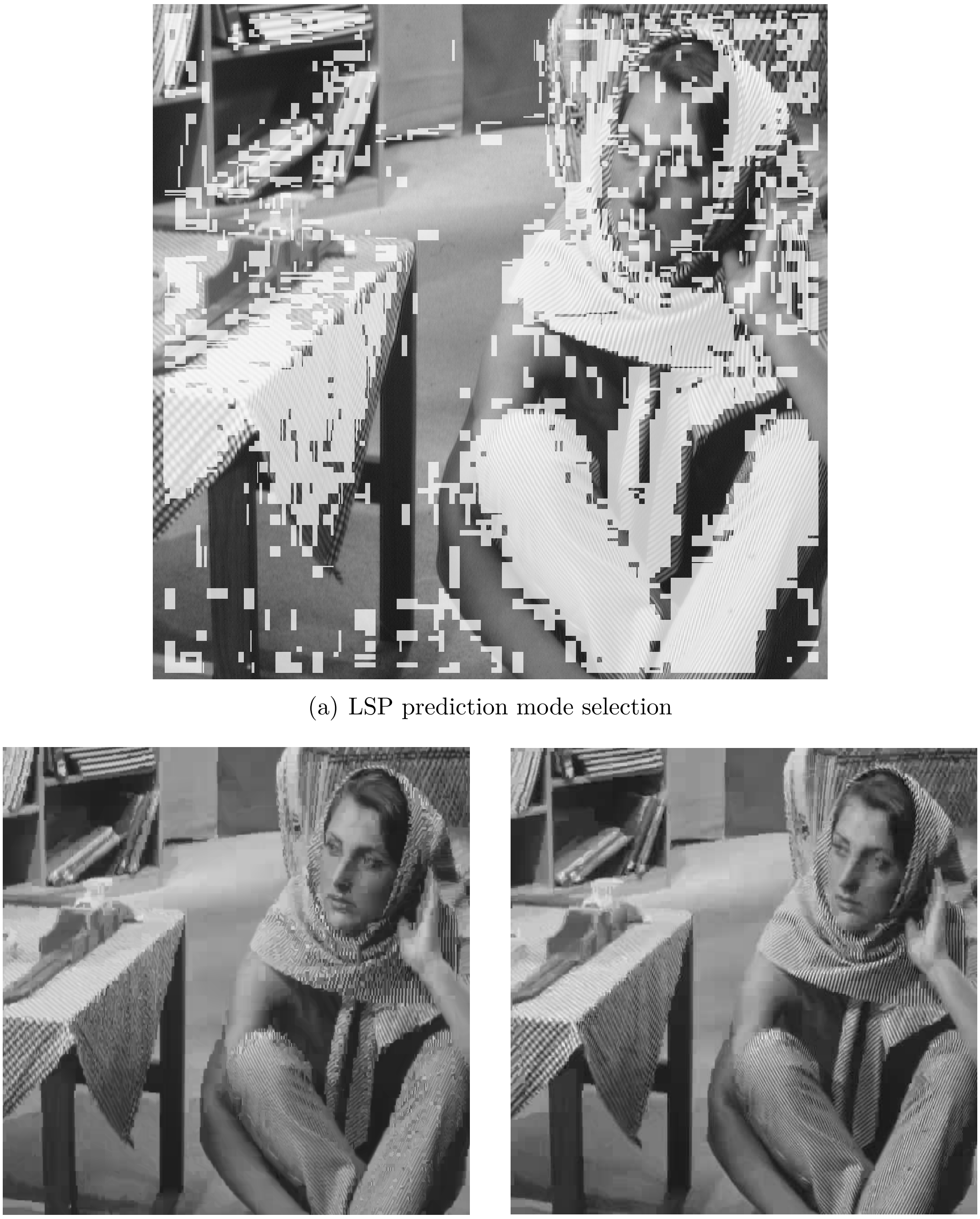
example at diagonal pattern at Lena's hat and at the feathers of her hat, as seen in Figure C.15. LSP is more efficient than the H.264/AVC directional modes at some edge areas, when the training sequence is correlated to the edge to be predicted.

C.5 Conclusions

In this appendix we proposed an image encoding algorithm where least-squares prediction is used in a multiscale recurrent pattern image encoding framework. The addition of a LSP prediction mode can successfully estimate local texture features using linear prediction coefficients, derived on the fly. Due to this fact, LSP prediction mode tends to be predominant over the other ones. The LSP mode is usually chosen to predict the edges, and in images with large areas of high frequency content benefit from this adaptive prediction mode. This yields in some cases gains of more than 1 dB in the range of middle to high rates.

Besides increasing the encoding performance for smooth images, MMP with LSP prediction presented no rate-distortion performance losses for text and compound images. The proposed method outperforms state-of-the-art, transform-based compression algorithms for all image types, from smooth to text and compound images.

LSP has shown to be able to increase MMP's RD performance. Therefore, its was incorporated into MMP prediction modes set, and will be used for the rest of this thesis. Topics for future works include alternative methods for fast coefficient estimation such as Recursive Least Square (RLS).

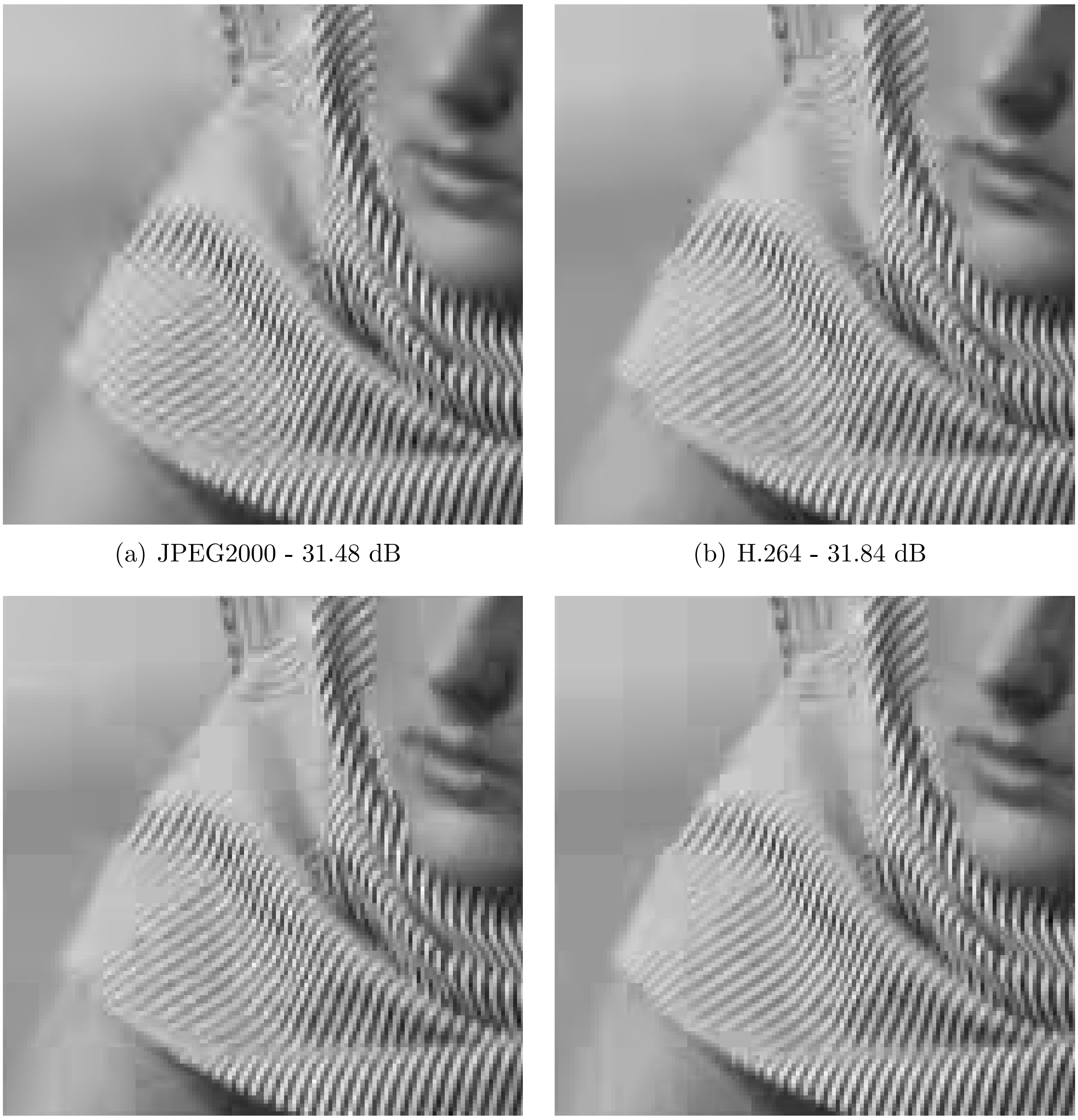


(a) LSP prediction mode selection

(b) Prediction Frame using H.264/AVC modes only

(c) Prediction Frame using H.264/AVC modes and LSP

Figure C.13: Barbara LSP prediction.The brighter blocks are in areas where the LSP mode were used. In Figures C.13(b) and C.13(c), the resulting prediction frame can be seen. The resulting rate for the compressed image was 0.947 bpp with final quality of 38.387 dB. The PSNR value of the prediction frame compared to image to be encoded is 26.816 dB.



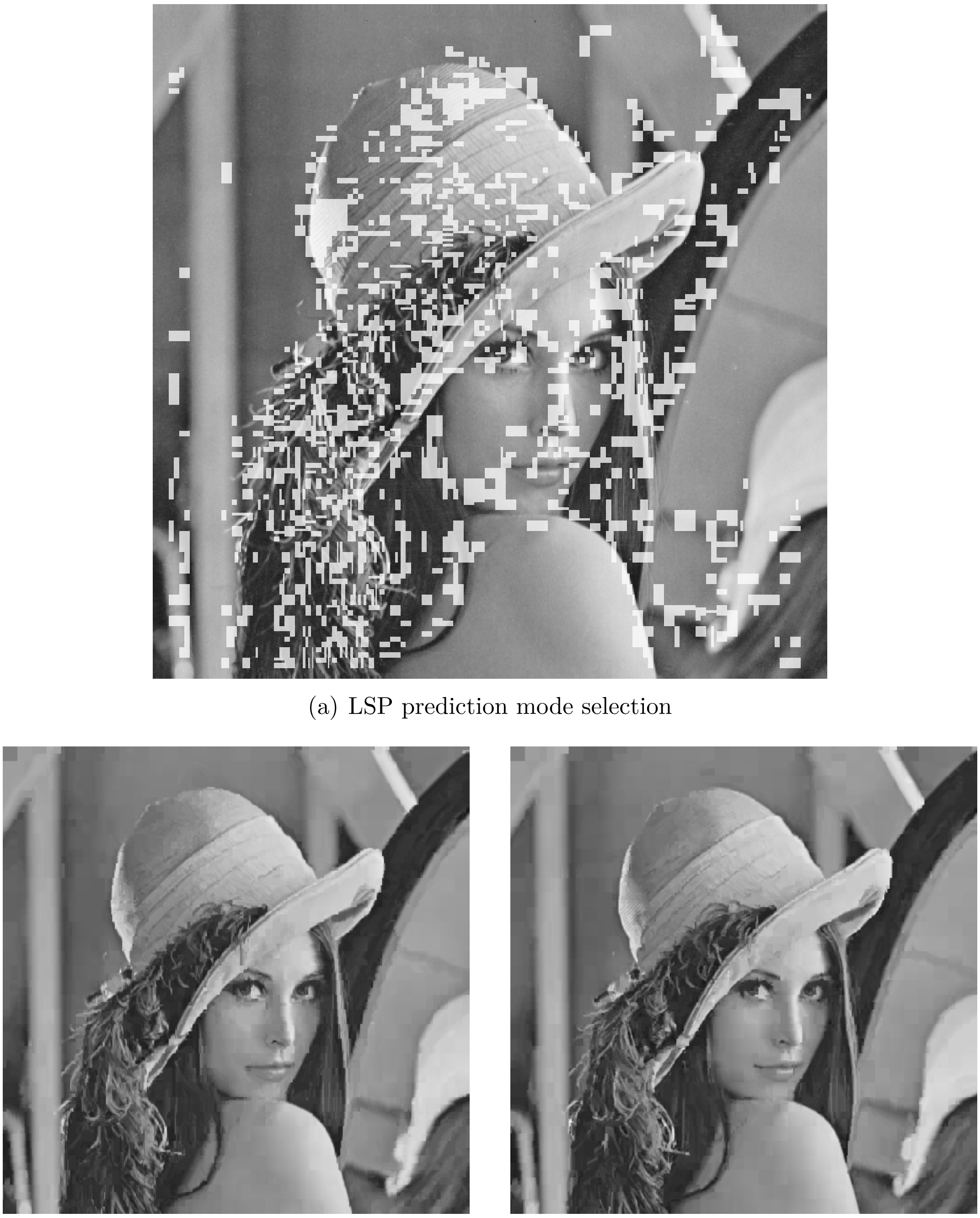
(a) JPEG2000 - 31.48 dB

(b) H.264 - 31.84 dB

(c) MMP-FP - 32.048 dB

(d) MMP-FP with LSP - 32.612 dB

Figure C.14: Detail of Barbara image coded at 0.5 bpp. The detail comprises part of the cloth at the woman's shoulder, where a high frequency pattern can be found. The diagonal pattern of the cloth is better preserved with the LSP prediction scheme.



(a) LSP prediction mode selection

(b) Prediction Frame using H.264/AVC modes only

(c) Prediction Frame using H.264/AVC modes and LSP

Figure C.15: Lena LSP prediction. The brigther blocks are in areas where the LSP mode were used. Figures C.15(b) and C.15(c) shows the resulting prediction frames. The resulting rate for the compressed image was 0.454 bpp with final quality of 37.215 dB. The PSNR value of the prediction frame compared to image to be encoded is 28.999 dB.

Apêndice D

Lossless image compression using MMP

SUMMARY: This appendix deals with the usage of MMP as a lossless image encoder. Section D.2 reviews some of the state-of-the-art algorithms used for lossless image compression. Results for selected lossless coding algorithms are given in Section D.3. In Section D.4, a performance bound for MMP lossless compression is derived, showing that the MMP can asymptotically achieve the entropy rate of any stationary, ergodic, memoryless source with finite alphabet. Section D.5 proposes the adaptation of several well-known typical lossless coding techniques into MMP's framework. Proposals for enhancing MMP's prediction and residue coding are presented, and the results are compared with state-of-the-art lossless image encoders. Section D.6 draws some conclusions for this topic.

D.1 Lossless compression

Lossless compression is mainly used in applications where compression artifacts are unacceptable, as, for example, in the case of medical imaging, preservation of artwork or satellite images [108]. Several algorithms were developed taking into consideration the available resources and the final compression performance, like the JPEG-LS standard [66] and many others [108, 109]. With the increase in computational power and improved quality of terminals, together with the decrease in prices and the evolution of editing softwares, it makes sense to reevaluate the role of lossless coding, and once again consider its application in a broader scenario.

Most lossless algorithms compress an image using two independent steps [110]: image *modeling* and image *coding*. The image modeling process can be regarded as an inductive inference problem, where one needs to find an appropriate model class, model order and model parameters to suitably represent the data. If the model is successful in guessing the next pixel's value, then the residual information will have a skewed probability distribution centered at zero, and can be effectively compressed in the coding step.

For the coding step variable-length codes such as Huffman code, Rice codes or Arithmetic codes are usually used [108]. For such algorithms, codewords with high probability are represented with fewer bits, improving compression results from highly peaked probability distributions, such as the ones obtained from the prediction residues. For efficient implementation, some algorithms assume a distinct probability distribution, such as an Exponential or Generalized Gaussian distribution. Examples of lossless image encoders that use such models for their residue's statistics are JPEG-LS [66] and MRP [111] algorithms. However, compression efficiency decreases when the model does not fit the actual probability distribution, which can occur if the prediction fails and residues have many values distant from zero.

One way to avoid assuming a specific model for the sources' statistics is to use adaptive methods based on pattern matching. Pattern matching algorithms rely on the fact that sequences tend to occur repeatedly on the source to be encoded, and thus the source may be efficiently compressed by detecting and encoding these repeated patterns using fewer bits. This can be accomplished by using dictionary entries for the occurring patterns, and adapting the dictionary codewords as the source is encoded. Investigating the paradigm posed by pattern matching algorithms may result in a higher coding efficiency than the ones achieved now. Besides that, dictionary techniques already possess a good rate-distortion performance for non-smooth images, such as images with text and graphics.

Early versions of pattern matching lossless encoding algorithms were presented by Lempel and Ziv [31] and focused on text encoding, dealing with 1D data. Its evolution involved better dictionary updating techniques [37, 38]. There are also implementations focused on the application of approximate pattern matching for lossy encoding, also known as Lossy-Lempel-Ziv algorithms [42, 43, 112]. Other examples of pattern matching-based algorithms are the Grammar Codes [113] and the deflate compression algorithm used in the PNG file format [114].

D.2 Reviews of state-of-the-art lossless compression algorithms

In this section we will review some of the techniques used for well-known lossless image compression algorithms, for both stages: the image modeling and image coding. Some of the techniques were later adapted to the MMP algorithm, and will

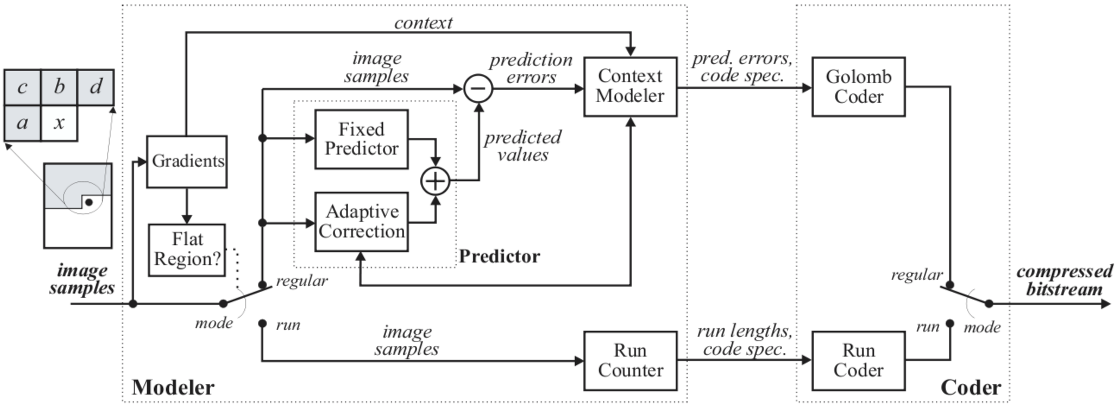


Figure D.1: Block diagram of JPEG-LS algorithm, taken from [66].

be explained in the next section. More details on lossless coding algorithm can be found in the literature [108, 115, 116] or in a survey done in the scope of this thesis, that is available online [117].

JPEG-LS: Also known as LOCO-I (LOw COmplexity LOssless COmpression for Images, [66]), JPEG-LS is a lossless image encoder based on a simple fixed context model followed by an adaptively chosen Golomb-type entropy encoder.

Coding of the pixels can be done in two modes, a run mode, for image regions that have little variation, and a regular mode. The image local gradient ($g_1 = d - b$, $g_2 = b - c$ and $g_3 = c - b$, represented by the “Gradient” block in Figure D.1) will determine the encoding mode.

For the run mode, used in flat regions, blocks of data with equal value are coded with a run-length encoder (see “Run Counter” and “Run Coder” blocks in Figure D.1). The runs are coded with an adaptive Golomb code.

For the regular mode, each pixel is predicted and the residue is entropy encoded also with a Golomb code. Prediction in JPEG-LS is constituted by a fixed part (depicted in Figure D.1 as the “Fixed Predictor” block), together with an adaptive part (depicted in Figure D.1 as the “Adaptive Correction” block).

The fixed part performs a primitive test to detect horizontal or vertical edges, choosing the best prediction in a non-linear fashion, by analyzing pixel neighbors (a, b and c). The guessed value can also be interpreted as a *median* of three fixed predictors, a, b and a+b-c, that is why the algorithm is also called “median edge detector” (MED). The criteria for choosing the predictor is the following:

$$\hat{x}_{MED} \triangleq \begin{cases} \min(a,b) & \text{if } c \geq \max(a,b)\ , \\ \max(a,b) & \text{if } c \leq \min(a,b)\ , \\ a+b-c & \text{otherwise.} \end{cases} \tag{D.1}$$

The statistics of residues from a fixed-predictor in continuous-tone image are well modeled by a two-sided geometric distribution (TSGD) centered at zero, $P(\epsilon) = \theta^{|\epsilon|}$, where ϵ is the integer value of the prediction error and $\theta \in (0,1)$. However, for context-conditioned prediction error signals, a DC offset is also present, due to integer-value constraints and possible bias in the prediction step. The DC offset, also called "bias", can be removed from the fixed prediction value, and this operation is regarded as the "adaptive" part of the prediction. The fixed prediction offset can be broken into an integer part R and a fractional part s, where $0 \leq s < 1$. The bias cancellation procedure eliminates the integer part of the DC offset using a low complexity algorithm, where the estimate of R is obtained by rounding the average of the last N residues avoiding division operation.

The prediction residue with bias correction assumes a probability distribution given by the TSGD

$$P_{(\theta,s)}(\epsilon) = C(\theta,s)\theta^{|\epsilon+s|}, \text{ where } \epsilon = 0, \pm1, \pm2 \tag{D.2}$$

and $C(\theta,s) = (1-\theta)/(\theta^{1-s}+\theta^{s})$ is the normalization factor. A specific Golomb code will be chosen for entropy encoding of the residue, based on the choice of parameters θ and s. For computational efficiency, Golomb-power-of-2 (GOP2, G_{2^k}) codes are used, and a simple rule is used for determining the best k, in accordance to the residue's context [66].

For a given prediction $\hat{x}$, the residue can only take values in the range $-\hat{x} \leq \epsilon < \alpha - \hat{x}$, where $\alpha = 2^{\beta}$ is the largest value of a finite size alphabet of length β. Based on this fact, the residue will be remapped, which for the probability means merging the tails of the distribution with their central part. This procedure does not significantly affect the two-sided geometric behavior, and is of great benefit for compound documents, since it assigns a large probability to sharp transitions. The mapping is easily done just by interpreting the β least significant bits of ϵ in 2's complement. The residue remapping process is detailed in Section D.5.2.

CALIC: The encoding procedure uses a large number of modeling contexts to condition a nonlinear predictor and adapt it to various source statistics [65].

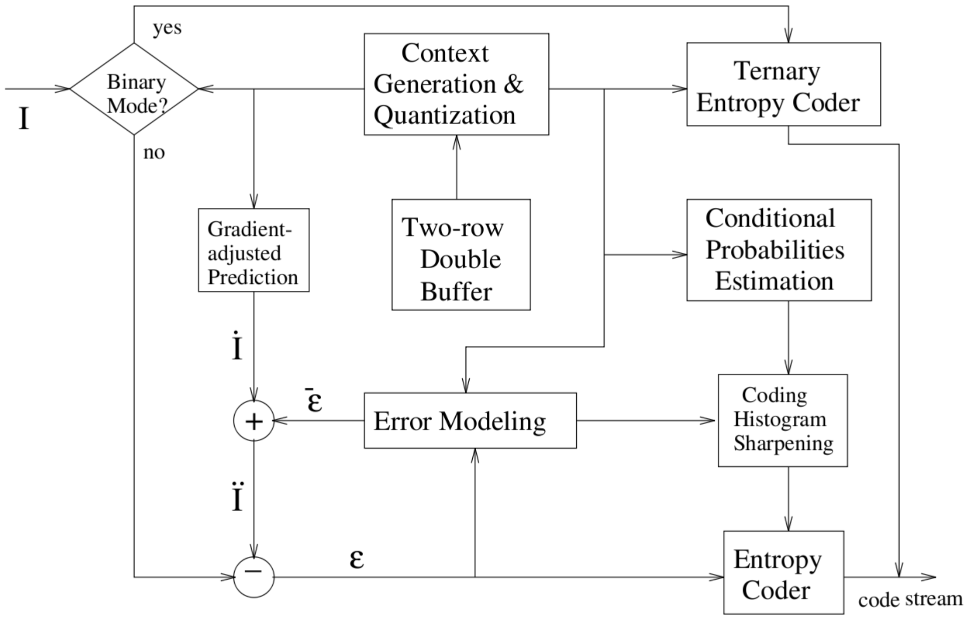

yes
Binary Mode?
I
no
Context Generation & Quantization
Ternary Entropy Coder
Gradient-adjusted Prediction
Two-row Double Buffer
Conditional Probabilities Estimation
İ
ε̄
Error Modeling
Coding Histogram Sharpening
Ï
ε
Entropy Coder
code stream

Figure D.2: Block diagram of CALIC encoder, taken from [118].

In Figure D.2, a block diagram of the CALIC encoding process is shown.

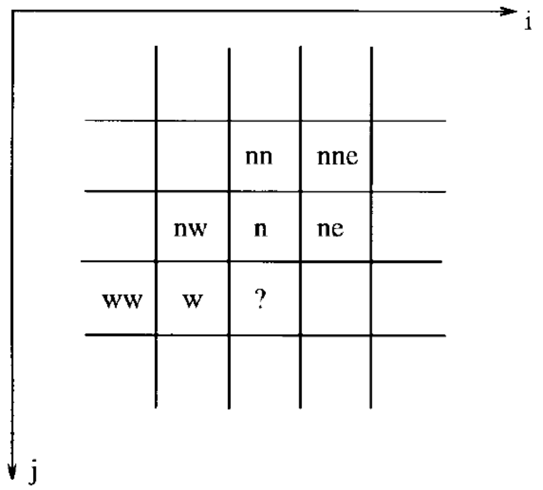


Figure D.3: Neighborhood labeling used in CALIC's prediction, taken from [118].

Binary mode is for situations in which the current locality of the input image has no more than two distinct intensity levels. The algorithm checks six neighboring pixels: I_{ww}, I_w, I_{nw}, I_n, I_{ne}, I_{nn} (see Figure D.3). If these six pixels have no more than two different values, the binary mode is automatically switched on. In such cases, predictive coding may have a poor performance, since the smoothness assumption does not hold anymore, and the direct coding of the pixel values can be more effective than coding the prediction errors. A context-based adaptive ternary arithmetic coder is used to code three symbols, the two possible levels and an escape code to exit the binary mode. The

6-order model context is determined by the quantization of a texture pattern around the pixel.

In the continuous mode, the encoder uses the gradient-adjusted prediction (GAP), in order to determine the appropriate prediction ($\hat{I}$) that will be used for the current position. GAP is more robust than traditional DPCM-like linear predictors, particularly in areas of strong edges. The prediction value is determined in the following manner. At first, an estimate of the image's local horizontal and vertical gradient (d_h and d_v, respectively) is given by:

$$\begin{aligned} d_h &= |I_w - I_{ww}| + |I_n - I_{nw}| + |I_n - I_{ne}| \\ d_v &= |I_w - I_{nw}| + |I_n - I_{nn}| + |I_{ne} - I_{nne}| \end{aligned} \tag{D.3}$$

According to the obtained gradients, the type of edge is empirically estimated and the respective prediction is given according to the pseudo-code below.

```
if (d_v - d_h > 80)
  {sharp horizontal edge} I_pred[i,j]=I_w;
else
  if (d_v - d_h < -80)
    {sharp vertical edge} I_pred[i,j]=I_n;
  else
  {
    I_pred[i,j]=(I_w+I_n)/2+(I_ne-I_nw)/4;
    if (d_v - d_h > 32)
      {horizontal edge} I_pred[i,j] = (I_pred[i,j]+I_w)/2
    else
      if (d_v -d_h > 8)
        {weak horizontal edge} I_pred[i,j] = (3I_pred[i,j]+I_w)/4
      else
        if (d_v - d_h < -32)
          {vertical edge} I_pred[i,j] = (I_pred[i,j]+I_n)/2
        else
          if (d_v -d_h < -8)
            {weak vertical edge} I_pred[i,j] = (3I_pred[i,j]+I_n)/4
  }
```

Note that most of the operations were chosen with the aim to simplify the computation. The thresholds and coefficients could be optimized for an image or class of image, but this procedure is computationally demanding and not recommended.

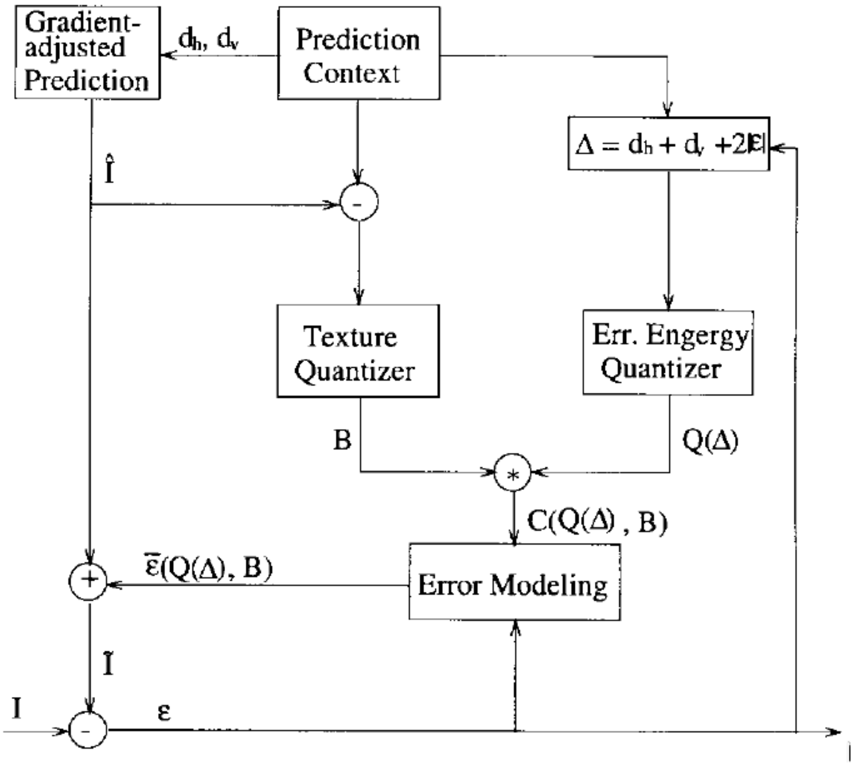


Figure D.4: Details of the context modeling and error feedback procedure for the two-stage adaptive prediction scheme, taken from [118].

Since the prediction step does not fully remove the statistical redundancy in the image, we can still observe a strong correlation between the prediction error variance and the image smoothness. Similar to JPEG-LS, an error feedback loop is added to the encoder, in order to adjust the GAP prediction, generating an improved context-sensitive prediction $\tilde{I} = \hat{I} + \bar{e}$ (see Figure D.4). The error correction factor is obtained by estimating the conditional expectations of the prediction error within each context, $\bar{e}E\{e|Q(\delta), B\}$. The model used for the context combines two measurements: an error energy estimator $(Q(\delta))$ and texture contexts, by a local neighborhood of pixel values (B). This results in a total of 576 compound contexts, where the thresholds were obtained training the set of all ISO test images. More details on the context creation are provided by [118]. With this error feedback loop, a new error prediction is calculated from $e = I - \tilde{I}$, and its absolute value will be arithmetic encoded conditioned on the context generated by the error energy estimator.

With this clean separation between context-based prediction and modeling of image data and entropy encoding of prediction errors, CALIC software can interface with any entropy coding techniques. In the standard, a simple ternary adaptive arithmetic coder is used for the binary mode, and an adaptive m-ary arithmetic coder is used for the continuous-tone mode. Besides that, some

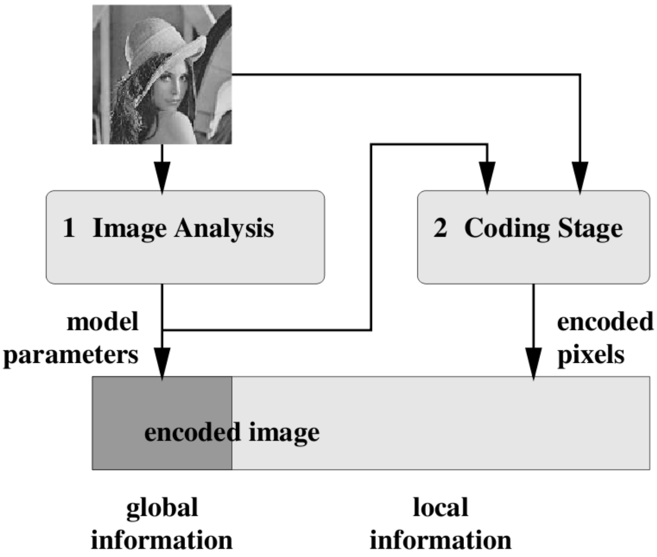


Figure D.5: Block diagram for TMW lossless image compression algorithm, taken from [119].

techniques were added to the entropy encoder, to increase efficiency: remapping errors (reducing the alphabet size), histogram tail truncation (based on the fact that most of the error population relies within a small range [-8,8], error residues will be entropy encoded using an escape code) and dynamic bit shifting (decomposition of the error in case of frequent occurrence of large errors, according to an adaptive average magnitude threshold, for sending the least significant bits of the error using a separate set of contexts or simply as is). More details can be found in [65].

TMW: This method uses a two stage encoding process [119]. In the 1st stage, called Image Analysis, a set of model parameters is designed in a way that minimizes the length of the encoded image. This set of model parameters is then used in the 2nd stage, the Coding Stage, to perform the encoding step. The model parameters are sent to the decoder as global information, while the encoded pixels are sent as local information (see Figure D.5).

The model is based on linear predictors. It uses multiple pixel-predictors of the form

$$\text{pred} = \sum_{i=1}^{M} w_i * \text{pixel}_i \tag{D.4}$$

where M is the number of causal neighbors (a value of 12 has found to be a good choice for images 512x512) and the weights (w_i) are calculated in the image analysis phase. Instead of combining the values obtained by each prediction, the error probability distribution from each predictor is blended instead. A probability of error is then calculated based on a variation of a t-student distribution.

$$p(x < X) = K \int_{-\infty}^{X} \left(\frac{1}{1 + \frac{v^2}{2\sigma^2 N}} \right)^N \partial v \tag{D.5}$$

with N currently hard coded to $\frac{20}{3}$, due to implementation issues (the author uses a look up table and chooses this value for simplicity), and K chosen such that $p(x < +\infty) = 1$. The probability of a residue value being between two values, X_1 and X_2, is given by

$$p\left(X_1 \leq x < X_2\right) = p(x < X_2) - p(x < X_1) \tag{D.6}$$

The parameter σ is estimated from the sigma-predictor of 30 causal neighbors in a Manhattan distance window (distance between two points measured along the axes at right angles). Therefore

$$\sigma^2 = \sum_{i=0}^{30} v_i * \text{pixel\_error}_i^2 \tag{D.7}$$

where $\text{pixel\_error}_0 = 1$, and v_is are model parameters, and are included in the first part of the encoded message (see more details in [119]).

Since the value for this probability may not be trusted, a “certainty” or “trust” parameter (c) is also sent in the message part of the file, in order to blend the probability distribution ($p(x)$) with a distribution that represents total ignorance:

$$\hat{p}(x = X) = c * p(x = X) + \frac{(1-c)}{L} \tag{D.8}$$

where L is the size of possible range of values for x.

At the coding part, the probability that is used for encoding the prediction error on the arithmetic encoder is the blending of all the predictors probability errors:

$$p_{\text{all}}(x = X) = \sum_{j=1}^{P} b_j \hat{p}_j(x = X) \tag{D.9}$$

where the blending weight b_j can be interpreted as the estimation of number of bits that predictor j would require to encode the current pixel, based on the number of bits required to encode the causal neighborhood.

$$b_j = \frac{c_j}{\sum_{k=1}^{M} c_k} \tag{D.10}$$

and

$$ln(c_j) = \sum_{k=0}^{Q} t_k ln(\hat{p}_j(x_k = pixel_k)) \tag{D.11}$$

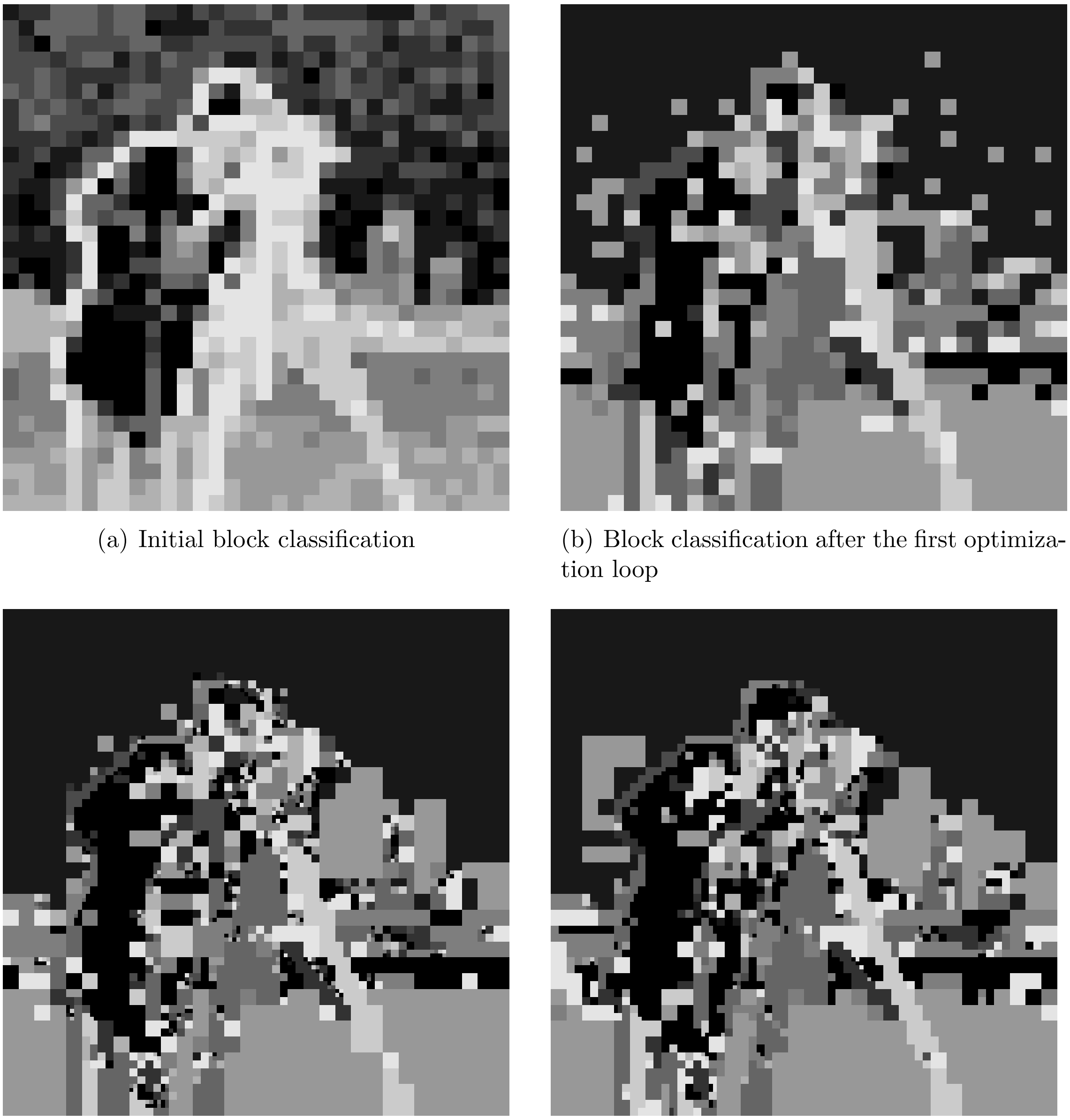
where t_k are model parameters determined during the image analysis phase. A fast implementation of the TMW algorithm, GLICBAWLS (Grey Level Image Compression by Adaptive Weighted Least Squares [120]), was also made available online and will be used for comparison. This algorithm combines the compression rate equivalent to the rates from complex algorithms with a moderate computational complexity.

EDP: Edge-directed prediction was proposed in [60], and uses the LSP procedure for the prediction step (see Appendix C). The good results achieved by this encoder rely on the excellent performance of the predictor, in comparison with other prediction schemes, like CALIC and JPEG-LS, that do not have a flexible predictor like the EDP algorithm has. Compared to other methods also based on linear prediction, such as the TMW algorithm, EDP compression gains are smaller, but this encoder does not perform a two-pass encoding. The gains for the LSP are conditioned to the training area, and local statistics of the image. The residue encoder separates the prediction residue in bit planes, and entropy encode the residue using context modeling based on local error variance, bias cancelation and also residue remapping.

MRP: MRP (Minimum-rate predictors [121]) is a state-of-the-art lossless codec that achieves the best compression rates for most of the tested images. This lossless compression method implements several sophisticated prediction and entropy coding schemes that are optimized in a multi-pass scheme.

The encoder is also based on linear prediction, and the choice of predictors are first refined with a multistage image analysis, minimizing the overall bitrate. The residue is entropy encoded using a range coder [122], based on the implementation available on [123], or just a simple Huffman coder.

The image is divided into blocks of 8×8 pixels, and every 8x8 block is first classified according to its variance. The blocks organized in classes will be used as training sequences for designing linear predictors. The pixels of the same class are considered realization of an auto-regressive process, and the linear coefficients are obtained using the Yule-Walker equations [107]. Once the coefficients are determined, the prediction for the image can be calculated. Then the cost for residue encoding is determined, considering a conditional probability density function modeled by generalized Gaussian functions [124]. The key element of MRP is actually choosing the best model to fit the data



(a) Initial block classification

(b) Block classification after the first optimization loop

(c) Block classification with different block sizes

(d) Final block classification

Figure D.6: Block classification optimization for cameraman image. Each luminance value represents one defined class, that is, one specific linear predictor. In the first optimization loop, only 8×8 blocks are used. In the second optimization loop, block sizes from 32×32 up to 4×4 are allowed.

to be encoded, in a set of very different probability distributions, all modeled by the generalized Gaussian distribution.

The context for the residue's conditional probability is determined by the quantization of the weighted sum of neighboring residues. As the blocks are being classified, the context quantization thresholds are optimized targeting the rate reduction via dynamic programming. With the new prediction coefficients and context quantization levels, the blocks are reclassified, and the optimization is performed until a satisfactory compression level is accomplished or the maximum number of iterations is achieved.

After this first optimization loop, the fixed block dimension is relaxed, and from this point on block sizes can vary from 32 × 32 up to 4 × 4. A quad-tree segmentation structure is used for block division, and once again the optimization loop will refine the prediction coefficients, context thresholds and block classification. Notice that this process might even lead to the reduction of the available classes. Figures D.6(a) to D.6(d) show the classification of image blocks for different stages in the optimization loops. Notice in this example that, by the end of the optimization loop, the number of classes used are reduced, and the same predictor is used in smooth areas, with larger block sizes, as can be seen at the background behind the cameraman in Figure D.6(d).

Other methods Other methods for lossless image compression are methods used for bi-level images, but can also be used for gray scale images, such as JBIG [4] e o JBIG2 [5], dictionary-based methods, such as the image formats GIF [125] and PNG [126], and the wavelet-based method JPEG2000 [6].

The JBIG standard (Joint Bilevel Image Experts Group) is used for lossless compression of bi-level images and was initially designed for fax transmission and document storage and distribution. For gray scale images, JBIG divides the pixel values into bit planes, and encode each bit plane as a bi-level image. The evolution of JBIG standard, JBIG2 [127–129], uses dictionary techniques to encode parts of the image, usually text, while smooth image areas are coded with an arithmetic or a Huffman encoder [108, 115].

The GIF image format was widely used for image transmission at the beginning of the internet. It is based on the LZW algorithm [37], which uses recurrent patterns from the image to update its dictionary, coding their symbols according to their occurrence frequency. This method is particularly effective for computer images, where all pixels in a smooth region have the same value.

The PNG format [126] was created in order to avoid the licensing issues of the GIF file format. It uses only license-free encoding techniques, such as a variation of the LZ77 method, known as *deflate* [130]. The format also has support for color and transparency. Details on the algorithm can be found in [108].

A different paradigm for image compression is the use of transform instead of prediction. JPEG2000 [6] uses wavelet transform, and does not suffer from the causality issues of the prediction based encoders. It also has an interesting scalability property, where the same bitstream is able to describe the image with different levels of accuracy, providing lossy and lossless compression in a single bitstream. More details can be found in the literature [6] or online [131].

D.3 Experimental results and benchmark

This section presents the encoding results for several lossless compression algorithms, some of them considered to be the state-of-the-art in lossless compression. Table D.1 summarizes the results of lossless compression of two groups of images: smooth images and compound images [1].

The chosen lossless encoders are:

- JPEG-LS[2], standard for lossless image compression, has an excellent performance for smooth images and also a good performance for compound images.
- CALIC[3] is more efficient than JPEG-LS due to a better prediction. However, it shows a higher computational cost.
- GLICBAWLS[4], fast implementation of the TMW algorithm, usually outperforms CALIC, but the predictors are even more complex.
- EDP[5] is a lossless encoder based on LSP prediction mode, recently incorporated into the MMP algorithm (see Appendix C).
- MRP[6]is an algorithm with high computational complexity, due to predictor optimization and residue coding. It achieves the best results in the literature for lossless compression.
- JBIG[7] is usually used for coding bi-level images, but can also be used for coding gray-scale images.

[1] all images can be found in the Appendix K

[2] Results obtained with software [132], using the command `./locoe -i<input>.pgm -o<output>.loco`

[3] Results obtained with software [133], using the command `./enCALICh <input>.pgm width height depth > <output>.calic`

[4] Results obtained with software [134], using the command `./glicbawls <input>.pgm <output>.glic`

[5] Results obtained with software [135], using the command `./edp <input>.pgm`

[6] Results obtained with software [136], using the command `./encmrp -o <input>.pgm <output>.mrp`

[7] Results obtained with software [137], using the command `./pnmtojbig <input>.pgm <output>.jbig`

- GIF[8] is an algorithm based on pattern matching and has a good performance for compound images.
- PNG[9] is an algorithm that is also based in pattern matching, however it does not have the same licensing issues as GIF.
- JPEG2000[10] is a state-of-the-art lossy image encoder that presents an hierarchical coding structure (from lossy to lossless, in the same bitstream), therefore it can also be used for lossless compression.

D.4 Theoretical bounds on MMP's lossless performance

In this section we show how the MMP algorithm can asymptotically achieve the entropy rate of any stationary, ergodic, memoryless source with finite alphabet, following a similar chain of ideas to the ones developed for the Grammar Codes in [139]. The results are derived first for the MMP algorithm presented in [90], without considering the prediction step and using a dyadic block segmentation. Later on, we show how prediction can contribute to improve its convergence rate.

Let us consider a two-dimensional data input x, with symbols from an alphabet $\mathcal{A}$. It is first divided into L blocks of size $N \times N$ (where the total number of image pixels is $|x| = n = L * N^2$). Then each block will be encoded by the MMP algorithm, where an $N \times N$ pattern gets converted into a sequence of flags and indexes, dependent on MMP's dictionary.

The dictionary used for coding is the union of two sets: $\mathcal{D} = \mathcal{S} \cup \mathcal{A}$. The set $\mathcal{A}$, the pixel's alphabet, contains 1×1 patterns, while the set $\mathcal{S}$ contains all the other multiscale patterns, different from 1×1 ones. Set $\mathcal{A}$ is initialized with all possible 1×1 patterns (for example, for an 8-bit gray-scale image, all values from 0 to 255), while set $\mathcal{S}$ is initialized with patterns generated from scale transformation of some elements present in set $\mathcal{A}$. This initialization may generate in $\mathcal{S}$ a non-uniform distribution of constant-value patterns. If we consider a dyadic block-segmentation, then we have a total of $2\log(N) + 1$ scales (including the 1×1 scale). The log operation stands for logarithm with base 2. The same base will be used throughout this deduction, and therefore it is suppressed for clarity reasons. Then, the set $\mathcal{S}$ is

[8] Results obtained with software [138], using the command `./convert <input>.pgm <output>.gif`

[9] Results obtained with software [138], using the command `./pnmtopng <input>.pgm > <output>.png` and optimized using the command `./pngcrush -brute <input>.png <output>.png`

[10] Results obtained with software [131], using the command `java JJ2KEncoder -i input.pgm -o output.j2k -lossless on`

Table D.1: State-of-the-art lossless compression algorithms results for smooth and compound images. Results are given in bits-per-pixel. Best results are highlighted in bold.

Images	**JPEG-LS**	**CALIC**	**GLICBAWLS**	**EDP**	**MRP**	**JBIG**	**GIF**	**PNG**	**JPEG2000**
Smooth Images									
airplane	3.817	3.819	3.668	4.002	**3.591**	4.716	6.546	4.220	4.047
baboon	6.037	6.031	5.666	5.981	**5.663**	6.983	9.360	6.210	6.146
balloon	2.904	2.959	2.640	3.078	**2.579**	3.582	6.315	3.238	3.060
barb	4.691	4.525	3.916	4.692	**3.815**	5.556	8.631	5.199	4.639
barb2	4.686	4.613	4.318	4.332	**4.216**	5.586	8.740	5.132	4.827
camera	4.314	4.245	4.208	4.396	**3.949**	4.816	6.769	4.674	4.596
couple	3.699	3.696	3.543	3.831	**3.388**	4.332	6.599	4.061	3.970
goldhill	4.477	4.493	4.276	4.564	**4.207**	5.296	7.690	4.662	4.639
lena	4.607	4.579	4.295	4.591	**4.280**	5.483	8.501	4.888	4.718
lennagrey	4.238	4.174	3.901	4.233	**3.889**	5.096	8.072	4.577	4.336
noisesquare	5.683	5.788	5.415	5.699	**5.270**	5.709	6.970	5.671	5.685
peppers	4.513	4.550	4.246	4.584	**4.199**	5.306	8.236	4.887	4.665
shapes	1.214	1.542	2.291	2.555	**0.685**	1.364	2.480	1.154	1.954
Average	4.222	4.232	4.029	4.349	**3.825**	4.910	7.301	4.506	4.406
Compound Images									
pp1205	4.472	4.188	5.435	5.319	4.039	4.948	2.897	**2.550**	5.040
pp1209	4.616	4.315	5.275	5.085	3.903	5.001	3.103	**2.785**	3.854
scan_002	**3.429**	3.446	4.204	4.145	3.029	4.189	5.250	3.729	3.996
scan_004	2.370	2.560	3.558	3.819	**2.041**	3.446	3.023	2.404	2.748
scan_006	3.202	3.461	4.589	4.816	**2.844**	4.205	3.954	3.037	3.578
Average	3.618	3.594	4.612	4.637	3.171	4.358	3.645	**2.901**	3.843

initialized with less than $2\log(N) * |\mathcal{A}|$ elements, since not all 1×1 patterns (e.g., non-uniformly spaced levels) may be used for pattern generation. During encoding, the dictionary grows with patterns generated from multiscale transformations of concatenations of reconstructed codewords.

Each block is coded with a sequence of flags and dictionary indexes, that are encoded by an arithmetic encoder. Supposing zero-order contexts [108] are used, the rate spent to encode each symbol is given by:

$$\text{rate}(\beta_i) = -\log \frac{c(\beta_i)}{\sum_{\beta \in \mathcal{D}} c(\beta)} \tag{D.12}$$

where $c(\beta_i)$ stands for the frequency counter for symbol β_i of dictionary $\mathcal{D}$.

All the counters $c(\beta_i)$ are initialized with zero, except the ones that account for the patterns that are already present in the dictionary, which are initialized with one. Once the symbol β_i is used, its counter is updated by adding 1; then if a new symbol is inserted in the dictionary, its counter increases from 0 to 1. In addition, we assume that a block is partitioned until a perfect match is found. The dictionary is then updated after the encoding of the block $N \times N$ finishes.

Suppose that, for the first block, we have to divide it in such a manner that n_1 indexes are used. So the rate spent for encoding the first block is given by:

$$\begin{aligned} \text{rate}(\text{block}_1) &= \text{rate(flags)} + \sum_{i=1}^{n_1} -\log \frac{c(\beta_i)}{\sum_{\beta \in \mathcal{D}} c(\beta)} \\ &= \text{rate(flags)} + \sum_{i=1}^{n_1} \log \frac{|\mathcal{D}| + i}{c(\beta_i)} \\ &\leq \text{rate(flags)} + \sum_{i=1}^{n_1} \log \frac{|\mathcal{A}| * (2\log(N) + 1) + i}{c(\beta_i)} \end{aligned} \tag{D.13}$$

After encoding the first block, the dictionary grows with $(n_1 - 1)$ concatenations of elements, and its expansions and contractions, which leads to the addition of at most $(n_1 - 1) * (2\log(N))$ new elements, considering all scales. Assuming that for the second block we have n_2 matches, and following a similar development, we can calculate the rate spent for encoding the second block, that is given by:

$$\begin{aligned} \text{rate}(\text{block}_2) \quad \leq \quad & \text{rate(flags)}+ \\ \sum_{i=n_1+1}^{n_2+n_1} \log & \frac{(|\mathcal{A}| + (n_1 - 1)) * 2\log(N) + |\mathcal{A}| + i}{c(\beta_i)} \end{aligned} \tag{D.14}$$

Therefore the total rate ($n * r^{MMP}(x)$, where $r^{MMP}(x)$ is the compression rate in bits per pixel and n is the number of pixels) spent for encoding the input image

x, is the sum of the rate spent for each block, given by:

$$\begin{aligned} n * r^{MMP}(x) &= \text{rate}(\text{block}_1) + \cdots + \text{rate}(\text{block}_L) \\ &\leq \text{rate}(\text{flags}) + \sum_{i=1}^{n_{\text{total}}} \log \frac{|\mathcal{A}| + j_{total} + i}{c(\beta_i)} \end{aligned} \tag{D.15}$$

where $n_{\text{total}} = n_1 + n_2 + \cdots + n_L$ and

$$j_{\text{total}} \leq (|\mathcal{A}| + n_{\text{total}} - L) * 2\log(N) \tag{D.16}$$

In this development, n_{total} represents the number of indexes used to encode the image, j_{total} is the number of new elements introduced in the dictionary due to the concatenation of the encoded codewords and the multiscale update routine, and L is the total number of $N \times N$ blocks in the image. Similarly to what is done in [139], we divide the rate into two terms, one that accounts for the growth of the dictionary and another that expresses the unnormalized empirical entropy of the symbols, yielding

$$\begin{aligned} n * r^{MMP}(x) &\leq \text{rate}(\text{flags}) + \log \prod_{i=1}^{n_{\text{total}}} \frac{|\mathcal{A}| + j_{\text{total}} + i}{c(\beta_i)} \\ &\overset{(1)}{\leq} \text{rate}(\text{flags}) + \log \frac{n_{\text{total}}!}{\prod_{i=1}^{n_{\text{total}}} c(\beta_i)} + \\ &\quad \log \binom{|\mathcal{A}| + j_{\text{total}} + n_{\text{total}}}{|\mathcal{A}| + j_{\text{total}}} \\ &\overset{(2)}{\leq} \text{rate}(\text{flags}) + \log \frac{n_{\text{total}}!}{\prod_{i=1}^{n_{\text{total}}} c(\beta_i)} + \\ &\quad |\mathcal{A}| + j_{\text{total}} + n_{\text{total}} \\ &\leq \text{rate}(\text{flags}) + \sum_{\beta \in \mathcal{D}} (c_\beta) \log \frac{n_{\text{total}}}{c_\beta} + \\ &\quad |\mathcal{A}| + j_{\text{total}} + n_{\text{total}} \\ &\leq \text{rate}(\text{flags}) + H_p(x) + |\mathcal{A}| + j_{\text{total}} + n_{\text{total}} \end{aligned} \tag{D.17}$$

where steps (1) and (2) follows from $\prod_{i=1}^{n}(a+i) = \binom{a+n}{a} n!$ and the inequality on the size of a type class, see [61]. In this reasoning, c_β is the number of times the element β appears in the sequence x, and $\frac{H_p(X)}{n_{\text{total}}}$ is the zero-order entropy of the sequence of indexes, since element β has a probability of $\frac{c_\beta}{n_{\text{total}}}$.

Now we derive an upper-bound for the above mentioned zero-order entropy by assuming that the same sequence will be encoded coordinate-by-coordinate using a k-context arithmetic encoder. Since the dictionary is multiscale, we have that

$\beta_i = y_1 y_2 \cdots y_l \in \mathcal{A}^l$. Then the probability of each codeword, as given for the hypothetical arithmetic encoder, is given by the formula [139]:

$$p^*(\beta_i) = Q_k k^{-1} r^{-2} \max_{c_0 \in \mathcal{C}} \sum_{c_1 c_2 \cdots c_r \in \mathcal{C}} \prod_{i=1}^{l} p(y_i, c_i | c_{i-1}) \tag{D.18}$$

where c_i are the possible contexts of the arithmetic encoder and Q_k is a constant value so that p^* is a probability distribution on $\mathcal{A}$. More details on the development of the k-context probability can be found in [139].

From the information inequality formula [61], we relate both entropies in the following manner:

$$H_p(x) \leq - \sum_{i=1}^{n_{\text{total}}} \log p^*(\beta_i) \tag{D.19}$$

Now substituting the value of p^* indicated in Equation (D.18), and considering that $Q_k < 1/2$ and $l \leq N^2$, we get

$$H_p(x) \leq n_{\text{total}} * (1 + \log(k) + 2\log(N * N)) + n r_k^*(x) \tag{D.20}$$

where $r_k^*(x)$ is the rate using a k-context arithmetic encoder for each letter of the sequence x, given by the log of the right-hand side of equation (D.18).

Now we define $R_{n,k}^{MMP}$ as the difference between the rate given by the MMP algorithm and the rate of the k-context arithmetic encoder previously defined. We have

$$\begin{aligned} R_{n,k}^{MMP} &= r^{MMP} - r_k^*(x) \\ &\leq \frac{\text{rate(flags)} + H_p(x) + |\mathcal{A}| + j_{\text{total}} + n_{\text{total}}}{n} - \\ &\quad \frac{H_p(x) - n_{\text{total}} * (1 + \log(k) + 2\log(N * N))}{n} \\ &\leq \frac{\text{rate(flags)} + n_{\text{total}} * (C) + |\mathcal{A}| + j_{\text{total}}}{n} \end{aligned} \tag{D.21}$$

where C is a constant value defined as $2 + \log(k) + 2\log(N * N))$.

It is known that, as the length of the sequence grows, the rate of the k-context arithmetic encoder approximates the entropy of the source [139]. So if we analyze how close we are from this rate, we can also give a bound on how close the MMP algorithm approaches the source's entropy, once the length of the sequence grows. As can be seen from equation (D.21), the performance bound for the MMP may approach the source's entropy depending on the behavior of the total number of indexes used and the size of the dictionary. So we need to analyze the terms $\frac{n_{\text{total}}}{n}$, $\frac{j_{\text{total}}}{n}$ and $\frac{\text{rate(flags)}}{n}$ as $n \rightarrow \infty$.

Let us then express the average number of indexes used to encode a block i

$(\bar{n}_i)$, according to a given probability of match at scale $(2^{\lfloor\frac{l+1}{2}\rfloor}, 2^{\lfloor\frac{l}{2}\rfloor})$ (we will use the notation $p_{(l,i)}$ for the probability of match at scale l of the i-th block, considering $l \in \{2\log(N), 2\log(N)-1, \cdots, 1, 0\}$). Notice that the number of indexes doubles as we decrease the scale by one.

$$\bar{n}_i = p_{(2\log(N),i)} + \sum_{m=0}^{2\log N-1} p_{(m,i)} \prod_{k=m+1}^{2\log N} 2(1-p_{(k,i)}) \tag{D.22}$$

By applying Equation (D.22) to every block, the average total number of indexes used $(\bar{n}_{\text{total}})$ is given by the sum of the average number of encoded indexes for each block, that is

$$\begin{aligned}
\bar{n}_{\text{total}} &= \bar{n}_1 + \cdots + \bar{n}_L \\
&= \sum_{i=1}^{L} p_{(2\log(N),i)} + \sum_{i=1}^{L} \sum_{m=0}^{2\log N-1} p_{(m,i)} \prod_{k=m+1}^{2\log N} 2(1-p_{(k,i)})
\end{aligned} \tag{D.23}$$

It is reasonable to state that $p_{(k,i)} \leq p_{(l,i)}$, if $k > l, \forall i$, that is, it is less probable to find a match at larger scales. Knowing that $p_{(k,i)}$ is a probability $(p_{(k,i)} \leq 1, \forall k, i)$, and that $p_{(0,i)} = 1$ (since for the lossless case, the dictionary at scale 1×1 is complete, so we will always have a match at this scale), we can derive an upper-bound for $\bar{n}_{\text{total}}$, given by

$$\begin{aligned}
\bar{n}_{\text{total}} &\leq \sum_{i=1}^{L} 1 + \sum_{i=1}^{L} \sum_{m=0}^{2\log N-1} p_{(0,i)} \prod_{k=m+1}^{2\log N} 2(1-p_{(2\log N,i)}) \\
&\leq L + \sum_{i=1}^{L} \sum_{m=0}^{2\log N-1} \prod_{k=m+1}^{2\log N} \underbrace{2(1-p_{(2\log N,i)})}_{A(i)} \\
&\leq L + \sum_{i=1}^{L} \sum_{m=0}^{2\log N-1} A(i)^{2\log N-m} \\
&\leq L + \sum_{i=1}^{L} \frac{A(i)^{2\log N+1} - A(i)}{A(i)-1}
\end{aligned} \tag{D.24}$$

Now assuming that the source is memoryless, it can be shown (the proof can be found in the Section D.4.1) that the limit for $\frac{n_{\text{total}}}{n}$ can be given by

$$\begin{aligned}
\lim_{n\to\infty} \frac{n_{\text{total}}}{n} &= \lim_{n\to\infty} \left(\frac{L}{n} + \frac{1}{n} \sum_{i=1}^{L} \frac{A(i)^{2\log N+1} - A(i)}{A(i)-1} \right) \\
&= \frac{1}{N^2}
\end{aligned} \tag{D.25}$$

From Equation (D.25) and Equation (D.16), we can also derive

$$\lim_{n\to\infty} \frac{j_{\text{total}}}{n} = 0 \tag{D.26}$$

This is reasonable because, after encoding a large number of blocks, the dictionary has almost all the probable patterns of the image, so it stops growing and becomes infinitely small when compared to the sequence's size. Likewise, the flags encoding process will have a similar behavior, since a match will occur in the largest scale, avoiding further segmentation and the further use of flags, and then we also have that

$$\lim_{n\to\infty} \frac{\text{rate(flags)}}{n} = 0 \tag{D.27}$$

Then from equation (D.21), we have that

$$\lim_{n\to\infty} R_{n,k}^{MMP} = \frac{1}{N^2} \tag{D.28}$$

We can see that the convergence is dependent on the block size. So the larger the block size, the closer we are to the entropy rate, that is, we approximate the entropy rate by increasing the block size. Then, for the MMP, we can make the block size as large as needed, to achieve an encoding rate as close as desired to the source's entropy.

Notice that by using the scale transformation, we actually increase the chances of a match, since we have more patterns on the dictionary. So the benefit of using scale transformations can be perceived in the transitory convergence of the algorithm, which is faster than the one of the algorithm without update at different scales.

For the convergence proof above we use the fact that the source is memoryless. This is not true for images, but may be approximately accomplished by using prediction. If the prediction step is successful in inferring the value of the next block of pixels, it will then decorrelate the source, and as we have seen from the convergence point of view, we converge to the source's entropy.

Table D.2 shows some compression results for the MMP without using prediction. The results for the PNG algorithm [126] were also displayed for comparison. We can see that MMP performs well for images where prediction cannot fully decorrelate the image, such as scanned documents (e.g. pp1205 and scan_004), but for smooth images MMP is still outperformed by PNG, since it does not profit from the prediction step. This indicates the importance of the prediction step also for lossless encoding. In the next section we investigate ways to enhance prediction for the MMP in the lossless case.

Table D.2: MMP's lossless compression performance (without prediction). Results in bits-per-pixel. Best results are highlighted in bold.

Images	MMP	PNG
airplane	5.403	**4.220**
barb	6.331	**5.199**
lennagrey	6.043	**4.577**
pp1205	**2.385**	2.550
scan_004	**2.285**	2.404
shapes	1.841	**1.154**

D.4.1 Proof of the convergence of $\lim_{n\to\infty}\frac{n_{\text{total}}}{n}$

In this subsection, we will prove Equation (D.25), repeated here for convenience.

$$\lim_{n\to\infty}\frac{n_{\text{total}}}{n} = \lim_{n\to\infty}\left(\frac{L}{n}+\frac{1}{n}\sum_{i=1}^{L}\frac{A(i)^{2\log N+1}-A(i)}{A(i)-1}\right) \tag{D.29}$$

The first term on the right-hand side generates the result $\frac{1}{N^2}$, since $n = L * N^2$. This term can be identified as the steady-state rate of the encoder, dependent on the largest scale, that is, on the chosen block size. The second term can be identified as a transitory behavior of the encoder, and is linked to the smaller scales. We see that the convergence depends on the matching probability, which in turn depends on the source's probability distribution. Consider a memoryless source X and a word of size s (i.e., $x \in \mathcal{A}^s$). Suppose that k segments of size s have been encoded. After encoding the k segments, the k encoded segments will be added to the dictionary. Then, the matching probability of x is

$$\begin{aligned} Pr(x \in \mathcal{D}) &= Pr\{x \text{ occurred at least once in } k \text{ segments}\} \\ &= 1 - Pr\{x \text{ did not occur in } k \text{ segments}\} \end{aligned} \tag{D.30}$$

Notice that the number of segments k for a sequence of size n is given by $k = \frac{n}{s}$. Since we divide the image into blocks of size $N \times N$, the same correspondence can be made taking the block size into account, that is $k = \frac{i*N^2}{s}$, after encoding i blocks ($n = i * N^2$). Considering all possible values for the input x, and the fact that X is memoryless, we can rewrite the matching probability for scale l after encoding i blocks, in the following manner

$$p_{(l,i)} = \sum_{x\in\mathcal{A}^{2^l}}[1-(1-p(x))^{\frac{i*N^2}{2^l}}]p(x) \tag{D.31}$$

where $p(x)$ is the probability of the sequence x.

We have that, as the sequence grows, the matching probability for any scale tends to 1. This is so because the dictionary contains most of the possible patterns of the image in all scales. Then, we have that

$$\lim_{n\to\infty} \sum_{x\in\mathcal{A}^{2^l}} [1-(1-p(x))^{\frac{n}{2^l}}]p(x) = 1 \tag{D.32}$$

This implies that the term $A(i) = 2(1 - p_{(2\log N,i)})$ of the summation in Equation (D.29) has the following properties:

(i) $A(i+1) < A(i)$

(ii) $\lim_{i\to\infty} A(i) = 0$

Considering the general term of the series in Equation (D.29)

$$r(i) = A(i)\frac{A(i)^{2\log N} - 1}{A(i) - 1} \tag{D.33}$$

we can show that this is a convergent series by applying the ratio test, that is, the series is convergent if

$$\lim_{i\to\infty}\left|\frac{r(i+1)}{r(i)}\right| < 1 \tag{D.34}$$

Therefore, we have

$$\begin{aligned}
&\lim_{i\to\infty}\left|\frac{A(i+1)}{A(i)}\frac{A(i+1)^{2\log N}-1}{A(i)^{2\log N}-1}\frac{A(i)-1}{A(i+1)-1}\right| \\
&= \lim_{i\to\infty}\left|\frac{A(i+1)}{A(i)}\right|\left|\frac{A(i+1)^{2\log N}-1}{A(i)^{2\log N}-1}\right|\left|\frac{A(i)-1}{A(i+1)-1}\right| \\
&= \lim_{i\to\infty}\left|\frac{A(i+1)}{A(i)}\right| < 1
\end{aligned} \tag{D.35}$$

Since this is a convergent series, the limit of this series divided by the sequences' size (n) tends to zero, and thus $\lim_{n\to\infty}\frac{n_{\text{total}}}{n} = \lim_{n\to\infty}\frac{L}{n} = \frac{1}{N^2}$, which concludes our proof.

D.5 Proposal for enhancing MMP lossless image compression performance

In order for MMP to perform lossless compression, some pre-requisites must be fulfilled by the algorithm. MMP must be able to provide at least one zero-distortion representation of the image, even if the representation does not provide compression. Since MMP uses a set of multidimensional dictionaries, a requirement for lossless coding is that the 1×1 dictionary is complete, that is, it contains all possible values for a pixel residue. In case a matching in higher scales cannot be done, due to the completeness of the 1×1 dictionary, the lossless representation of the input image is assured.

MMP multidimensional dictionary can be considered also as an overcomplete basis, that is, there are many possibilities of representing the input signal using elements of the dictionary. MMP uses a greedy approach for choosing the codewords that will represent the input, where the encoding choices are made considering the block only, and not the whole image. The choices are based on a Lagrangian cost criteria, where the λ factor determines the relationship between the accepted distortion and the required bitrate. When setting this factor to zero, only the distortion will be considered, and the optimization function will eventually choose one of the zero-distortion representation for the input signal. Nevertheless, there might be more than one possible zero-distortion representation. Therefore, the rate-distortion optimization algorithm was also modified to choose the codewords that minimize the bitrate, in case the lossless distortion was chosen (that is, in case λ factor of the Lagrangian cost is zero).

Both of the above mentioned pre-requisites were implemented in the MMP algorithm, without any rate-distortion losses for the lossy version. The complete 1×1 dictionary is rarely used for lossy coding, and the new criteria of minimum rate for the codeword choice just moved the rate-distortion point over the convex hull curve towards less bits. In the next subsections we will propose some modifications in the image modeling stage of the algorithm, that is, in the prediction stage, and also in the residue coding stage, based on common techniques used in lossless image compression algorithms.

D.5.1 Study on the effects of prediction for MMP lossless coding

A common technique for lossless image compression is to apply prediction, in order to decorrelate the neighboring pixels and obtain a skewed high-peaked distribution for the residue. Higher peaked distribution can be efficiently entropy encoded using a variable-length entropy encoder, such as Huffman encoder or arithmetic coding. Therefore, many methods for image prediction targeting lossless compression have been proposed [140, 141].

Lossless image prediction models

Some of the most well-known prediction methods were incorporated in the MMP framework. Three lossless predictors were chosen: MED, used in JPEG-LS [66], GAP, used in CALIC [109], and the least-square adaptive predictor used in EDP [60]. MED is a simple, but effective predictor for smooth images. GAP is more complex than MED, however it is able to better capture characteristics of the surrounding texture and provides a more efficient prediction. Nevertheless, it still uses a fixed set of predictors for image modeling. LSP does not use fixed predictor coefficients, instead it adapts the prediction coefficients using a training window. This results in a more efficient prediction than the former mentioned algorithms.

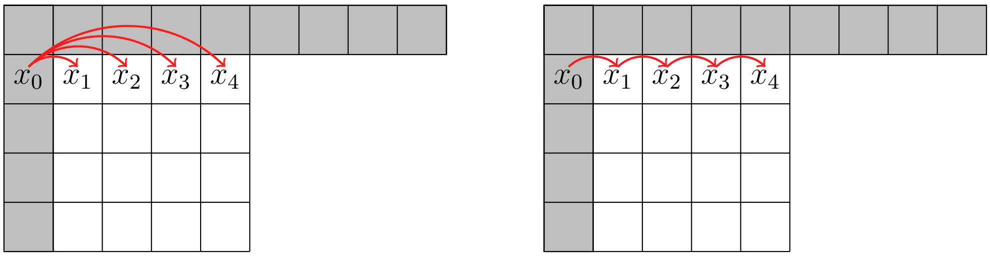
All three methods tested will replace the intra block prediction of the MMP algorithm, and perform first prediction in the whole image, using the raster scanning order. Then the residual image will be entropy coded using the MMP algorithm. No prediction mode information will be sent, since only one predictor is used. As MMP does not rely on the smoothness constraint, its residue coding might be able to cope with the inefficiencies of using only one single predictor.

For a more complete comparison, we also provide results for MMP with no prediction at all, and with the MMP-FP algorithm, that uses the intra prediction modes of the H.264/AVC algorithm (see Appendix B). Notice that the intra-prediction modes are not constrained to a half non-symmetric plane, like the other predictors, and might profit from the use of a blocking prediction structure. Since perfect reconstruction is obtained with lossless coding, the block prediction modes can also be modified to use prediction neighborhood closest to the position to be predicted, therefore enhancing the intra prediction capability. In the next subsection, a more detailed explanation of enhanced intra block prediction modes for lossless compression is given.

Enhanced intra prediction modes

High-performance lossless image coding algorithms usually perform image encoding on a pixel-by-pixel basis, using a small set of prediction neighbors. The H.264/AVC lossless video coding standard adopted a modification to its horizontal and vertical Intra prediction modes to enhance compression performance [63]. For the horizontal and vertical prediction modes, the closest pixels are used for prediction, instead of the ones adjacent to the upper and left block edges. This results in better coding efficiency, since an adjacent sample to the pixel to be predicted is usually a better prediction than a pixel closer to the block edge. Figure D.7 depicts the adaptation for the horizontal prediction mode. This procedure is similar to a DPCM (Differential Pulse Code Modulation) procedure [63].

In the lossless compression case, MMP also adopts this intra residual DPCM process for the horizontal and vertical prediction modes, therefore the MMP algorithm that uses the adapted prediction modes for lossless coding will be called MMP-DPCM. With this modification, the algorithm tends to use larger block-sizes



(a) Lossy compression

(b) Lossless compression

Figure D.7: Adaptation for the horizontal mode in the lossless case. In previous implementation, the neighboring block provides the value for prediction, which is spatially distant from some elements of the block. By using the residue information, both encoder and decoder are able to update the last predicted value with the original value. Then for the next position, the closest original value is available for prediction and can be used instead.

in the prediction step, saving bits associated with the prediction information. Furthermore, such enhanced prediction favors the narrowing of the residue's histogram, which also helps the dictionary adaptation process [21].

The same concept can also be applied to the LSP mode, that can use the original pixel values for the training instead of the predicted values, as was done for the lossy version (see Appendix C). At the decoder side, the LSP prediction needs to be performed after receiving all the codewords related to the residue of the block prediction. Then for each pixel position, where the LSP is optimized, the predicted value can later be added to the residue, in order to reconstruct the original value.

Experimental results

Table D.3: Lossless compression results for several different prediction proposals, given in bits-per-pixel. Best results are highlighted in bold.

Images	**NO PRED**	**MED**	**GAP**	**LSP**	**MMP-FP**	**MMP-DPCM**
Barbara	6.331	5.452	5.339	4.451	4.519	**4.352**
Cameraman	6.582	4.933	4.905	4.788	4.772	**4.570**
Gold	6.544	5.020	4.853	4.805	4.947	**4.505**
Lena	6.043	4.546	4.506	4.220	4.297	**4.212**
PP1205	**2.385**	3.421	4.745	5.715	2.809	2.736
PP1209	2.709	2.997	4.856	5.418	2.692	**2.567**
Shapes	1.841	1.104	1.458	2.174	1.056	**0.916**
Average	4.634	3.925	4.380	4.510	3.585	**3.408**

Table D.3 shows the results of compression achieved with the MMP algorithm, when using different prediction models. MED, GAP and LSP are suited for smooth images, and give higher compression results than using no prediction at all. However, for compound and computer generated images, the prediction cannot capture the texture of the image appropriately, and the algorithm that uses no prediction outperforms the proposed methods, achieving higher compression gains.

For the MMP algorithm that uses the prediction methods based on the H.264/AVC modes and LSP (the MMP-FP method), compression rates for smooth images are comparable to ones achieved by the LSP predictor. Moreover, compression of compound and computer images achieves a higher gain due to the use of the MFV prediction mode, providing evidence that the set of prediction modes available for the MMP-FP algorithm is appropriate for smooth and compound lossless image coding. Consistent gains were also reported by applying the DPCM modification, achieving the best compression results for the tested images. Therefore, the DPCM adaptation of the intra prediction modes was adopted by the lossless MMP algorithm.

D.5.2 Study of the effects of residue coding techniques for the lossless MMP

For the residue coding, MMP uses pattern matching with an adaptive dictionary updating scheme, which has proved to be very efficient when lossy coding smooth and compound images. Nevertheless, the context used for entropy coding is not related to the neighboring residual information, a common procedure adopted for several lossless image compression. Also techniques for improving the probability model used for entropy coding, such as histogram truncation or error feedback were not adopted by the MMP algorithm yet, since they are usually applied for pixel coding, and not for block coding.

Some of the techniques used for lossless pixel entropy coding are adapted for block coding and integrated into the MMP framework. The effectiveness of such techniques is also evaluated.

Histogram restriction

Due to the finite dimensionality of the input alphabet, when applying prediction only a limited range of possible residues are allowed. Then it is possible to remove the probability attributed to these forbidden values, once the prediction is known, and fine-tune the residue probability that is going to be used for residue coding. This technique is successfully applied in many lossless encoders, such as CALIC and MRP. This is similar to use a residue probability conditioned to the predicted value, that is, the residue histogram is restricted to values conditioned on the prediction value. The conditional probability of a residue e, given the predicted value x_{pred},

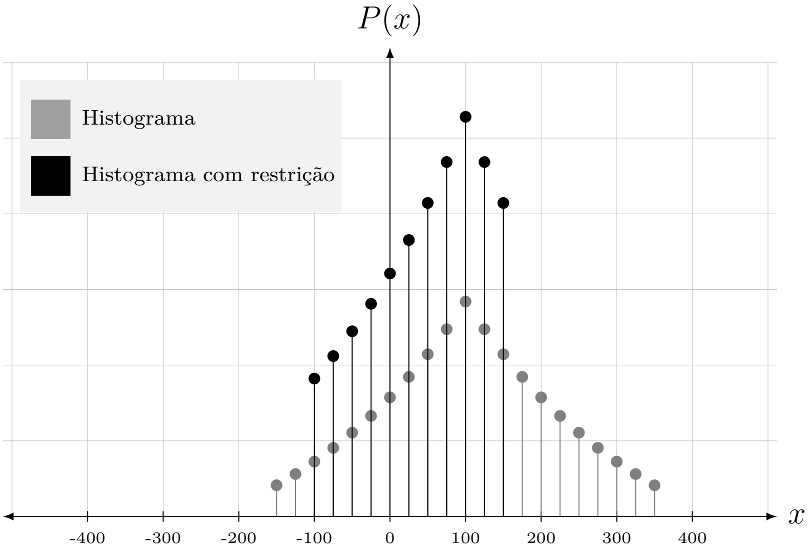


Figure D.8: Histogram restriction for a prediction value equal to $x_{\text{pred}} = 100$. In this example, a Laplacian distribution is used to illustrate the probability distribution of the residue. This model is widely used to represent prediction residue of smooth image areas.

can be obtained in the following manner.

$$\begin{aligned} Pr(e|x_{\text{pred}}) &= \frac{Pr(e, x_{\text{pred}})}{Pr(x_{\text{pred}})} && \text{(Bayes' rule)} \\ &= \frac{Pr(x_{\text{pred}}|e) * Pr(e)}{Pr(x_{\text{pred}})} \\ &= \frac{Pr(e)}{\frac{Pr(x_{\text{pred}})}{Pr(x_{\text{pred}}|e)}} \end{aligned} \tag{D.36}$$

Which is equivalent to state that the residue can only be found between values $-x_{\text{pred}}$ and $255 - x_{\text{pred}}$. Therefore, the probability of the prediction x_{pred} in Equation D.36 can be replaced by the probability of the error assuming values $-x_{\text{pred}} \leq e \leq 255 - x_{\text{pred}}$, and we have:

$$= \frac{Pr(e)}{\frac{Pr(-x_{\text{pred}} \leq e \leq 255 - x_{\text{pred}})}{Pr(-x_{\text{pred}} \leq e \leq 255 - x_{\text{pred}}|e)}} \tag{D.37}$$

Once the residue value e is known, we have that $Pr(-x_{pred} \leq e \leq 255 - x_{pred}|e) = 1$, and the conditional probability of Equation D.37 becomes:

$$= \frac{Pr(e)}{\sum_{x=0}^{255} Pr(x - x_{\text{pred}})} \tag{D.38}$$

which indicates that the probability of the forbidden residues assume a zero value and the remaining probabilities are rescaled, so that the sum of residue probabilities is equal to one. The process is also illustrated in Figure D.8.

The same principle was implemented for the MMP residue encoding. Given the block prediction value, the codewords of the dictionary that resulted in forbidden residue patterns assumed a zero value probability, in order to use a probability distribution that takes into account only the possible residue patterns. The new adjusted probability distribution is used by the arithmetic encoder for entropy coding of the chosen dictionary codeword. Notice however that the use of this technique requires the knowledge of the predicted value for residue encoding, which is the contrary of what is required by the DPCM technique, described in Section D.5.1. Therefore, this technique is mutually exclusive with the DPCM enhancement prediction method.

Residue remapping

An image with 8-bit pixel can have 256 different values. The difference between prediction and the actual pixel can assume values between -256 to 256, which needs 9 bits for its representation. However, as already stated, the possible values assumed by the residue are found in the interval $[-x_{\mathrm{pred}}, 255-x_{\mathrm{pred}}]$, that contains 256 values, and can be represented by 8 bits only. A common procedure used in lossless encoders, such as CALIC [65] or JPEG-LS [66] is the residue remapping. A typical residue remapping function for a residue $\delta = x - x_{\mathrm{pred}}$ is as follows:

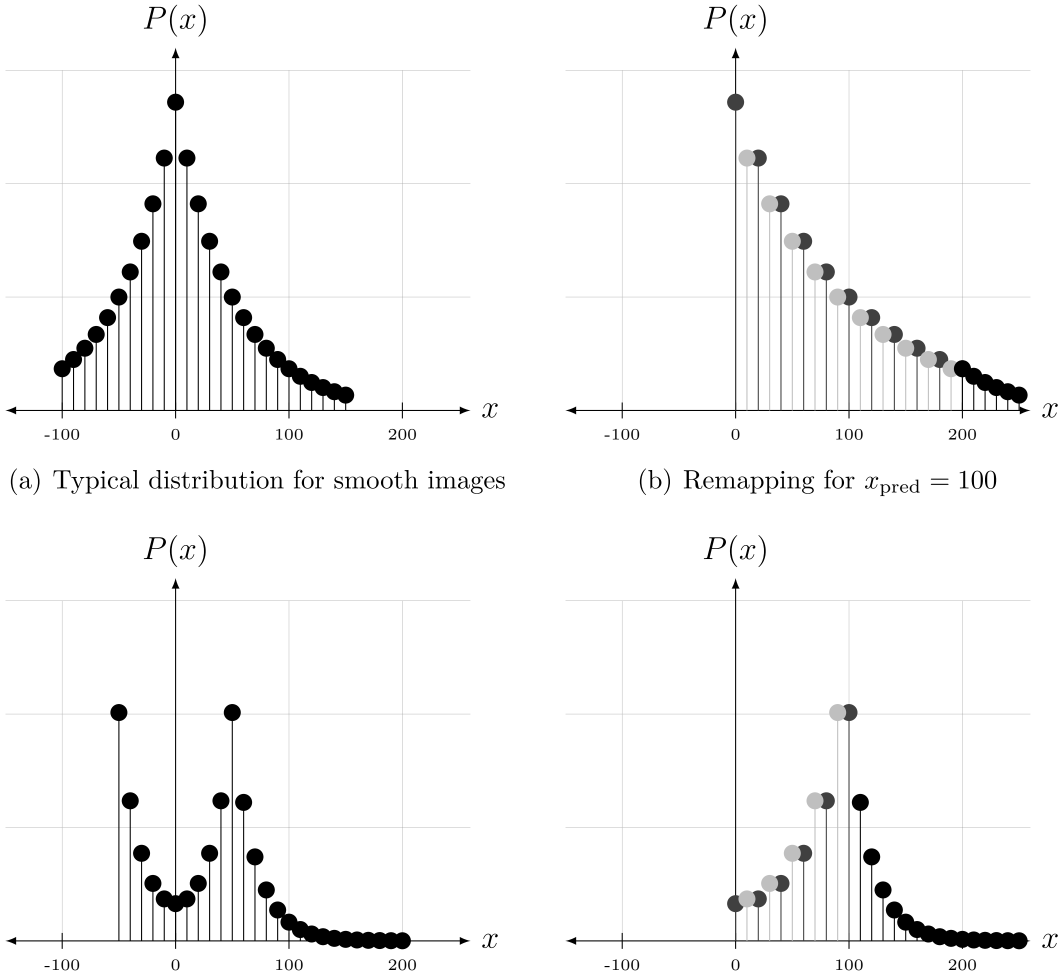
$$\mathcal{M}^{-+}(x, x_{\mathrm{pred}}) = \begin{cases} 2\delta & \text{if } 0 \leq \delta \leq \theta \ , \\ 2|\delta| - 1 & \text{if } -\theta \leq \delta \leq 0 \ , \\ \theta + |\delta| & \text{otherwise.} \end{cases}$$

where $\theta = \min(x_{\mathrm{pred}}, 255 - x_{\mathrm{pred}})$

Residue remapping reduces the alphabet size for entropy coding, by merging the tails of distributions with their central part. For a high-peaked distribution centered at the predicted value, the remapping function does not alter the two-sided geometric behavior of the residue distribution. For the bimodal distribution, the remapping function will merge the probability tails, enhancing the probability of an edge value. Both cases are illustrated in Figure D.9. A revision of different remapping functions can also be found in RICE [67]. In the case of the MMP algorithm, the residue remapping was applied to each coordinate of the multidimensional residue block after block prediction. The new remapped block was then used for pattern matching.

Error feedback

The error feedback technique is used in JPEG-LS and CALIC encoder, and consist of adaptively adjust the predicted value in accordance to the residue neighborhood.



(a) Typical distribution for smooth images

(b) Remapping for $x_{\text{pred}} = 100$

(c) Typical distribution for compound images

(d) Remapping for $x_{\text{pred}} = 50$

Figure D.9: Example of residue remapping for two kinds of residue distribution: for smooth images or for compound images. For smooth images, the residue distribution has a Laplacian characteristic, as shown in Figure (a). The remapping will not alter the behavior of the distribution, as shown in Figure (b). In the case of compound images, the residue distribution can be modeled by a bimodal function, as seen in Figure (c). The residue remapping will merge the tail of the distribution, resulting in a high probability for edge values, as seen in Figure (d).

The value that is used for updating is usually an average value of the neighboring residues, and a context is also used to increase adaptation speed of the mean value.

For the MMP algorithm, the same principle was applied. The average values of the residues were kept in a structure, that was later used for updating the block prediction value. The block dimension was used as context, as well as the chosen prediction mode. For example, a horizontal prediction block value was updated with average values of residues, obtained from the same block dimension after horizontal prediction.

Experimental results

Table D.4: Lossless compression results for several different residue encoding techniques, given in bits-per-pixel. Best results are highlighted in bold.

Images	**Histogram Restriction**	**Residue Remapping**	**Error Feedback**	**MMP-FP DPCM**
Barbara	4.427	4.423	4.587	**4.352**
Cameraman	4.631	4.640	4.707	**4.570**
Gold	4.711	4.833	4.914	**4.505**
Lena	4.347	4.227	4.317	**4.212**
PP1205	**2.696**	2.706	2.717	2.736
PP1209	2.623	2.724	2.678	**2.567**
Shapes	1.040	0.970	**0.907**	0.916
Average	3.496	3.503	3.547	**3.408**

The tested techniques did not provide any consistent gains over the MMP-FP algorithm using the DPCM enhancement on the intra prediction modes, as can be seen in Table D.4. Since these techniques are mutually exclusive, the DPCM enhancement technique was adopted, instead of the residue enhancement techniques. With a multiscale dictionary, the MMP algorithm already exploits a common neighborhood for residual encoding. Also the adaptive update of the dictionary already introduces only patterns that rely inside the restriction of predicted values, which are patterns close to zero.

By restricting the adaptation of MMP's dictionary, we may also be hindering the dictionary adaptation process, which may impact the global compression capability of MMP. Many other techniques for restricting the behavior of the dictionary adaptation, such as histogram truncation, were also tested, with a similar outcome. Therefore, no entropy adaptation technique was adopted by the MMP lossless algorithm, and the arithmetic encoder in the lossless version is used exactly the same as in the lossy version. This is also advantageous, since the encoding procedure for lossless and lossy coding is also the same.

D.5.3 Comparison with state-of-the-art lossless algorithms

In the lossless case, we chose three different algorithms to compare with MMP: JPEG-LS [110], MRP [111] and PNG [108]. JPEG-LS is a benchmark for lossless compression and uses implicit prediction and adaptive Golomb codes for residue encoding to efficiently compress the image in a single pass. The used codec is available at [132]. It allows an overall good compression for smooth images with relatively small computational cost. MRP uses explicit prediction and adaptive residue encoding on a multiple-pass optimization routine. Although computationally intensive,

Table D.5: Lossless compression for smooth images. Results in bits-per-pixel. Best results are highlighted in bold.

Image	**JPEG-LS**	**MRP**	**PNG**	**MMP**
airplane	3.817	**3.591**	4.220	3.943
baboon	6.037	**5.663**	6.210	6.028
balloon	2.904	**2.579**	3.238	2.830
barb	4.691	**3.815**	5.199	4.352
barb2	4.686	**4.216**	5.132	4.659
camera	4.314	**3.949**	4.674	4.570
couple	3.699	**3.388**	4.061	3.915
goldhill	4.477	**4.207**	4.662	4.505
lena	4.607	**4.280**	4.888	4.489
lennagrey	4.238	**3.889**	4.577	4.212
noisesquare	5.683	**5.270**	5.671	5.365
peppers	4.513	**4.199**	4.887	4.500
Average	4.472	**4.087**	4.785	4.447

Table D.6: Lossless compression for compound and artificial images. Results in bits-per-pixel. Best results are highlighted in bold.

Image	**JPEG-LS**	**MRP**	**PNG**	**MMP**
pp1205	4.472	4.039	**2.550**	2.736
pp1209	4.616	3.903	2.785	**2.567**
scan_002	3.429	**3.029**	3.729	3.471
scan_004	2.370	**2.041**	2.404	2.224
scan_006	3.202	**2.844**	3.037	2.943
shapes	1.214	**0.685**	1.154	0.916
Average	3.217	2.757	2.610	**2.476**

this algorithm produces one of the best compression rates reported for a large set of images, outperforming most of other lossless algorithms. The presented results were obtained with the optimization flag for the predictors set, using the codec available at [136]. PNG uses a combination of prediction and pattern matching, which produces relatively good results, specially for compound images, when compared to other algorithms. It was selected because, as MMP, it is dictionary-based, resembling the LZ77 algorithm [31]. However, one should bear in mind that, unlike MMP, that employs a block-based approach, PNG employs a pixel-based approach. We used the PNG software implementation available at [142], and the results were further optimized with the *pngcrush* tool [143].

Tables D.5 and D.6 show the compression results, in bits per pixel, for all four encoders.

In the case of smooth images, the residue after prediction has a probability distribution that is well fitted by an exponentially decaying model. JPEG-LS and MRP use this assumption, being able to capture the residues structure using models, such as Generalized Gaussian models and obtaining good compression results for smooth images, as we can see in Table D.5. PNG, on the other hand, does not explore this smoothness property and cannot compress such images as efficiently. Since PNG codes its residue using a dictionary approach, despite being asymptotically optimum, it takes too long to adapt its dictionary to the image statistics. MMP also uses a dictionary approach for residue encoding. However, as can be seen from Table D.5, its compression performance is compatible to JPEG-LS and MRP, due to its faster dictionary adaption to the image statistics [21].

Algorithms that model the residue using an specific probability function, like the exponential distribution in JPEG-LS, usually have a good performance for smooth images. The exponential assumption fits relatively well the residue distribution after prediction for these type of images. But in the case of compound images, the smoothness assumption generally does not hold, and thus prediction is not effective. Therefore, for compound images, the residues distribution does not match an exponentially decaying function. Then, the compression efficiency of MRP and JPEG-LS is much lower than the one of pattern matching algorithms like PNG, which does not assume any distribution for the residual data, and adapts itself as the image is being coded. In Table D.6, we can see that dictionary-based encoders such as PNG and MMP outperform the other algorithms when encoding some of the compound images of the chosen testset. Those images have a sparse histogram also, which is well exploited by the dictionary based approaches. If a palletization tool is used in JPEG-LS algorithm, a much higher compression efficiency can be achieved (namely 2.53 and 2.29 bpp for PP1205 and PP1209 images, respectively). Nevertheless, all approaches had either to pre-analyze the entire image for coding optimization or perform multi-pass encoding, while MMP does only a block optimization, and is able to achieve competitive compression rates for any kind of image, without any previous knowledge of the source to be encoded.

The MMP prediction step, combined with flexible partitioning, is able to leverage its dictionary adaptation process and provide a competitive performance for smooth images, while maintaining a good performance for compound images. Our results suggest that, on the average, MMP has a very efficient performance for compound images and a competitive compression efficiency for smooth images, thus having a good overall performance, when a wide range of image types is considered.

D.6 Conclusions

In this appendix we evaluated the performance of MMP acting as a lossless encoder. A theoretical bound on its redundancy was provided, showing that its rate converges to the source's entropy for large enough blocks. The theoretical results obtained suggest that enhancing the prediction step may lead to an increase in its lossless coding performance. Based on these results, we propose modifications in its prediction modes. The experimental results in this appendix and in Appendix C show that the same algorithm, when applied to lossy or lossless encoding, achieves state-of-the-art PSNR performance for a wide range of image types, from smooth to textual images, outperforming in most of the cases state-of-the-art encoders such as JPEG-LS, H.264/AVC Intra and JPEG2000. Therefore, MMP can be viewed as an universal tool for image encoding, that can be used for both lossy and lossless compression. All the code used for the simulations and the results are available online and can be found in [144].

Apêndice E

Multiview image compression

SUMMARY: Due to the novelty of this topic, this appendix provides an overview of the technologies present in the 3D chain of operation. Section E.1 reviews the state-of-the-art technology that enables the production, coding and display of 3D video content. Due to its importance, the view synthesis process is detailed in Section E.2. The standardization activity that has been going on is reviewed in Section E.3.

E.1 Review of the state-of-the-art in multiview coding

Lately we have been witnessing a growing interest in 3D-media related production, delivery and displays. A number of major studios have announced releases in 3D. Broadcasters are starting to offer channels with stereo content using their established mono video infra-structure. There are several 3D-ready displays available in the market, which allow the visualization of 3D content using several different techniques. The push from production, delivery and display side, that is, all the elements in the chain of operation of a 3D-video system, is fueling this renewed interest in 3D [68].

The present 3D technology incorporates two important concepts: 3DTV and Free-viewpoint TV (FTV). These are similar technologies, but for historical reasons are dealt with separately, based on the main focus of each activity [145]. 3DTV is the perception of depth information from a scene, which is accomplished by providing clues obtained from binocular parallax[1] or motion parallax[2]. FTV allows the user to change his viewpoint freely, being able to navigate within real world visual scenes. Both technologies require the capture of multiple views and the rendering of virtual

[1] The ability to provide depth sensation by showing two slightly different images to each eye.

[2] The ability to provide depth sensation by showing occluded parts of the scene, when the spectator moves its viewpoint.

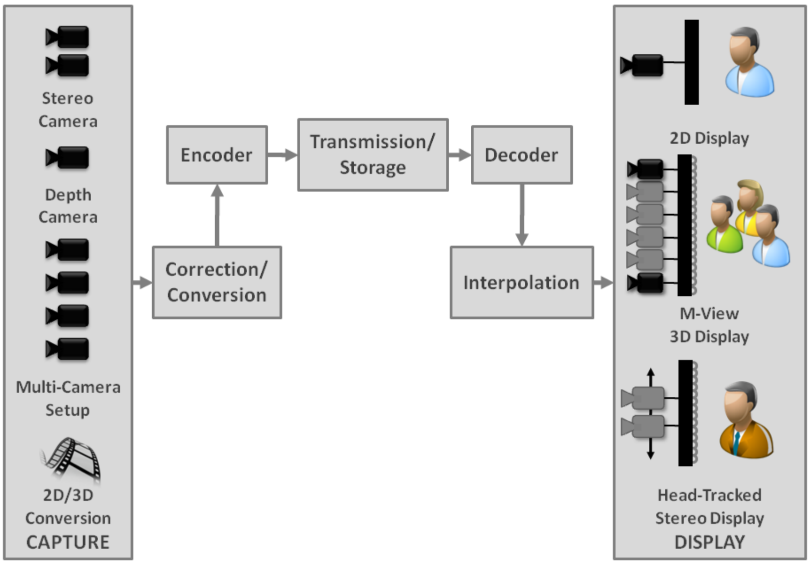


Figure E.1: Complete chain of operation for a 3D system.

scenes. They are expected to revolutionize the way we watch television nowadays, providing an innovative immersive experience [146].

Many advanced techniques are used to accomplish 3D-vision. Next we will review the state-of-the-art technology used in the various stages of a complete chain of operation for 3D video, as depicted in Figure E.1.

E.1.1 The 3D operation chain: capturing, coding and displaying

Representation of 3D media

There are several different ways of representing a 3D data format. The choice of a certain format will determine both the complexity and the capability of the system, and is of central importance for its design. In [146], data representation is classified according to the number of views (image-based representation) and the number of depths (geometry-based representation), as depicted in Figure E.2.

3D computer graphics use a geometry-based representation, where the scene is described by 3D meshes, with associated texture mapped onto them. View synthesis usually requires sophisticated computational resources, and the rendering quality may suffer from error prone image processing algorithms, such as object segmentation and 3D geometry reconstruction. Over the last few years, technological evolution has been able to bridge this gap, but still high-quality productions demands costly human intervention [145].

Image-based approaches, like the 4D ray-space representation [146], 5D McMil-

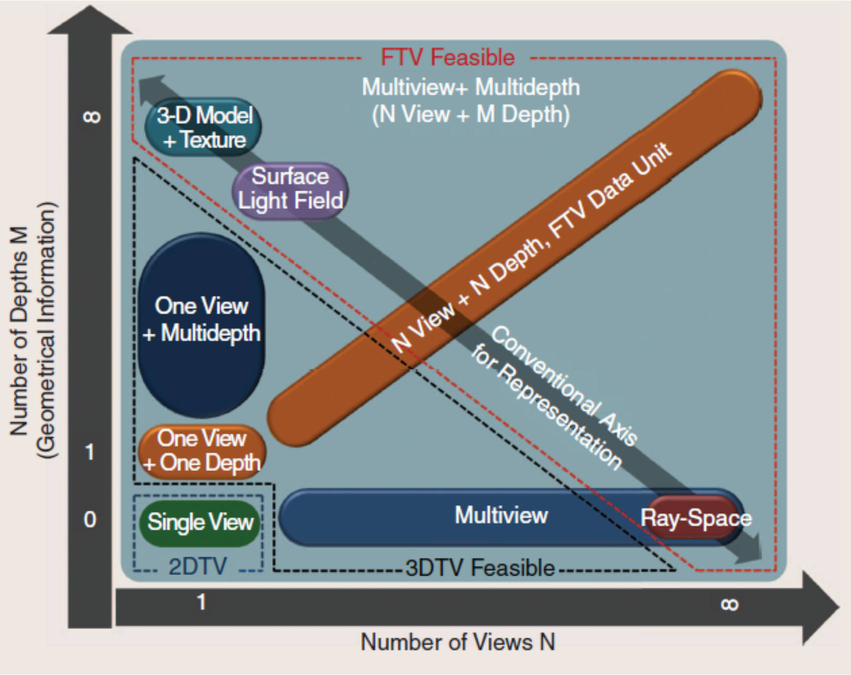


Figure E.2: View-geometry representation, taken from [146].

liam and Bishop's plenoptic modeling [147] or lightfields [148] have low computational requirements and can provide a high quality view synthesis. However, a dense camera setting is mandatory, in order to avoid poor rendering results, which may occur due to the presence of occlusion, depth quantization and geometry distortion. This implies also increased complexity in terms of data acquisition and data compression.

Legacy 2D systems use only one view and no geometry information is required. Image-based stereoscopic perception can be achieved by using two or more views. The process of transmitting them separately is called simulcast and requires at least double the data-rate compared to conventional stereo video. According to the binocular suppression theory [149], stereo vision quality is dominated, to some extent, by the quality of the higher fidelity view. Therefore, an uneven bitrate allocation may be used to compress the stereo pair more efficiently.

In order to use the well established 2D video codecs and delivery infrastructure, two views may be packed into a single frame, using a process called stereo interleaving [150]. This format describes how the stereo signal may be multiplexed into a single frame, for example by horizontal or vertical decimation and/or storing in a side-by-side or top-bottom format. Additional signaling is used to inform the chosen Frame Packing method. For this purpose, new SEI messages have been specified as an amendment of the AVC standard [151]. This method can be quickly deployed, but on the other hand spatial and temporal resolution may be lost.

In the case of multiview, such as the 9-view configuration used in autostereoscopic displays, a 9-fold increase in data rate is expected if simulcast is used. More efficient approaches that exploit the interview dependency have been proposed and the Multiview Video Coding (MVC) standard, an extension of H.264/AVC, emerged [152]. However MVC still presents a linear relationship between the data required and the number of transmitted views [152], and does not provide a continuum of output views.

A hybrid approach is the use of 2D+Depth format, that was standardized by ISO/IEC 23002-3, also referred to as MPEG-C Part 3 [70, 71]. 2D+Depth signals enable a display independent solution, which can generate an increased number of views, if needed, being also backward compatible with legacy 2D systems. This format is also more efficient than the multiview format, since depth data can be compressed at 10-20% of the bitrate spent to encode the color video. However, this format is only capable of rendering a limited depth range, adds complexity to the decoder (that now needs to perform view synthesis) and cannot handle occlusions appropriately.

Advanced 3D video formats are being considered by the MPEG group. MPEG envisions a new format that is capable of handling high-quality auto-stereoscopic displays, is suited for free-viewpoint TV, enables a variable baseline in stereoscopic displays and decouples the creation from the display of 3D content [69].

Multiple views with associated depth maps (MVD) can be regarded as an extension of the Video+Depth format. This approach is backward compatible with 2D and 2D+Depth standards and is flexible enough to implement Free-viewpoint TV. Structures for efficient representation of the N View + N Depth approach have also been proposed, like the limited view selection approach by [153] or the FTV Data Unit from [154].

An alternative approach to MVD is the use of layers to separate objects in a scene, and separately sending texture and depth from each layer [155, 156]. However, the extraction of layer utilizes unreliable depth data, which can result in artifacts at the reconstructed view.

Following the trend in industry for using stereo content in the production phase, another format compatible to the present stereo configuration is the depth enhanced stereo (DES) format. This format proposes the use of the usual stereo pair video, along with its corresponding depth map information, for an enhanced view synthesis. It is backward compatible with conventional stereo, and still fulfills MPEG requirements for data rate constraint and high quality view reconstruction [157].

Capturing

A big challenge in multiview coding is the appropriate capture of a 3D scene. For image-based data representation, the number of views is directly proportional to

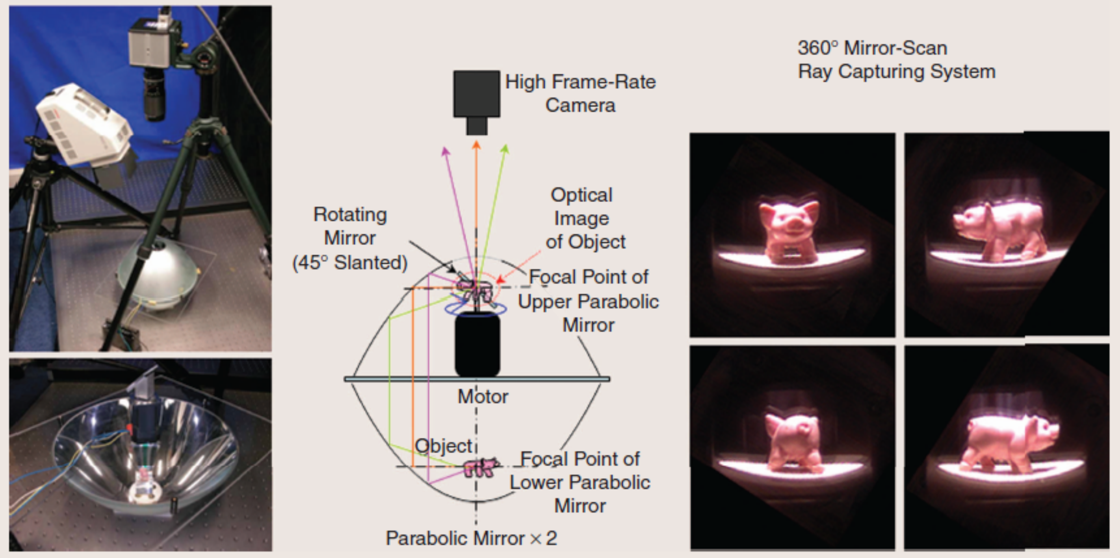

High Frame-Rate Camera
360° Mirror-Scan Ray Capturing System
Rotating Mirror (45° Slanted)
Optical Image of Object
Focal Point of Upper Parabolic Mirror
Motor
Object
Focal Point of Lower Parabolic Mirror
Parabolic Mirror × 2

Figure E.3: Acquisition system for Free-Viewpoint of static scenes, taken from [146].

the quality of the decoded image. However, a tradeoff between costs (number of cameras, processors, etc.) and quality (navigation range, quality of virtual views) needs also to be considered.

In the case of static scenes, usually just one single camera is used, where multiple images of the object are captured with the help of robotic arms, turn tables or a camera gantry. Figure E.3 shows an example of acquisition system that is able to capture all-around convergent views of an object. For dynamic scenes, an array of cameras is used. The camera arrangement can vary. Typically, horizontally aligned cameras are used, but also 1D half-arch, 2D or even dome-based camera arrangements can be found in the literature, such as the ones seen in Figure E.4.

In a multi-camera arrangement many issues have to be considered. Cameras need to be synchronized, the amount of data involved in such systems is huge, and a post-processing of the acquired images may still be needed. Solutions consider the external synchronization of the cameras, the use of cluster PCs for real-time processing of the captured multiple video, lens distortion adjustments and color correction algorithms to cope for differences between the cameras and also improvement of interview decorrelation [161]. Another post-processing step includes the centralization of the principal point and the equalization of the focal lengths, which aims to eliminate misalignments between the cameras. This process, depicted in Figure E.5, is called normalization.

In both scenarios it may be necessary to synthesize virtual views at space positions where there is no captured view. For such operation the scene geometry contained in the depth data, is used to improve the quality of the rendered view. In order to use the depth data and the texture of different views to synthesize an image, the pinhole camera model is used [162]. Knowledge of camera's intrinsic

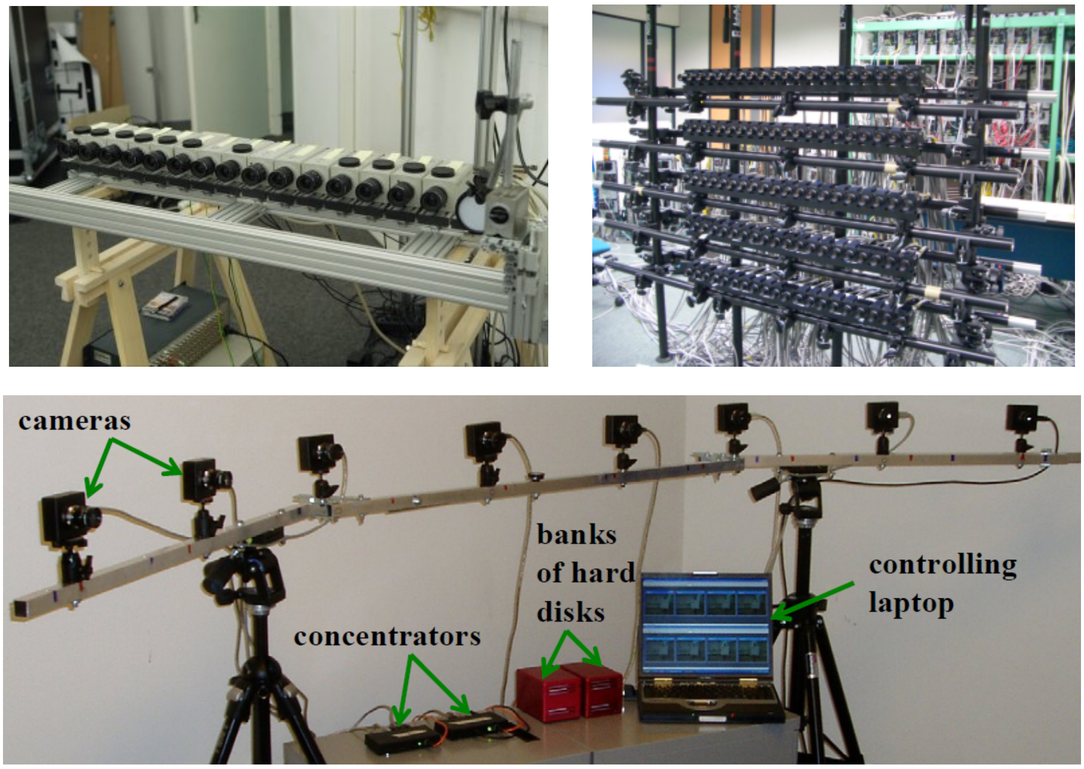


Figure E.4: Horizontal camera arrangement at HHI [158], 2D camera arrangement at Nagoya [159] and eight cameras placed along a 1D arc spanning about 30° from one end to the other, at Microsoft [160].

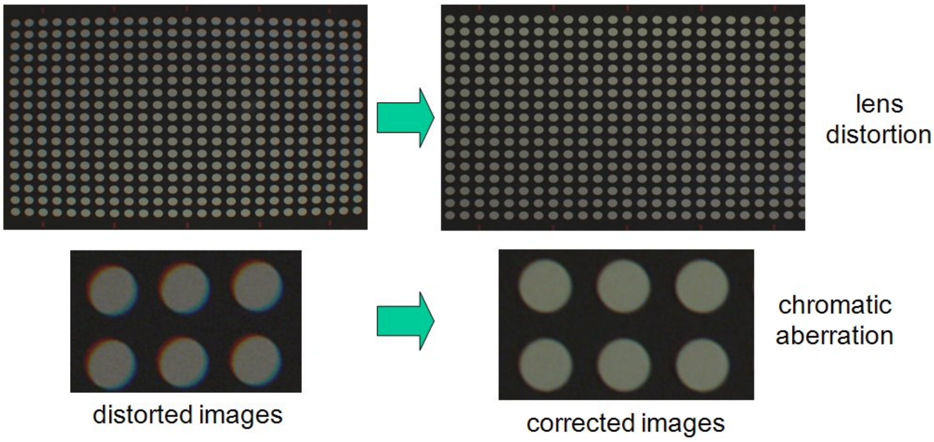


Figure E.5: The normalization process contains a correction of lens distortions, a correction of chromatic aberration, a centralization of the principal point and an equalization of the focal length. Figure taken from [158].

(focal distance, pixel aspect ratio, principal point, etc.) and extrinsic parameters (position and rotation) is assumed. The process of estimating those parameters is called camera calibration [163].

Depth information can be obtained in three different ways. For geometry-based models, it can be extracted directly from the camera position and the geometry model information. Another approach is the use of active depth range cameras, which use techniques such as time-of-flight, structured light or light coding, to measure the distance between objects and the camera. Depth camera usage is limited to nearby objects, and it usually captures depth of a scene at a slightly different position and with lower resolution. It is expected that new cameras will come to the market being able to provide color and pixelwise depth information simultaneously [82].

A very common method of depth estimation is the use of stereo matching algorithms to extract depth information from a pair of images. This class of algorithms does a search for corresponding points of the same object across different views. Once the corresponding points are found, depth information can be extracted by triangulation. Several algorithms have been proposed for both static [73] and dynamic images [160]. Although there has been a remarkable evolution in stereo matching algorithms, mismatch at smooth areas and occluded or partially occluded areas may still cause errors in the depth estimation.

One way to reduce the dimensionality of the search for stereo matches is to align cameras horizontally, so that the disparity between images occurs only in the horizontal direction (vertical disparity is null). This is achieved by using an image rectification process, which corresponds to a virtual rotation of the cameras, until they are all aligned [162].

Coding

From all the presented formats for 3D video representation, one common characteristic stands out: the increased amount of data. The straightforward independent encoding of each view requires low complexity and enables backward compatibility with 2D systems, but storage and channel capacity limits the use of this solution. Efficient compression of this great amount of data is then crucial for the deployment of 3D systems [72].

Video encoders exploit the temporal redundancy in a video sequence to achieve compression. Therefore, it is expected that higher compression may be achieved for multiview sequences if the interview redundancy is also used for view decorrelation. Since cameras capture the same scene from different viewing angles, there is a high probability that significant interview statistical dependency exists.

To exploit the interview dependencies, disparity compensation techniques are used. Similar to motion compensation, a block-based search can be done between views at the same temporal instant, therefore providing interview decorrelation. So by inserting the decoded neighboring view in the reference picture list, the encoder is able to choose between temporal prediction and interview prediction, through the use of flexible reference picture management. The best prediction mode is selected in a block-by-block-basis [164].

The standard for Multiview Video Coding (MVC) has been specified as an amendment of the H.264/AVC standard [9]. It is able to exploit the redundancy among views with just a minor change in syntax. A backward compatible view is included in a base layer stream and encapsulated in Network Abstraction Layers (NAL) units defined for single view video. The other enhancement views are encapsulated in a new NAL unit type for multiview video [68]. It is important to emphasize is that the decoder does not need to be aware of whether the reference picture is a time or multiview reference, which implies no change of low-level syntax, and allows it to be compatible with existent H.264/AVC codecs.

A drawback of MVC encoder is the complex structure that arises from temporal and interview prediction. This limits random access, increases system delay, memory requirements and adds complexity to the encoding procedure. The limitation of interview dependency and the omission of interview-prediction for pictures that already have temporal references can significantly decrease the complexity with only a small rate penalty, since major gains occur mostly at anchor positions [72].

Coding gains of up to 3.2dB were achieved, but they can be strongly related to the type of sequence that is being encoded. Problems with large disparities, occluded areas and color differences between frames can compromise the coding efficiency of a disparity-based prediction approach. Also a linear relationship exists between the resulting data rate and the number of views. So there is a practical limitation in the use of MVC for coding multiple views.

Color and illumination inconsistencies can be corrected by pre-processing the input sequence. Depth-based view synthesis prediction can solve the problem of large disparities. YEA e VETRO [86] used depth data to warp pixels from neighboring views and create a synthetic view that is added to the frame buffer structure of the H.264/MVC encoder. The encoder may choose this new View Synthesis Prediction (VSP) mode in a rate-distortion sense. KITAHARA *et al.* [87] presented a different approach. The encoder compresses an anchor view and enhancement layers, obtained from the residual frames between the original neighboring views and the synthesized view, generated by an anchor view and its corresponding depth data. Both methods proposed by YEA e VETRO [86] and KITAHARA *et al.* [87] were not adopted by the MVC standard due to a change requirement in macroblock decoding level, but might be revisited for a new 3D coding format. A structure that uses a depth-based prediction was proposed for the MPEG group, the FTV Data Unit [154].

Another focus of 3D video coding research is the compression of depth maps. Depth map images are usually piecewise-smooth images, where sharp edge information indicates the boundaries of objects at different viewing distances [73]. These grey scale images can be fed into the luminance channel of a video signal and coded as luminance-only signals with state-of-the-art video encoders, such as H.264/AVC or H.264/MVC. However, video encoders are highly optimized for visual content coding, whereas depth maps are used for rendering only. Coding artifacts (e.g. depth edge smoothing) may affect the 3D rendering, causing errors such as color displacement artifacts around foreground objects [82].

A common way to avoid edge artifacts and still profit from transform-based encoder is to use a Region of Interest (ROI), and code the depth map by parts. In [77], JPEG2000 is used, together with a ROI that is determined based on the edge information and encoded separately, using different bit planes. In [165], edges are encoded near-losslessly in a separate layer, while the rest of the depth map can be encoded with any other lossy encoder. Kim et al [166] propose to use a new distortion measurement, by deriving a relationship between artifacts in the depth map and distortion in the rendered view. Bitrate savings can be achieved by heavily quantizing areas that are immune to depth variations, but are directly related to the target view position. Similar results are also obtained with a multipass approach, like the one presented in [167].

Several filtering techniques have also been proposed, that aim to preserve edge information. In [168], a low-resolution map is used. The up-sampling filter uses information from the high resolution color texture to correct and sharpen the edge information. OH *et al.* [169] propose an inloop reconstruction filter that eliminates coding artifacts in the depth maps and improves the reconstructed images. The same technique is applied to an up sampling function in [170], for lower resolution depth maps.

Another approach is to code the depth using specific-purpose algorithms, based on geometric representation of the data. In [78] a platelet-based coding algorithm is proposed. The algorithm employs a quad-tree decomposition of the image, and approximates each block segment by a piecewise-linear function. Due to its ability to preserve sharp object boundaries and edges, it presents a high rendering quality. A drawback of the platelet-based approach is that it appears difficult to extend this scheme to video.

Other proposals have been presented for depth map compression, which are based on different encoding tools, such as the combination of non-uniform sampling and adaptive meshing of images or compressive sensing, as in the work presented by

Sarkis et al [80, 81]. In [171], multiple views are warped to a central view and then encoded using a 3D-DCT scheme, in order to exploit the interview correlation using the depth data and view synthesis.

Displaying

Lately we have witnessed a remarkable evolution on display technology, and 3D-ready high definition television is now the newest hit on consumer markets everywhere. The acronym 3D-ready relates to televisions that can identify uncompressed 3D content that was sent using frame packing methods, and appropriately transmit the respective content for each eye, usually with the help of glasses [172].

Glasses used in 3D systems have the purpose of separating the images for each eye and providing stereoscopic vision. This separation can be done by wavelength-division multiplexing (e.g. anaglyphs), by light polarization (e.g., linear polarization or circular polarization) or by light shuttering (e.g. the widely used shutter glasses) [173].

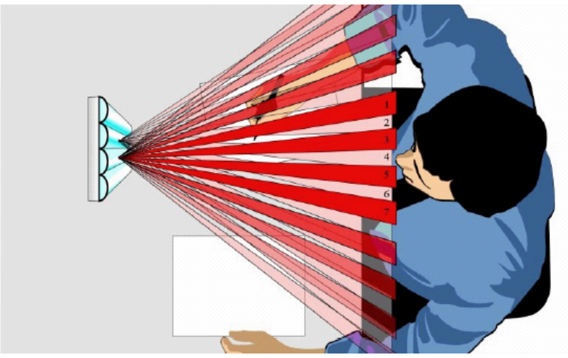


Figure E.6: Auto-stereoscopic display requiring 9 views (N = 9), taken from [173].

Another promising technology for 3D displays is the autostereoscopic display (see Figure E.6), which does not require the use of glasses. The multiplexing of left and right images is done spatially, followed by a light-directing mechanism. The views can be separated with the use of parallax barriers (e.g, a very thin sheet of aluminum or inked screen) or lenticular lens (e.g., thin cylindrical microlenses), see Figure E.7. Nevertheless, this technology usually presents a blurred image due to pixel crosstalk, requiring high-resolution displays. Moreover, many views are necessary in order to provide depth cues for motion parallax.

Multiview 3D displays require a greater number of views, but are able to provide motion parallax. They can be active displays, where the head position is tracked, or passive, where a large number of views are displayed simultaneously by spatial multiplexing. An emerging class of multiview displays use scalable multi-projectors. Those systems offer very high resolution, flexibility, excellent cost-performance, scal-

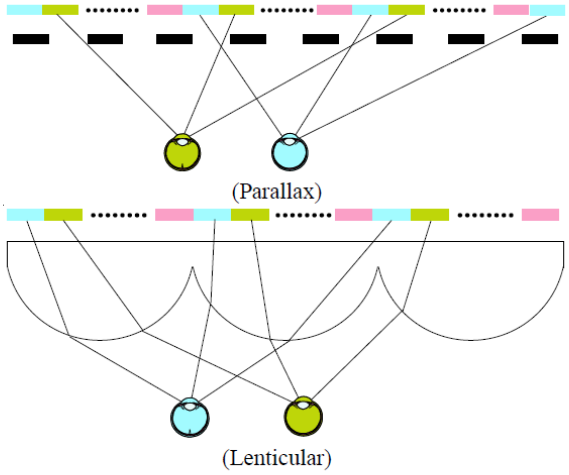


Figure E.7: Comparison of lenticular and parallax barriers for 3D displays. Figure taken from [174].

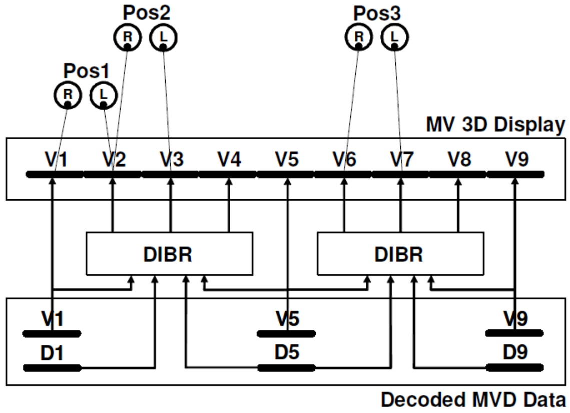


Figure E.8: Advanced 3D video concept based on a multiview plus depth format; Pos: viewpoint, R: right eye, L: left eye, V: view/image, D: depth. Figure taken from [145].

ability, and large-format images. Examples of the use of such displays are the multiprojector displays using lenticular sheet at MERL [175] and joint MIT/Cambrigde work [176].

E.2 Depth image based rendering

The need for sending a larger number of views for a multiview display may be suppressed by the use of depth-image base rendering algorithms (DIBR). Figure E.8 shows how DIBR can be used in multiview displays.

DIBR is the process of synthesizing an arbitrary "virtual" view from the reference images and associated pixel-wise depth information. It uses the framework of projective geometry, where a 3-element vector $(x, y, z)^T$ can be described using a

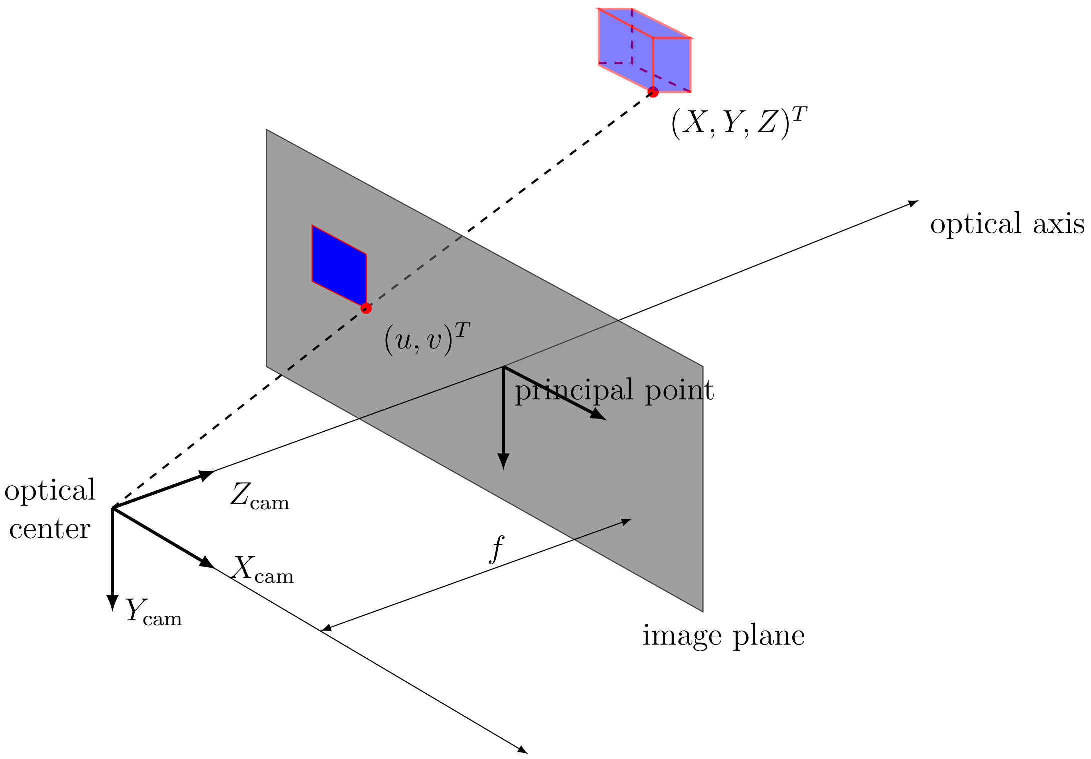


Figure E.9: Pinhole Camera Model. Notice that all coordinates are taken with origin at the optical center, that coincides with the camera position defined by the coordinate $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})^T$.

4-element vector $(x', y', z', w')^T$ such that

$$x = \frac{x'}{w'} \tag{E.1}$$

$$y = \frac{y'}{w'} \tag{E.2}$$

$$z = \frac{z'}{w'} \tag{E.3}$$

where $w' \neq 0$. The coordinates $(x, y, z)^T$ and $(x', y', z', w')^T$ are called inhomogeneous and homogeneous coordinates, respectively.

By perspective projection, a 3D point can be projected onto an image plane using the so-called pinhole camera model. The center of the perspective projection is called optical center or camera center. Optical axis is the line perpendicular to the image plane that passes through the optical center and intersects the image plane at the principal point (see Figure E.9).

The projection of a 3D world point $(X, Y, Z)^T$ onto the image plane at pixel position $(u, v)^T$ can be determined by the projection mapping (using homogenous coordinates), given by

$$\lambda \mathbf{p} = \left[\mathbf{K}|\mathbf{0_3}\right] \mathbf{P} \tag{E.4}$$

where $\mathbf{P} = (X, Y, Z, 1)^T$ corresponds to the 3D point and $\mathbf{p} = (u, v, 1)^T$ corresponds to its 2D projection, both defined with homogeneous coordinates. $\mathbf{0_3}$ denotes the all zero element vector. The projection matrix $\mathbf{K}$ is also known as the camera's *intrinsic* matrix, and is given by

$$\mathbf{K} = \begin{bmatrix} f & \eta & O_x & 0 \\ 0 & \tau f & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{E.5}$$

where f denotes the focal length, $(O_x, O_y)^T$ corresponds to the principal point coordinate, η models the pixel aspect ratio and τ the skew of the pixels.

Equation (E.4) assumes the camera position at origin of the world coordinate system, which does not necessarily need to be true. Once the camera's position and orientation are known, translation and rotation operations can be incorporated into Equation (E.4). The camera's rotation is modeled by a 3×3 matrix $\mathbf{R}$, and its position is modeled by a 3×1 vector $\mathbf{C}$. Their concatenation is a matrix known as *extrinsic* matrix. Hence, Equation (E.4) results in:

$$\begin{aligned} \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{K} | \mathbf{0_3} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0_3} \\ \mathbf{0_3}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I_3} & -\mathbf{C} \\ \mathbf{0_3}^T & 1 \end{bmatrix} \mathbf{P} \\ &= \begin{bmatrix} \mathbf{K} | \mathbf{0_3} \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0_3}^T & 1 \end{bmatrix} \mathbf{P} \\ &= \mathbf{KR} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \mathbf{KRC} \end{aligned} \tag{E.6}$$

With the help of depth data, a captured image from a reference camera can be projected into the 3D world, and then reprojected into a target camera viewing point, a process also called 3D image warping [162], which is depicted in Figure E.10. At first, the 2D point from the reference view is back-projected into the 3D space, and will fall at a ray $P(\lambda)$ connecting the 2D point and the camera center $\mathbf{C} = (C_x, C_y, C_z)^T$. The 3D coordinate is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\mathbf{C} + \lambda \mathbf{R}^{-1} \mathbf{K}^{-1} \mathbf{p}}_{\text{ray } P(\lambda)} \tag{E.7}$$

In the case the depth Z is known, the coordinates X and Y are obtained by calculating λ using the relation $\lambda = \frac{Z - C_z}{z_3}$, where $(z_1, z_2, z_3)^T = \mathbf{R}^{-1}\mathbf{K}^{-1}(u, v, 1)^T$. Next, the

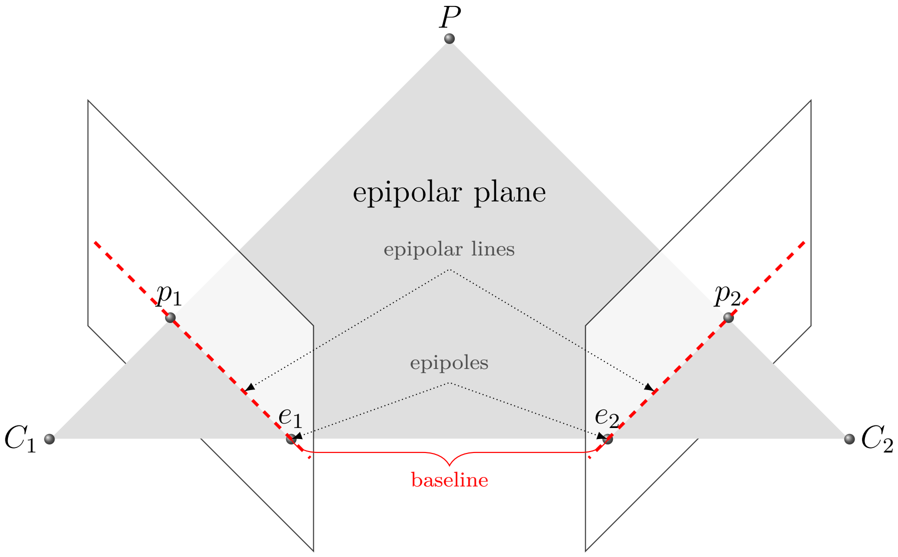


Figure E.10: The 3D warping operation is governed by the epipolar geometry. The epipolar plane is defined by the 3D point P and the camera centers C_1 and C_2. The baseline is the line going through the two camera centers, and the epipole is the image-point determined by the intersection of the image plane with the baseline. The epipolar line is the intersection of the image plane with the epipolar plane, and the projection of point P for both planes will be along their respective epipolar lines.

projection onto the target camera is obtained by using Equation E.6 and the intrinsic and extrinsic matrices of the corresponding camera. When pixels from the reference view are reprojected, an occlusion-compatible warp order has to be respected, that is if multiple pixels warp to the same location in the virtual view, then the pixel closest to the virtual camera has to assume the warped position. Notice that errors in the warping operation, such as holes, cracks or ghost contours may occur due to round-off errors, sampling or ill-defined depth boundaries. Nevertheless, these projection artifacts may be removed with a post-processing technique.

A warping operation is usually conducted to reconstruct virtual views that are located between camera positions. So it is possible to have two references and blend the projected image from both cameras onto the virtual camera view. The remaining disoccluded parts of the virtual image (that is, the newly exposed image that could not be visualized by the other two reference cameras) are filled with an inpainting process.

Different proposals to perform virtual view rendering have been made. ZINGER *et al.* [177] performs forward warping for both texture and depth. A median filter is applied to the warped depth image, and the modified positions are inverse-warped with the altered depth value, in order to remove cracks and holes. Ghost contours are removed by omitting warping of edges at high discontinuities and the occluded regions are filled by an inpainting algorithm that explicitly uses depth information.

In [153], an edge detector identifies objects' borders, and a 7-pixel wide area along

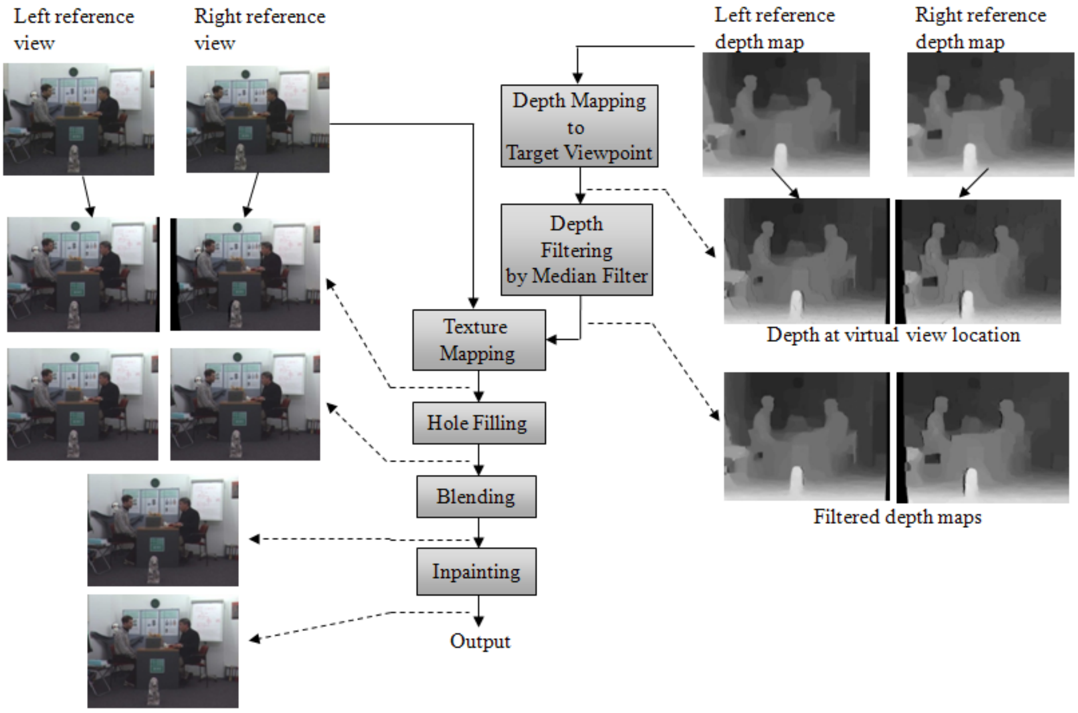


Figure E.11: Diagram for VSRS general mode, from [178].

the detected borders is considered as boundary area. The remaining area is called the main layer. The extended boundary area is further divided into background and foreground layer, obtaining a total of three layers. First the main layer is warped, where pixels that have similar depth from both reference views are linearly weighted. For pixels with very different depth values (determined using a pre-defined threshold), than the closest pixel (the one with lowest depth) is chosen. Next the foreground boundary layer is warped and merged with the main layer, where the front-most pixel from either layer is chosen directly. At last the remaining holes are filled with the background layer, or inpainted if no texture information is available. For an improved subjective quality, foreground objects' borders are low-pass filtered to provide a more natural appearance.

MPEG's reference software for view reconstruction (VSRS) enables two possible reconstruction modes: general mode (Figure E.11) and 1D mode (Figure E.12) [178]. The general mode is based on the work published by Mori et al [179]. First, the depth maps of the reference cameras are warped to the virtual viewpoint with the aid of a homography transform [180]. Then the empty pixels are filled with a median filter. The textures are retrieved by performing an inverse warping from the projected depth maps back to the reference cameras, a process called "reverse warping". After that, the texture images are blended, depending on the blending mode choice. The blending-on mode is a weighted blending based on the baseline distance. So pixels from the reference camera that is closer to the virtual view are assigned a higher

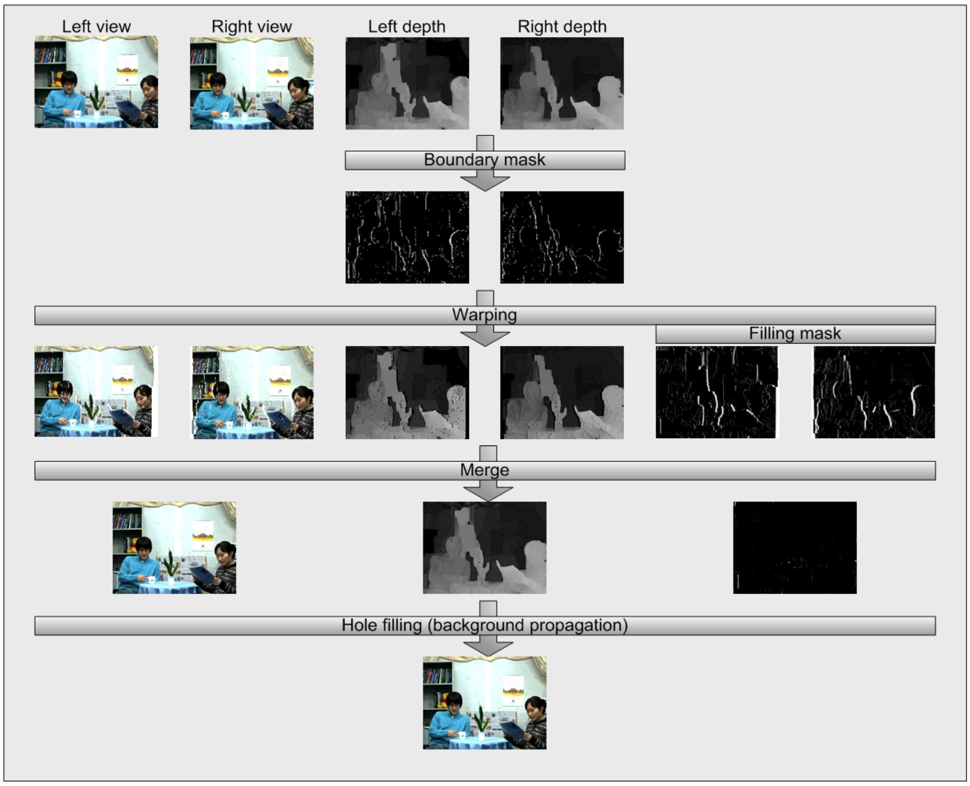


Figure E.12: Diagram for VSRS 1D mode, from [178].

weight, based on the baseline ratio. In blending-off mode, all pixels visible in the closer reference view are copied to the virtual view, and areas are filled from only the farther reference view if there is no other view available. At last, the remaining disocclusions are inpainted using a method similar to the one proposed by TELEA [181].

Additionally, VSRS also contains a Boundary Noise Removal algorithm [182]. In this mode, the holes caused by occlusion are used to identify object boundaries. After identifying the background side of the holes based on their depth values, the holes are expanded into the background, that is, background information is erased from the warped views and considered as holes. At the same time, noise around object boundaries is reduced, since the ghost effect is also reduced or removed in this process. These new holes are filled either with information from the opposite reference views, or with the inpainting algorithm.

The 1D mode is implemented assuming that the optical axes of the cameras are parallel to each other and the views are rectified, such that no vertical disparities exist. Only horizontal pixel shifting is necessary for the warping operation. At first, chrominance information is upsampled to 4:4:4 format and depth information is temporally filtered using a simplified algorithm proposed in [183]. The color video may also be upsampled, depending on the chosen sub-pixel precision (half-pel or quarter-pel). Warping of depth and texture is performed with the use of an enhancement technique [184], which limits the artifacts caused by texture-depth misalignment (where foreground pixels are scattered to the background) and miscategorized holes in the foreground (causing leakage of the background texture in foreground objects). The warped references are merged into one single picture, where three merging options are available: take the pixel closest to the camera, average the values proportional to the baseline distance, or perform adaptive merging, choosing one of the two mentioned merging methods, depending on a given threshold. At last, the remaining hole areas are filled with background texture information, by propagating the background boundary pixels horizontally inside the holes. The image is then downsampled to its original size, and the color format 4:2:0 is restored. Additionally, VSRS 1D mode can also use the Boundary Noise Removal algorithm [182].

As pointed out by DO *et al.* [185], the inpainting task is responsible for most of the rendering artifacts and can even decrease the final quality of the image. Notice that holes in the new view are caused when a foreground object disoccludes a background area, so missing texture is always present on the background of the scene. Several proposals were made for this stage (solving the hole filling problem), taking into account the depth information as well. In [186], the border that belongs to the background is reflected to the other side of the hole, and the boundary of the hole will contain only elements from the background. Therefore, any hole-filling method that uses color information propagated inward from the region boundaries will correctly fill the holes with patterns from the background only. ZINGER *et al.* [177] uses a weighted combination of the boundary pixels that belong to the background only, inversely proportional to the distance of the infilling position and the boundary neighbor position.

E.3 3D standardization

Standards are an important element of a technical concept, since they enable the interoperability between different systems of a content delivery chain, by defining the formats in which data will be handled. These interchangeable formats are typically specified by international standardization bodies such as ITU-T Video Coding Experts Group (VCEG) or the ISO/IECJTC 1 Moving Picture Experts Group (MPEG). The success of a standard relies on its ability to allow the introduction of new devices with enhanced capability, but also accounting for the existence of legacy devices [172].

The 3D area is a clear evidence of the need for standardization. Many different formats are in use, but still no predominant technology is in place. Just to name a few, new standards will be required for 3D video formats aimed at the

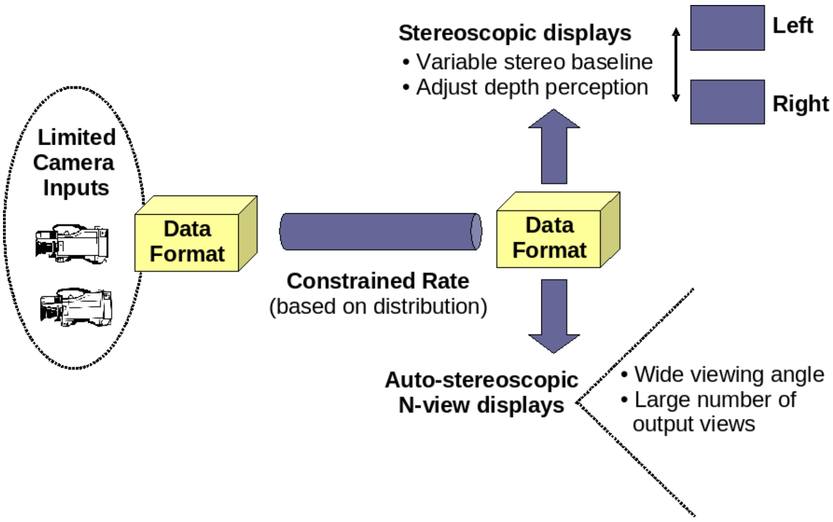


Figure E.13: Target of 3D Video format illustrating limited camera inputs and constrained rate transmission according to a distribution environment. The 3DV data format aims to be capable of rendering a large number of output views for auto-stereoscopic N-view display. Figure appeared in [69].

next-generation of 3D displays, transmission of 3D data, synchronization of eye-wear shutter glasses, etc. Ongoing standardization efforts are being made by Digital Cinema Technology Committee (DC28) of the Society of Motion Picture and Television Engineers (SMPTE), by the Society of Cable Telecommunications Engineers (SCTE), by the Consumer Electronics Associations (CEA) and the MPEG society in a group called 3DAV (for audio-visual) [68].

The standardization activities in the 3DAV group aim to develop a new 3D data format focused on the next generation of 3D displays. The new format will enable baseline adjustment for a more comfortable depth perception and also facilitate support for high-quality autostereoscopic displays [69]. This will allow the decoupling of content creation and display requirements, providing a higher number of high-quality views at the decoder side, without the need to increase the number of cameras in the production or to increase the data rate for transmission. Figure E.13 shows the scope of MPEG's standardization work on the new 3D format.

MPEG also envisions a data format that is backward compatible with the existent standards, and also with 2D television (see Figure E.14). A natural evolution of the available formats would be based on multiview video and depth, but other proposals should also be taken into consideration (such as the transmission of auxiliary data, segmentation information, transparency or specular reflection, occlusion data, etc.) [187].

To develop the new 3D standard, MPEG is up to issue a "Call for Proposals" (CfP) [11] on 3D Video Coding Technology. Proponents can send video content coded with their coding scheme, which will be evaluated using the Single Stimulus

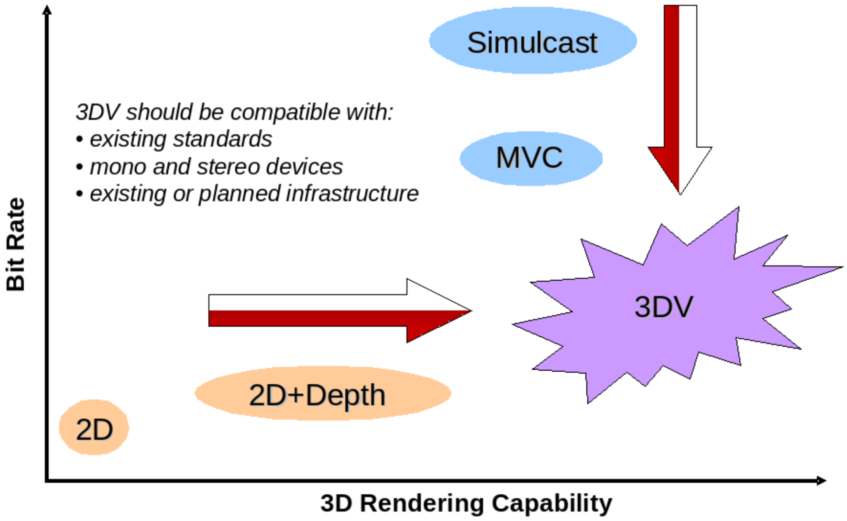

Bit Rate
3D Rendering Capability
Simulcast
MVC
3DV
2D+Depth
2D
3DV should be compatible with:
• existing standards
• mono and stereo devices
• existing or planned infrastructure

Figure E.14: Illustration of 3D rendering capability versus bit rate for different formats, where 3D Video aims to improve rendering capability of 2D+Depth format while reducing bit rate requirements relative to simulcast and MVC. Figure appeared in [69].

MultiMedia (SSMM) test method. This subjective evaluation will be done in a dark room using stereoscopic displays (with the help of polarized or shutter glasses) and also autostereoscopic displays.

For its experimental framework, MPEG selected an appropriate set of test data. Eleven multiview sequences have been chosen. They all share common characteristics: linear camera arrangement, rectified video, per-pixel depth data for each view and available camera parameters. All recorded video is progressive and the test data set covers a representative range of scene content complexity (e.g. in- and outdoor scenes), resolutions (720x540 - 1280x960 pixel), frame rates (16.7 - 30 fps), and number of cameras (3 - 80 cameras with 3.5 - 20 cm spacing between two neighboring cameras) [188].

The 3D video coding processing chain also comprises a depth estimation step and a view synthesis stage. In contrast to all previous coding approaches, interdependencies between the pure coding approach and the depth estimation and view synthesis exist, and need to be considered for the overall 3D video solution. Reference softwares for both activities are supplied by the MPEG group: DERS (depth estimation reference software) and VSRS (view synthesis reference software) [178].

MPEG includes the provision of anchors to be used for comparison. These are being developed in the context of 4 Exploration Experiments (EE): depth map generation (EE1), view synthesis (EE2), coding experiments (EE4) and 4-View test scenario (EE5) [188]. The results will provide a suitable reference technique based on already known encoders. The anchor coding is carried out for different quality levels and will serve as a reference for future novel coding proposals.

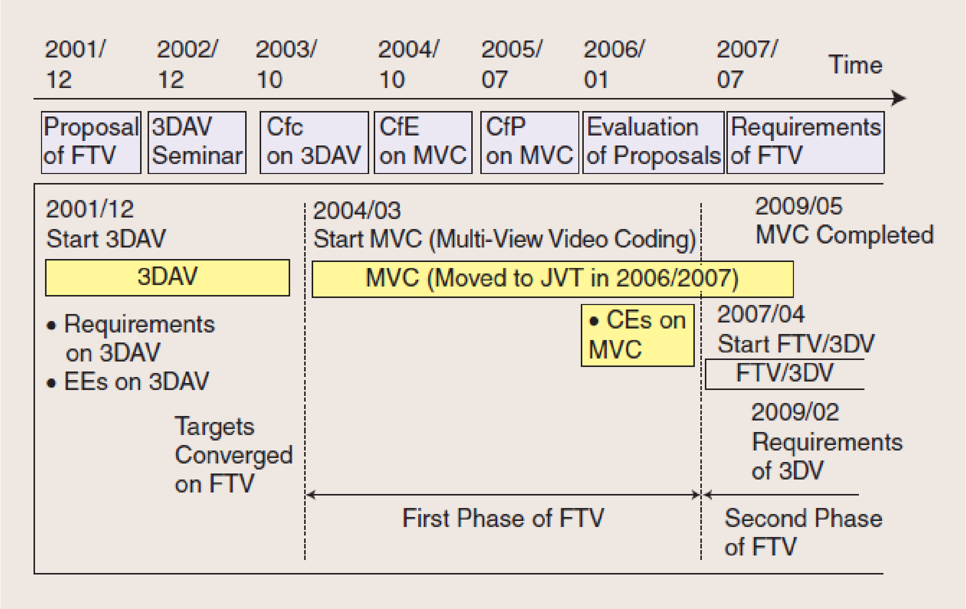


Figure E.15: FTV standardization in MPEG, taken from [146].

After the evaluation of the Call for Proposals the collaborative phase of the standard will start. This will include improvement, extension, or replacement of algorithms in comparison to the actual reference until best possible technology is defined. Further detailed textual specifications, performance evaluations, reference software and conformance bitstreams will be developed. Typically, the collaborative phase takes 2-3 years, so that the 3DV standard can be expected in 2012 or later (see Figure E.15).

Apêndice F

Depth coding using MMP

SUMMARY: The new format for 3D images comprises the transmission of multiple texture and depth signals. In this appendix, the encoding of depth maps with MMP will be investigated. Section F.1 presents the results of using MMP for coding depth maps. A discussion on metrics for depth map evaluation is also done and several results are presented. A proposal for reducing rendering artifacts by improving coding of depth map edges is given in F.2. Section F.3 presents some conclusions on this topic.

F.1 Coding depth information with MMP

In this work we propose the use of the MMP algorithm for depth map coding. Edge information plays a fundamental role in DIBR algorithms, as it is specially sensitive to coding artifacts. A minor error may cause disturbing visual effects on the reconstructed image, such as sample scattering and color bleeding. Therefore, it is expected from depth maps algorithms to preserve this edge information, that is, its high frequency content.

MMP shares some similarities with the recently proposed platelet-based depth map coding [78], also being able to preserve edges through a flexible segmentation and therefore suited for depth/disparity map encoding. It has though an advantage over Platelet encoders, that is, its coding procedure can be used without any modification for depth and texture, whereas in the case of platelet encoders this does not seem to be the case.

MMP also benefits from its dictionary update routine when coding depth maps. Once a pattern, say a sharp edge, is sliced into several uniform pieces, it can be efficiently encoded by codewords of the dictionary. The new encoded pattern is incorporated in the dictionary through an adaptation process. Since an edge usually spans over several blocks, the recently added edge pattern may be used to efficiently encode future segments, without further recurring to expensive block segmentation, while still being able to represent the sharp edge with high fidelity.

F.1.1 Evaluating coded depth maps

The coded depth maps are objectively evaluated. The depth map images used (or for some images the equivalent disparity map images used) were taken from the Middlebury test set[1] (Teddy and Tsukuba, [24, 73]), from Microsoft[2] (Ballet and Breakdancers, [160]) and from MPEG's set of reference anchor sequences (Book Arrival and Champagne Tower, [158, 159]).

Depth maps from the stereo pairs Tsukuba and Teddy are representative of depth maps without any errors, that is, they are the ground truth. For the other sequences, the estimated depth map provided by the sources were used, and they do not represent the ground truth. Depths maps from the Microsoft group are related only to the first frame, from two selected viewpoints that are aligned in a 1-D half arch arrangement. Likewise, depth maps from MPEG's sequences use only the first frame. However, the camera arrangement for those sequences are aligned horizontally. Notice that for all tests, we are working with an instantaneous texture, and not with the video sequence as a whole.

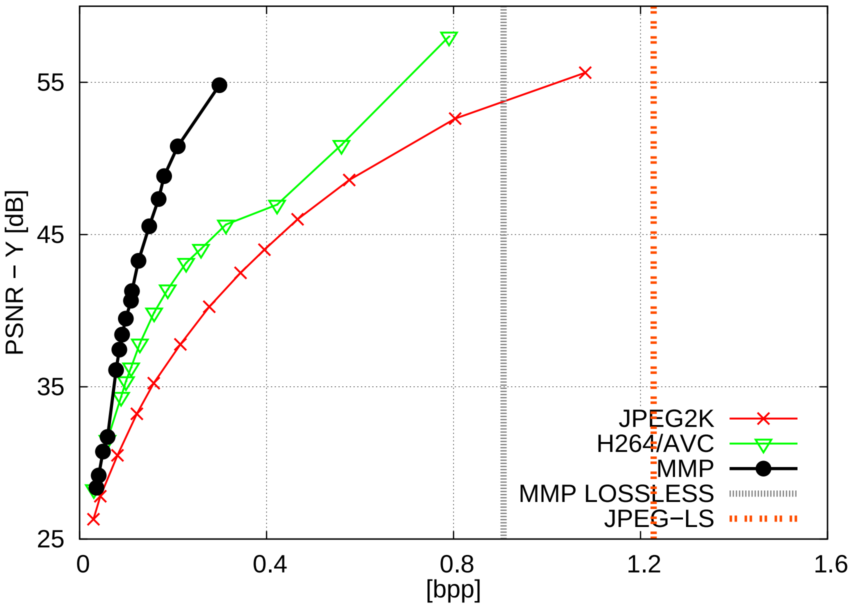
All depth maps were compressed with JPEG2000, H.264/AVC INTRA and MMP. Furthermore, results for the Platelet encoders of the Ballet and Breakdancers sequences[3] are also presented. For the H.264/AVC rate-distortion results, software JM 16.2 [189] was used with the high profile. The images were intra encoded, with a QP varying from 10 to 50. For the JPEG2000, the Kakadu software was used, with Qstep parameter varying from 15 to 0.005. MMP's results for lossless encoding are also presented and compared with lossless coding standard JPEG-LS [66]. This provides us an attainable bound of the compression capability of our algorithm.

The PSNR results for depth map coding are depicted in Figures F.1 to F.3. For the encoding of depth values, MMP exhibits an outstanding performance, reaching almost 10dB of gain over the other tested encoders. Moreover, MMP is also a good alternative for lossless coding of depth map, providing better results than the standard lossless encoder, JPEG-LS. MMP is able to efficiently preserve the PSNR of the high-frequency structures present in the depth values images, which represent the boundaries of the objects. Since inside the object there is no or little fluctuation of the depth value, several smooth structures can be found, resulting in a low variety of patterns in the image. MMP dictionary adaptation scheme benefits from this low variety of patterns, providing a very efficient encoding of the depth images, as can be seen on the presented results.

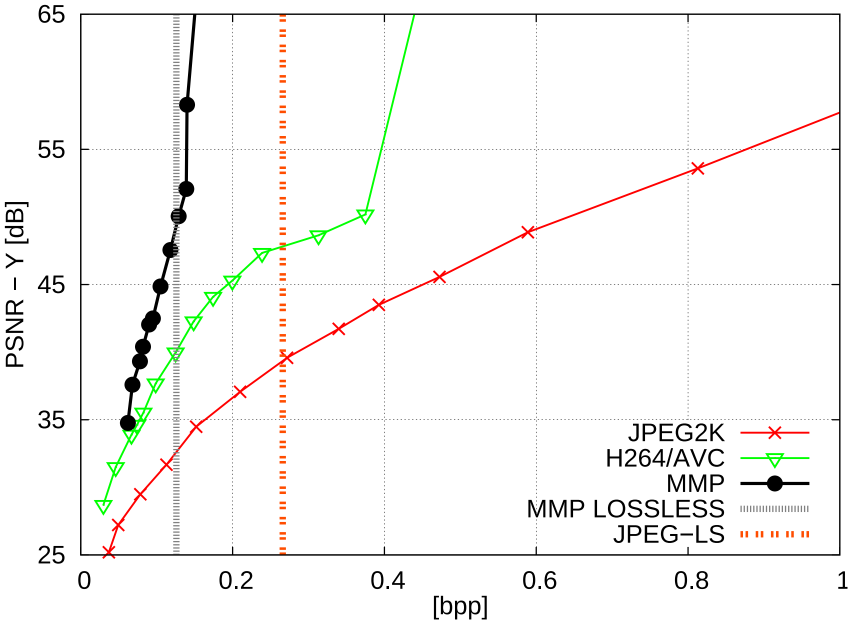
[1] http://vision.middlebury.edu/stereo/

[2] http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/

[3] http://vca.ele.tue.nl/demos/mvc/PlateletDepthCoding.tgz

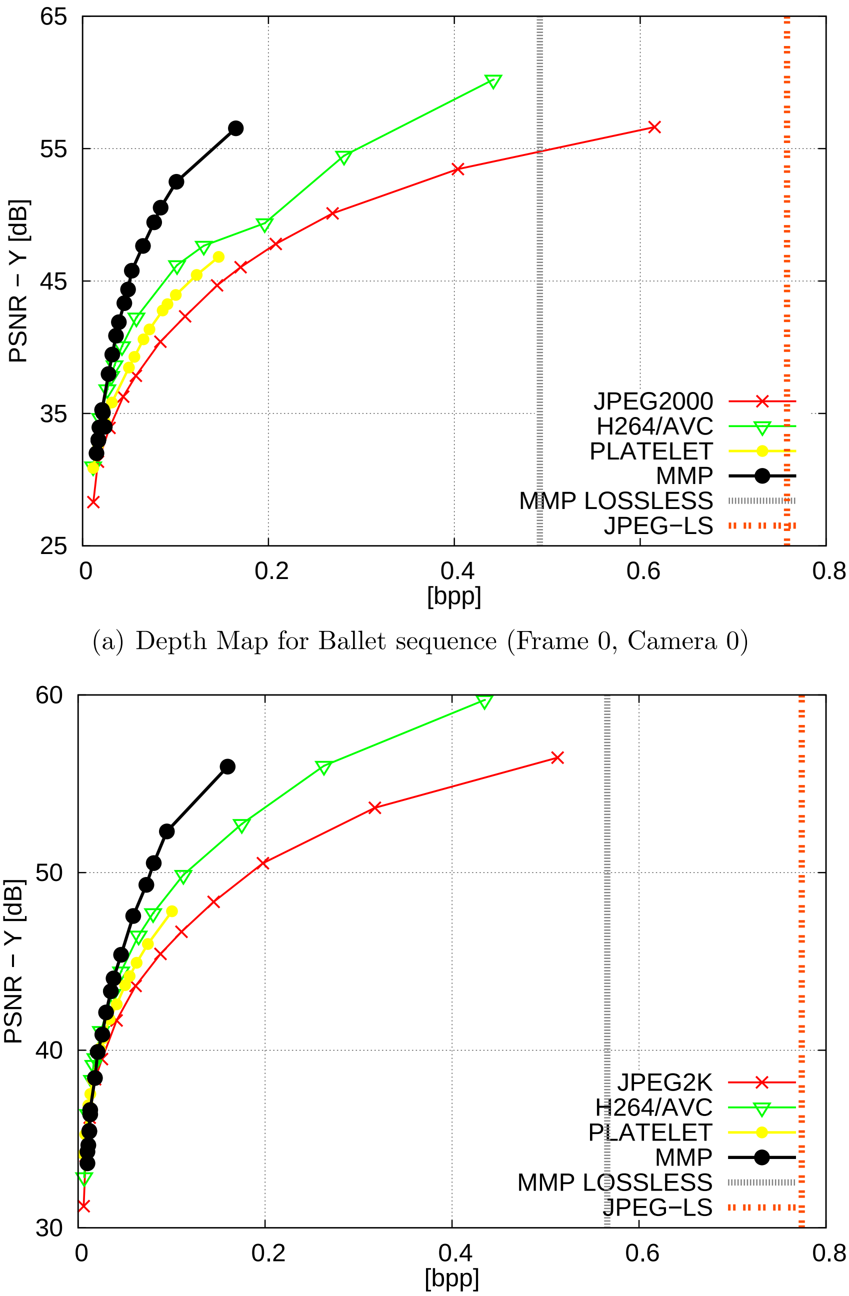


(a) Disparity map for Teddy image



(b) Disparity map for Tsukuba image

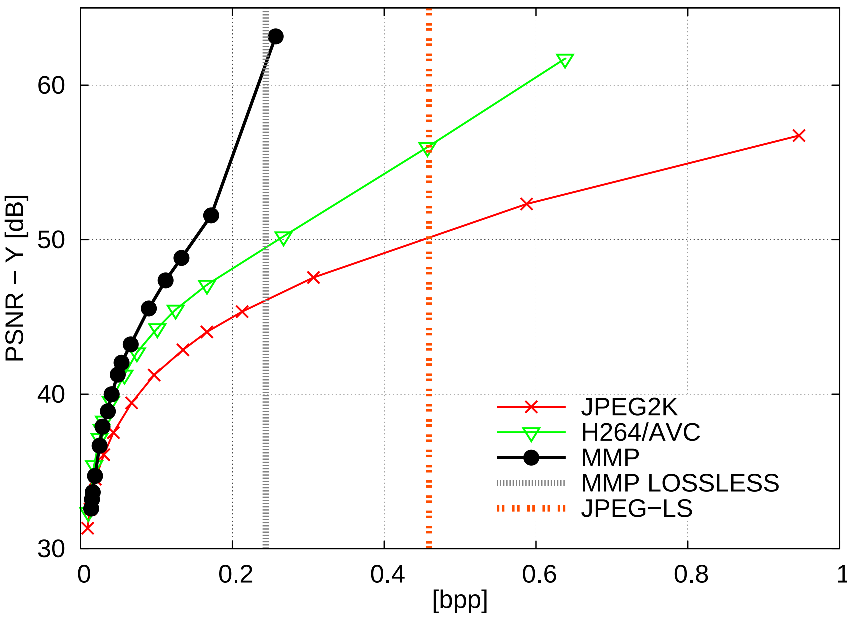
Figure F.1: Rate-distortion for several encoders using disparity maps with ground truth. The vertical lines represent the rate achieved with lossless compression.



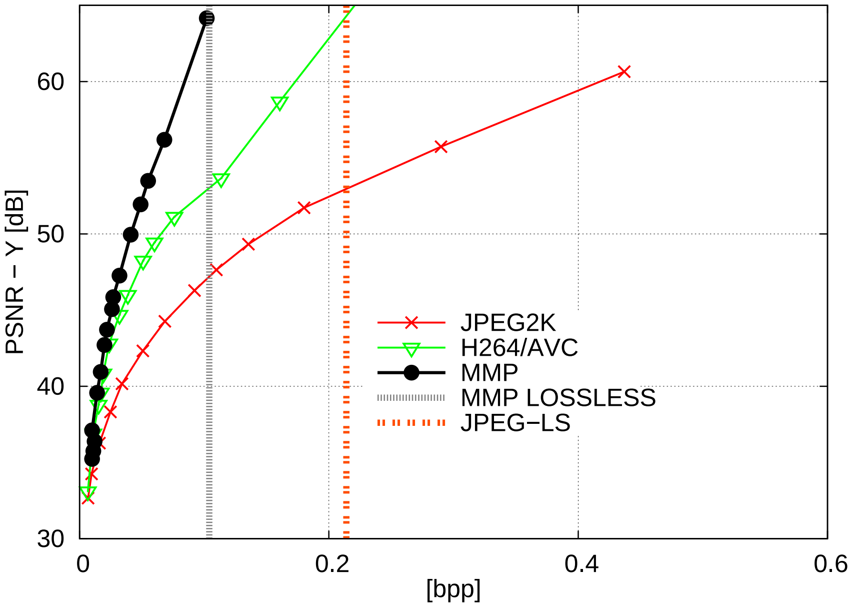
(a) Depth Map for Ballet sequence (Frame 0, Camera 0)

(b) Depth Map for Breakdancers sequence (Frame 0, Camera 0)

Figure F.2: Rate-distortion performance for coded depth maps with non-aligned camera arrangement.



(a) Depth Map for Book Arrival Sequence (Frame 0, Camera 10)



(b) Depth Map for Champagne Tower Sequence (Frame 0, Camera 37)

Figure F.3: Rate-distortion performance for coded depth maps with horizontal alignment.

Alternative objective depth map measurements

The PSNR value is a common measurement used for image evaluation, since it is related to the human visual perception. Nevertheless, depth maps will not be visualized, but yet will be used for the rendering process. Therefore, the PSNR values can be deceiving and do not always reflect how the algorithm will reconstruct the image [78, 80]. Here we present other objectives measurements, that aim to evaluate depth map compression and are not based on the human visual system. Three different distortion criteria were used: the percentage of wrong disparity values [81], the Hausdorff distance [78], and the SSIM index [83].

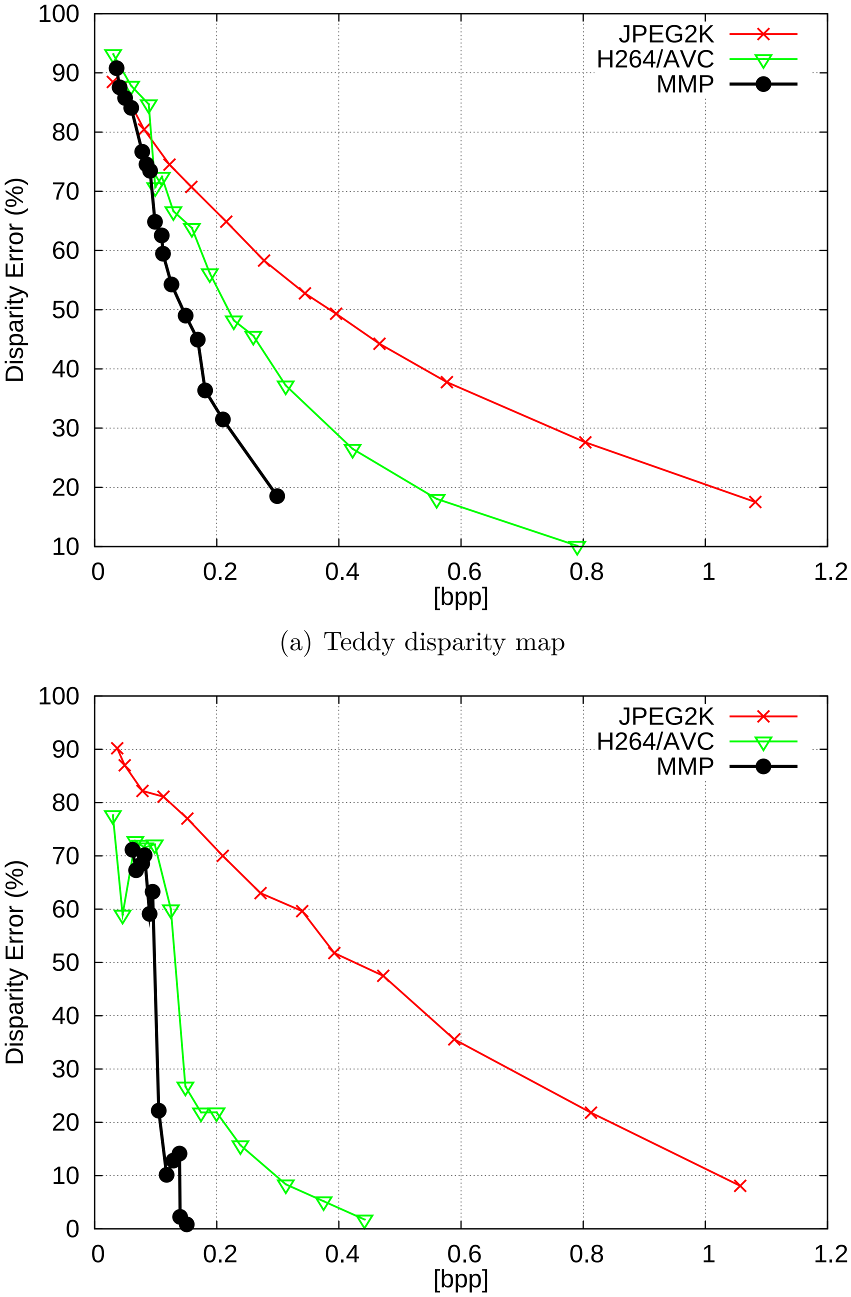
As can be seen in Figure F.4, the percentage of disparity errors shows that disparity maps coded with MMP have fewer errors. The total number of errors decrease are more rapidly when coding disparity maps with MMP, than with the other two approaches. At very low bitrates, all algorithms have a bad performance, however from middle to higher bitrates, MMP presents a clear advantage, producing around half the total errors of JPEG2000 and H.264/AVC.

The Hausdorff Distance is an MSE-like measurement for 3D models, that defines the distance between two nonempty sets. The depth maps represent the vertices of 3D meshes, and two surfaces can be defined: one using uncoded depth data and the other with coded depth data. The geometrical distortion between these two meshes can be given by the average RMS value of the Hausdorff distance over the complete mesh surface. Therefore, the lower the Hausdorff Distance, the closer the coded depth map is to the reference value. A tool name MESH [190] was used to obtain the values for the coded depth maps.

Figure F.5 shows the Hausdorff Distance for the first frame of the Microsoft sequences. The efficiency of the platelet approach is explicit, especially at low or medium bitrates. MMP also performs well at middle rates, and even outperforms all the tested encoders at higher rates. However, the Hausdorff Distance indicate a weakness of MMP's dictionary-based approach: the relatively bad performance at low bitrates. At such rates MMP is not able to grow its dictionary with many patterns, therefore suffering from serious blockiness.

Beyond the mentioned measurements, we propose here the use of the Structural Similarity Index (SSIM) as a depth map quality assessment tool. The SSIM indicates if the structure of the image is preserved, since the structural component of an image is a factor that draws the attention of a human being. In the case of depth maps, structures are also important, since they are usually related to the objects on the scene, that will shift position during the warping operation.

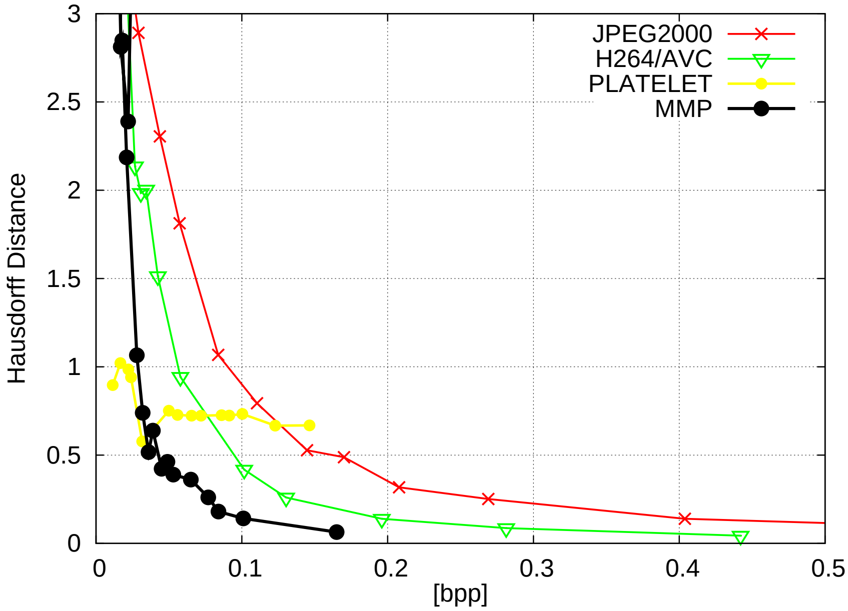
For the case of the depth maps from horizontally aligned camera arrangement, the conclusion that can be drawn analyzing Figure F.6 is similar to the ones made for



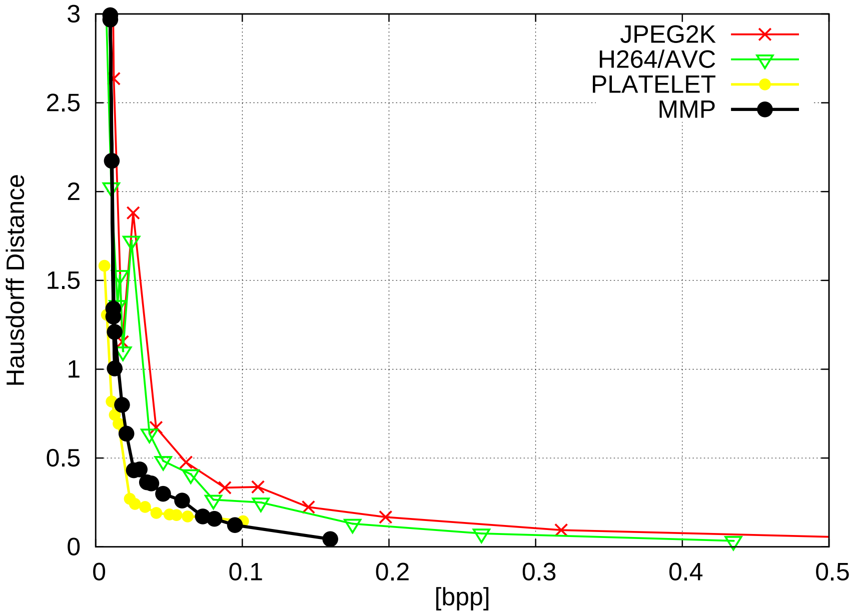
(a) Teddy disparity map

(b) Tsukuba disparity map

Figure F.4: Percentage of different pixels between the coded disparity maps and the ground truth. The noisy measurement for the Tsukuba image is due to the sparse histogram characteristic of the depth map.

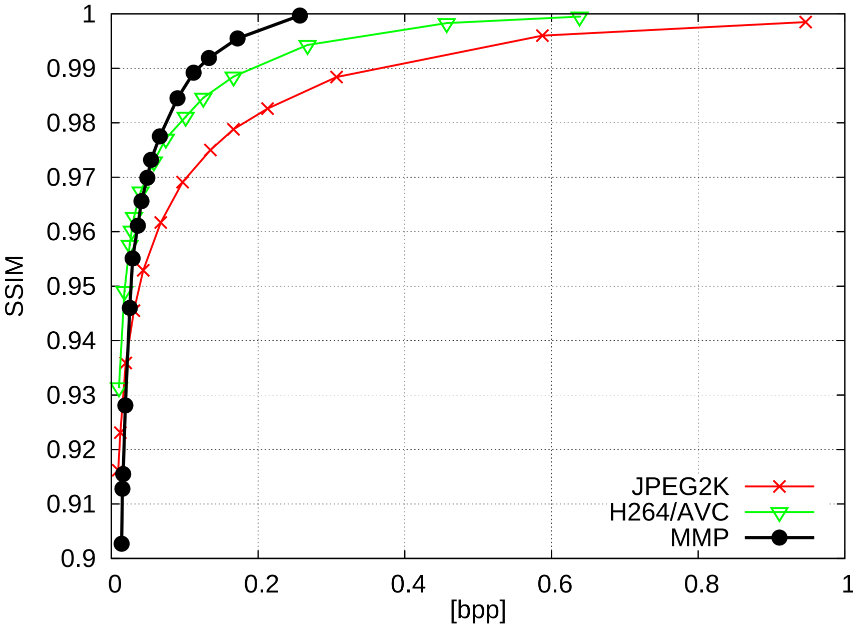


(a) Depth Map for Ballet sequence (Frame 0, Camera 0)

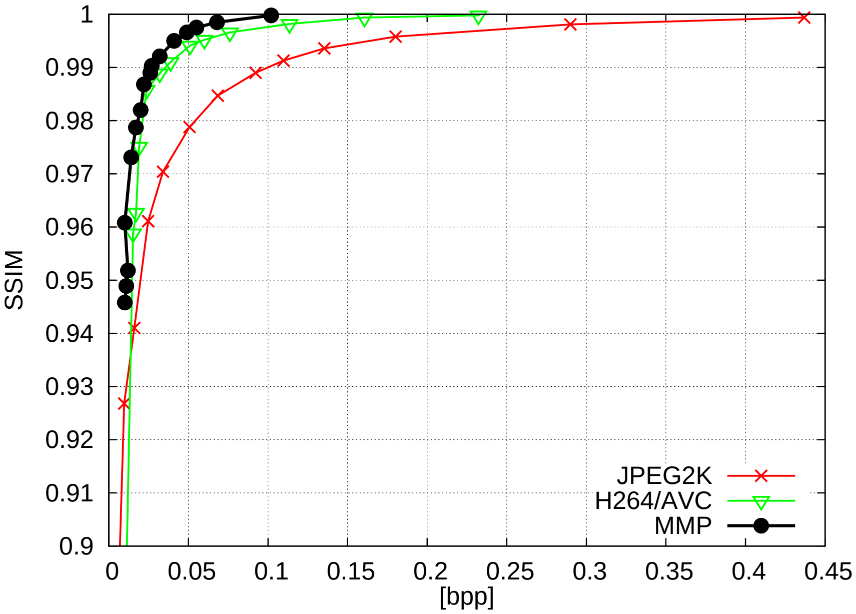


(b) Depth Map for Breakdancers sequence (Frame 0, Camera 0)

Figure F.5: Measurement of the Hausdorff distance of coded depth maps with non-aligned camera arrangements.



(a) Depth Map for Book Arrival Sequence (Frame 0, Camera 10)



(b) Depth Map for Champagne Tower Sequence (Frame 0, Camera 37)

Figure F.6: SSIM of coded depth maps with horizontal camera alignment.

the other measurements. Due to its efficient encoding scheme, flexible segmentation and high frequency preservation property, MMP is able to code depth maps with higher fidelity, preserving the structure of the objects in the depth map, as indicated by the high SSIM value.

Reconstructed images using coded depth map and uncoded texture

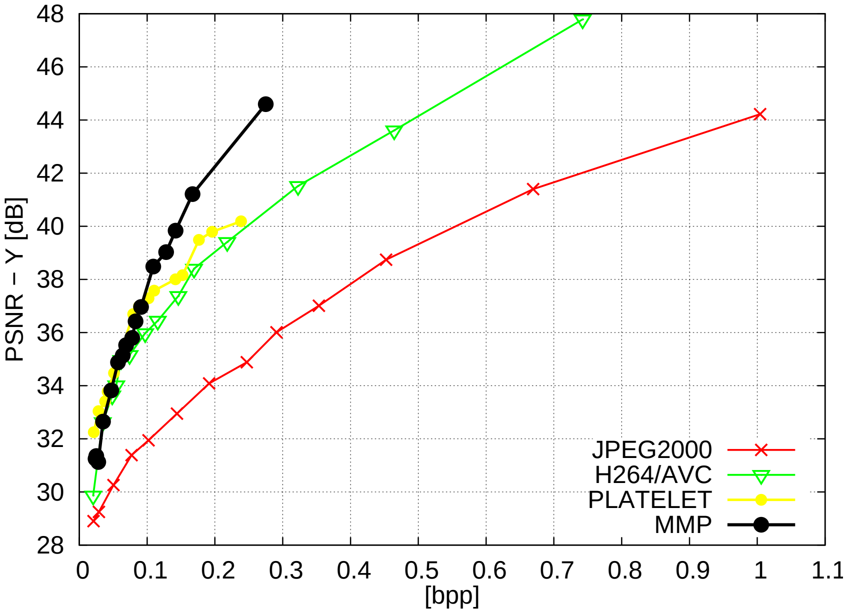
It is still recommended to always evaluate not the coded depth maps' artifacts, but the rendering artifacts resulted from coded depth data [78, 80]. In our tests we have used uncoded texture and the respective coded depth maps of both views to synthesize a central view. Objective measurements were done by comparing the rendered view with coded depth with the same rendered view with uncoded depth. Subjective analysis were also made. To measure the quality of the rendered virtual view with encoded depth maps, a synthetic image was rendered at the same location and orientation of a selected camera.

In Figure F.7, an objective measurement is performed by showing the PSNR values between the virtual view reconstructed with uncoded depth data and the rendered views with encoded depth maps. MMP suffers from blockiness artifacts, and performs similar to H.264/AVC in low bitrates, but at medium to high bitrates it is able to capture the images structure efficiently and outperforms all the tested algorithms.

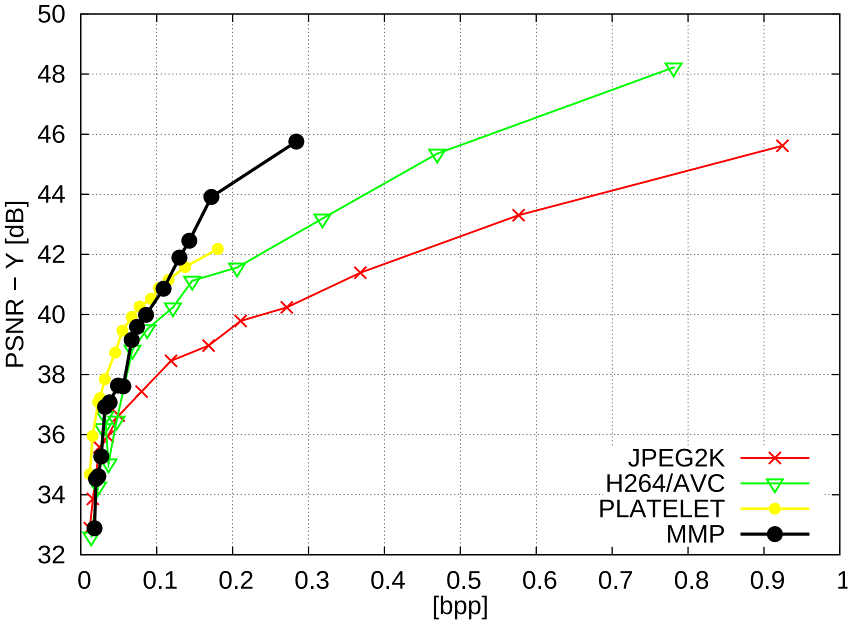
Figure F.8 show artifacts generated by each encoder. JPEG's ringing artifacts seriously compromise the reconstructed view, and all around the ballerina's boundary a mixture of background and foreground information can be seen. A better reconstruction can be achieved with the H.264/AVC encoder, but still many artifacts can be seen around the ballerina's finger and along her limits, due to the quantization of high frequency information. In the case of the Platelet encoded depth maps, we notice a better preservation of the objects boundaries, although we can still see some artifacts, such as the lump on the ballerina's back and the dislocated finger. MMP encoded depth maps generate rendered views with fewer artifacts than H.264/AVC-Intra and JPEG2000, better preserving shapes and object boundaries, similar to the Platelet encoder, but achieving the highest PSNR. Although we can still see some boundaries issues, MMP is the algorithm that came closest to the synthesized view, due to its high frequency preservation property, that is, the preservation of edge information.

Evaluating the reconstruction software with coded depth data

One important issue with evaluating the performance of a system using reconstructed views is that the DIBR algorithm directly influences the final result. Ac-

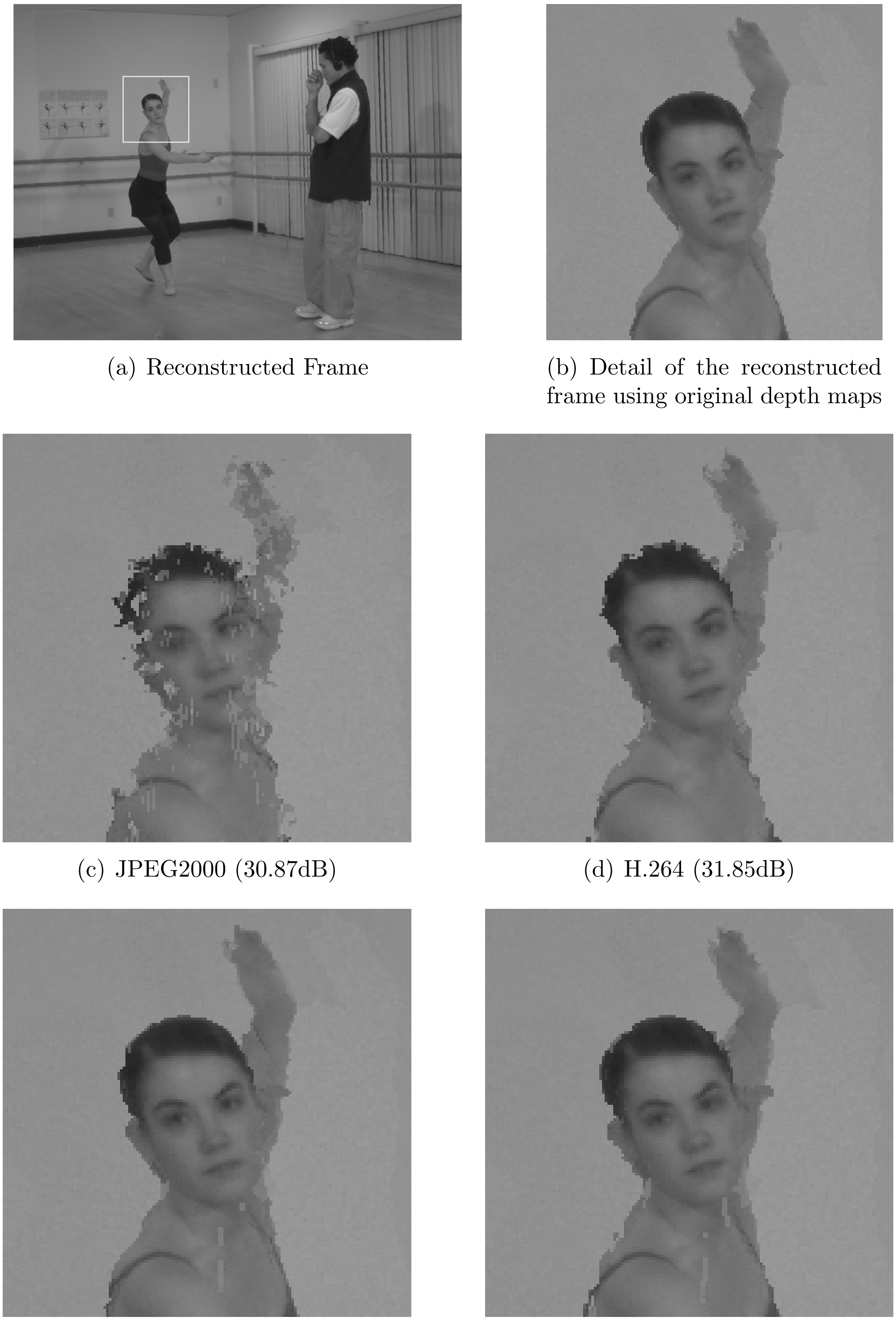


(a) Reconstructed view for Ballet sequence (Frame 0, Camera 4)



(b) Reconstructed View for Breakdancers sequence (Frame 0, Camera 1)

Figure F.7: Rate-distortion performance for reconstructed views using uncoded texture and coded depth maps.



(a) Reconstructed Frame

(b) Detail of the reconstructed frame using original depth maps

(c) JPEG2000 (30.87dB)

(d) H.264 (31.85dB)

(e) PLATELET (31.76dB)

(f) MMP (32.18dB)

Figure F.8: Reconstruction of the first frame of the 4th camera of *Ballet* sequence, using the original views of the 3rd and 5th cameras and their respective encoded depth maps. Figures (c)-(f) show details of the reconstruction with each depth map encoded at 0.075bpp.

tually, the synthesis algorithm creates an interdependency between several elements of the 3D chain of operation. If the coding stage takes into consideration how the information will be used for view synthesis, savings can be achieved by modifying the algorithm accordingly. For instance, KIM *et al.* [75] use an approximation of the warping error using specific camera parameters in the rate-distortion algorithm, instead of the depth distortion. The presented results show an improvement of almost 1 dB on average for several sequences. SILVA *et al.* [167] also use the synthesis algorithm to determine the distortion of the encoding options, and then optimize the encoding parameter selection using a genetic algorithm.

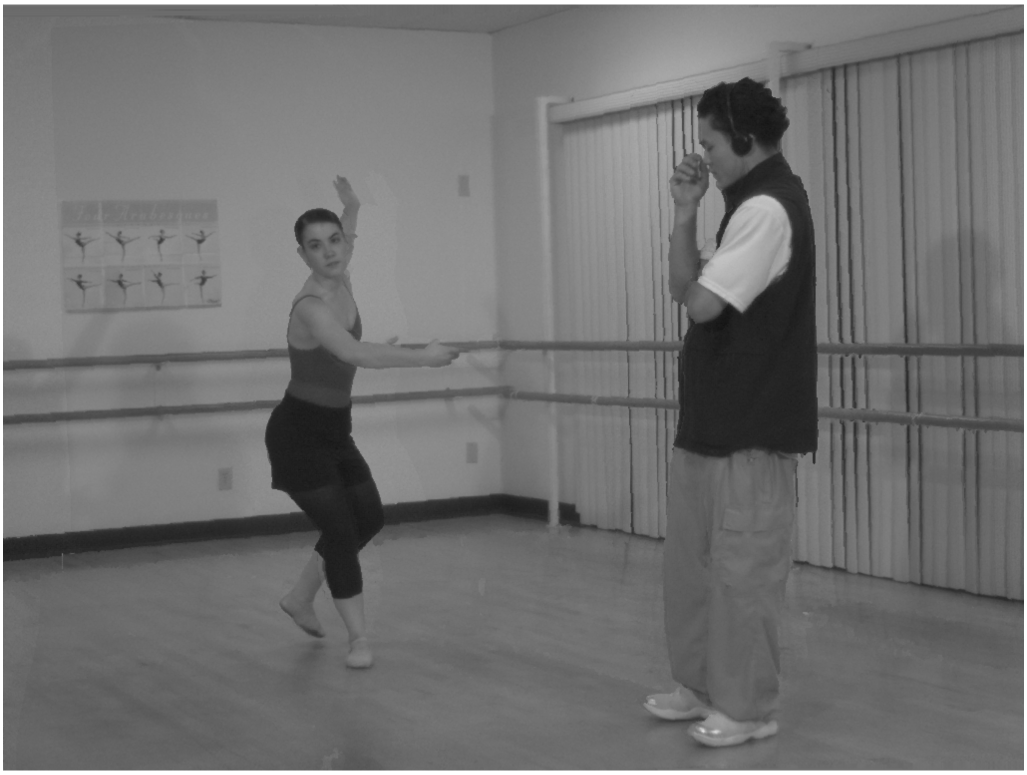
The effects of the synthesis algorithm are explicit in Figure F.9. The PSNR values were obtained comparing the obtained virtual views with the actual view captured by the camera at the synthesis position. One can notice that the VSRS software shows a higher PSNR value, although it still presents some reconstruction artifacts, such as the noticeable ghost effect behind the ballerina and some occlusion problems at the bottom right corner of the frame. Nevertheless, software parameters can be changed to mitigate this problems, and configuration files are provided by the MPEG group for their test sequences [188].

Figure F.10 shows the rate-distortion performance for the same coded depth map, only using different synthesis algorithms. This graphic illustrates that the choice of synthesis algorithm can influence the analysis of the obtained results for the encoding algorithm, and must be taken into consideration, when evaluating depth maps.

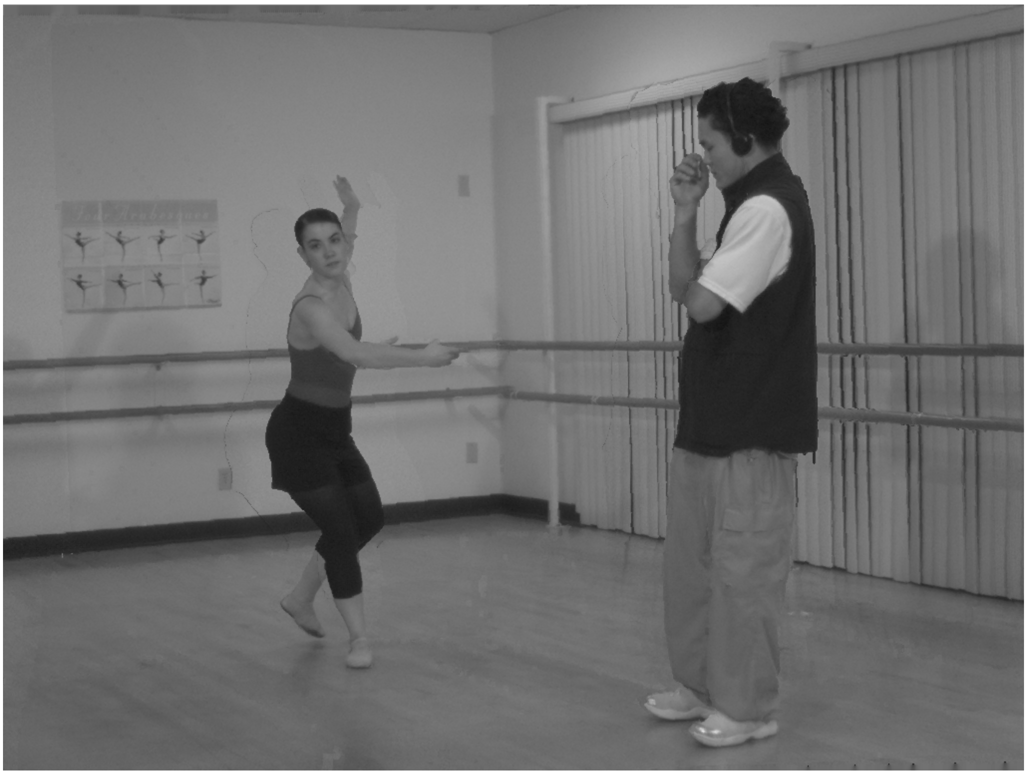
Reconstructed images using coded depth map and coded texture

The impact of the coded depth data in reconstructing virtual views with coded texture data is evaluated. For our experiments, we use the texture coded with the H.264/MVC algorithm. MPEG provides the best combination of QP parameters for an optimal bitrate allocation between texture and depth in several documents [191, 192], and the chosen values are reproduced in Table F.1 for clarity. The rates designated for the depth coding used by the MVC software are the target rates used for the MMP algorithm. An appropriate λ is chosen so that both depth maps coded with MMP do not exceed the total bitrate available.

Figures F.11 and F.12 show the clear advantage of using MMP instead of H.264/MVC for coding depth maps. Gains of more than 2 dB were reported. Notice that the bitrate allocation done by MPEG was already focused on the quality of the synthesized view, that is, the bitrate allocation was done to improve the synthesized view, so usually a fair amount of bits were reserved for depth coding. MMP improves even further the results presented by the MVC software, notably enhancing the edge information, as can be seen in Figure F.13. Notice also that the encoding of depth

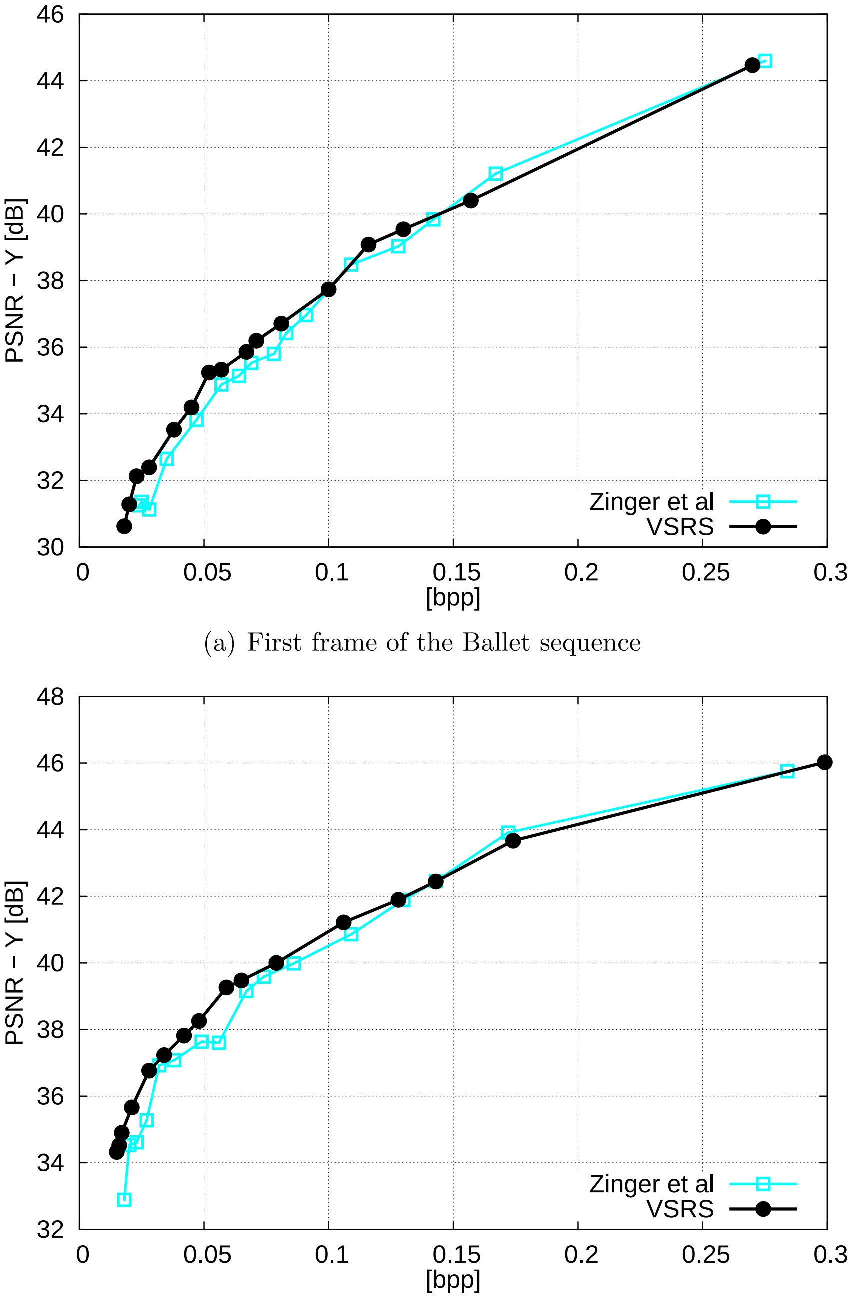


(a) 33.245 dB - DIBR algorithm from Zinger et al [177]



(b) 33.557 dB - MPEG's View Synthesis Reference Software [178]

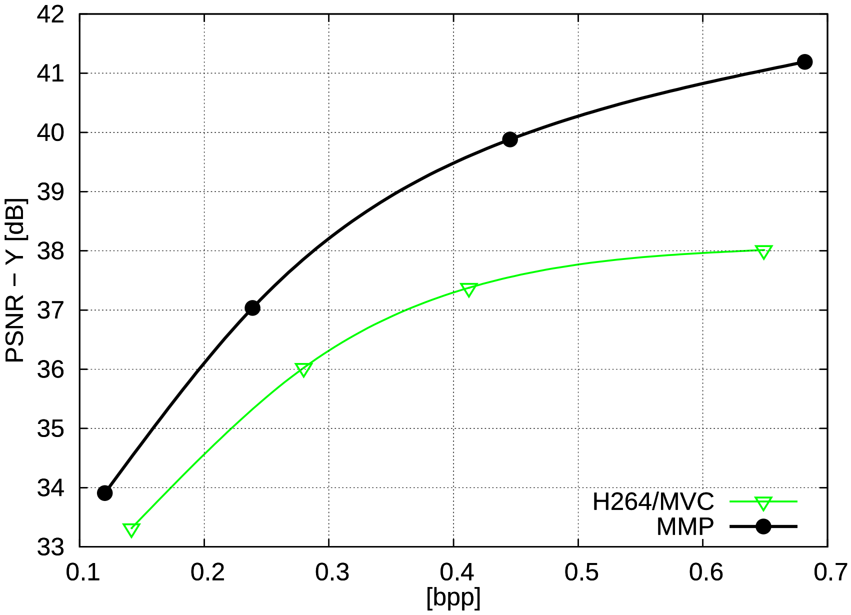
Figure F.9: Reconstruction of the view of the first frame from camera 4, for the Ballet sequence.



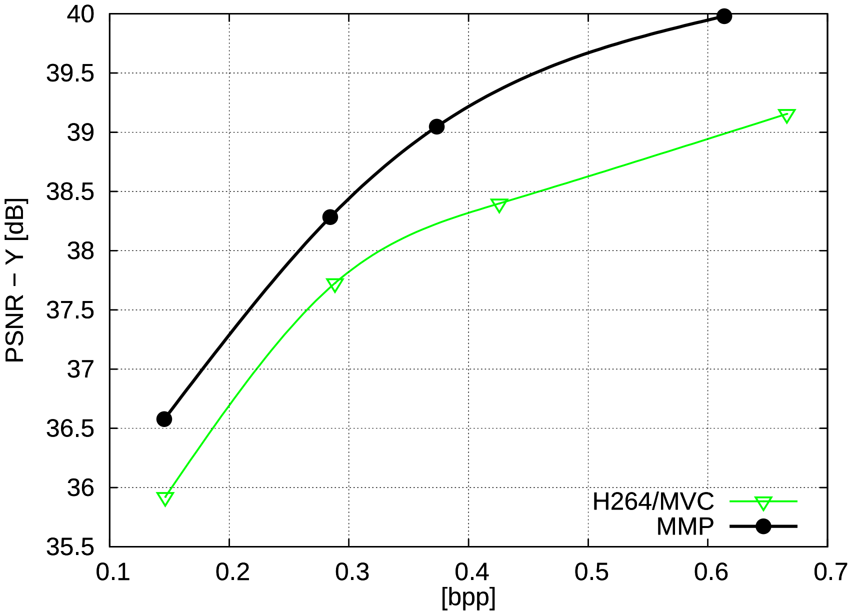
(a) First frame of the Ballet sequence

(b) First frame of the Breakdancers sequence

Figure F.10: Rate-distortion performance of coded depth maps reconstructing view 4 of the Ballet sequence, using different synthesis algorithms. This example illustrates the interdependency between the synthesis algorithm and the coding procedure for depth maps.

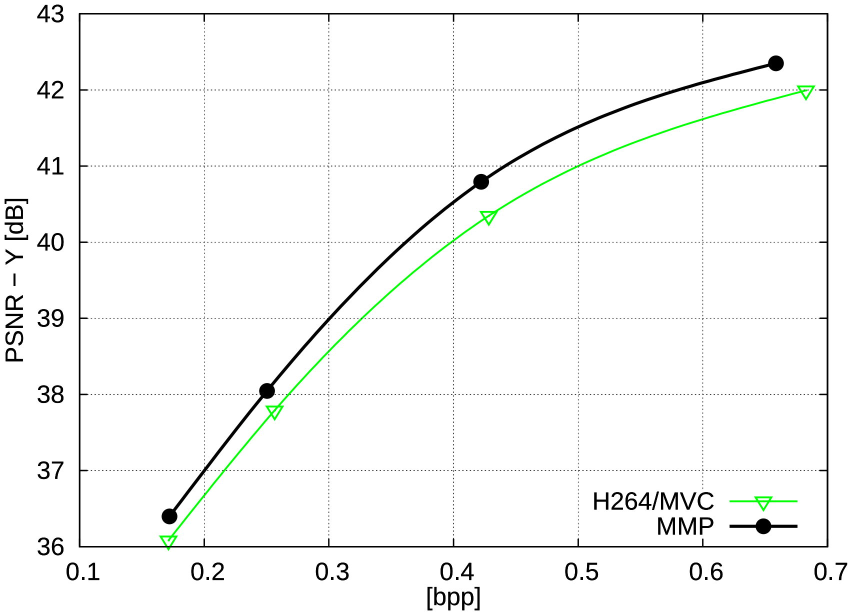


(a) Reconstructed view 4 of the Ballet sequence (frame 0)

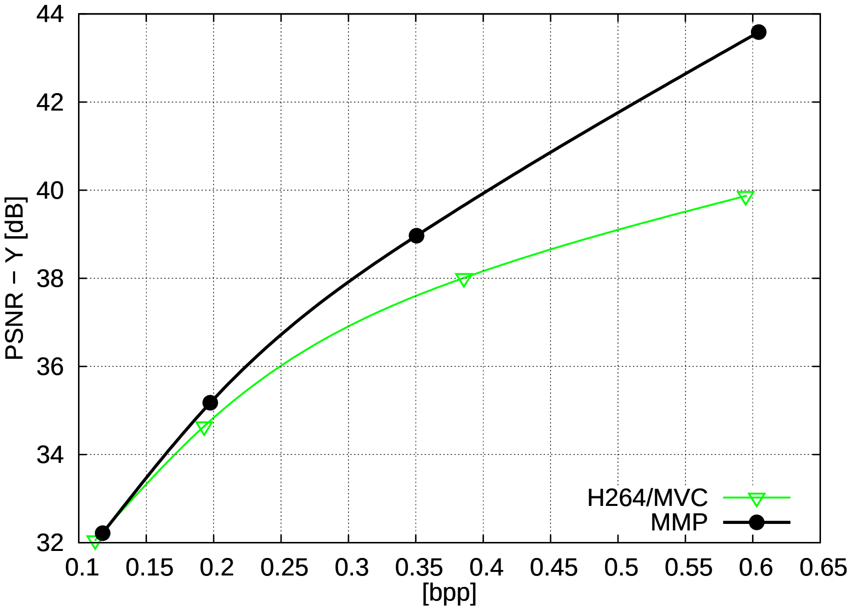


(b) Reconstructed view 1 of the Breakdancers sequence (frame 0)

Figure F.11: Comparison of coded depth maps using in the reconstruction with coded texture as well. The rates used for this experiment were similar to the rates obtained from the Book Arrival sequence [192], since both sequences' frame have the same size.

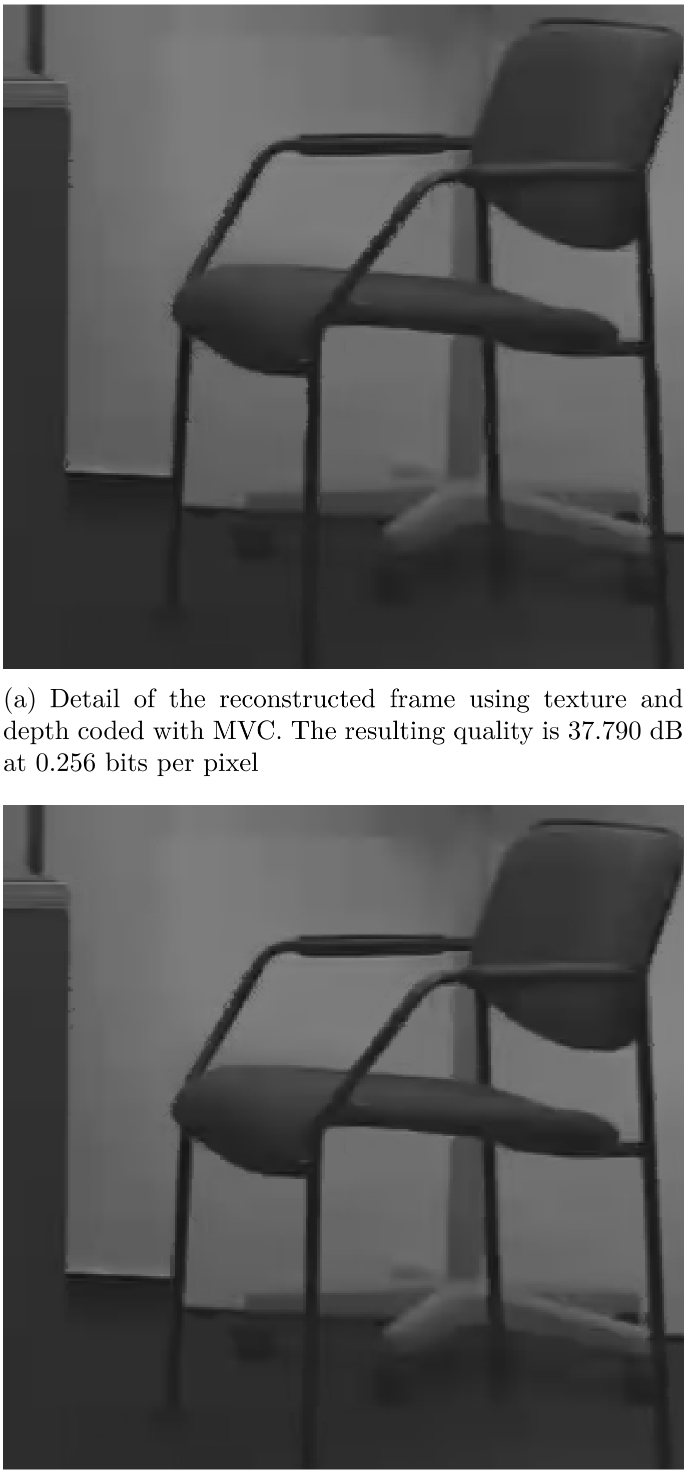


(a) Reconstructed view of camera 9 from Book Arrival sequence (frame 0)



(b) Reconstructed view of camera 40 from Champagne Tower sequence (frame 0)

Figure F.12: Comparison of coded depth maps using in the reconstruction with coded texture as well.



(a) Detail of the reconstructed frame using texture and depth coded with MVC. The resulting quality is 37.790 dB at 0.256 bits per pixel

(b) Detail of the reconstructed frame using texture coded with MVC and depth coded with MMP. The resulting quality is 38.046 dB at 0.250 bits per pixel

Figure F.13: Subjective comparison of coded depth maps using in the reconstruction with coded texture as well, for view of camera 9 from Book Arrival sequence (frame 0). The high frequency details of the chair (legs and arms) are better preserved with MMP.

Table F.1: Experimental set-up for multiview image coding, taken from the MPEG documents [188, 191, 192]. Since Ballet and Breakdancers Sequence have the same resolution as the Book Arrival sequence, the same rate-distortion points were used, and a similar optimization for the best QP combination was performed for those sequences. The table also shows the chosen cameras, left and right views, and the central view, which will be used for reconstruction

Sequences	View(L-C-R)	Target Bitrate (Mbps)				QPT				QPD			
Ballet	(3-4-5)	0.3	0.5	0.75	1.25	38	34	30	24	36	26	22	22
Breakdancers	(0-1-2)	0.3	0.5	0.75	1.25	36	32	30	26	34	26	22	22
Book Arrival	(10-9-8)	0.3	0.5	0.75	1.25	36	34	28	26	42	34	34	26
Champagne Tower	(39-40-41)	0.3	0.5	0.8	1.5	42	38	34	28	44	36	24	22

information with MMP is done independently, while the MVC software exploits the inter-view correlation, being able to compress even further the depth information. However, the introduced artifacts compromise view reconstruction, something that the MVC algorithm does not take into consideration.

F.2 Edge-aware coding of depth maps

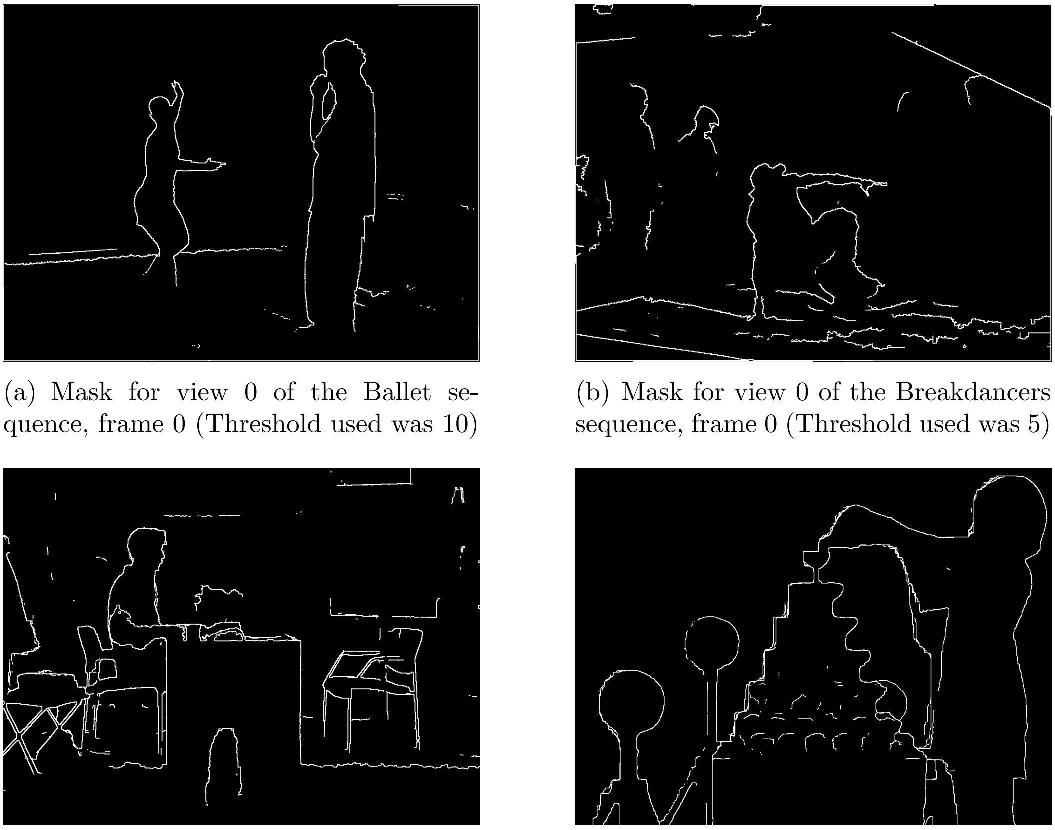
Edges in the depth map represent object's boundaries and are usually well-defined. Coding of depth maps usually introduces artifacts at such edges, and even encoders such as MMP may, eventually, alter the depth values at objects border reconstruction. Many proposals have been made for depth map coding, that take special care at edge areas around objects [76, 84, 85].

We propose a method to improve visual reconstruction of synthesized views that use coded depth maps by adding a restriction to the compression of the depth information. A process of edge detection identifies the critical areas of a depth map that can generate the boundary artifacts on the reconstructed view. Then these areas are marked as areas where its distortion cannot exceed a given threshold.

F.2.1 Edge identification algorithm

The first step of the proposed solution is to identify the boundaries of objects that are in the foreground. A 3 × 3 analysis window is used, and the distribution of the pixels inside this window is analyzed. If the population set of the analysis window contains only elements of the same object, their depth values are probably very similar (since we do not have high variations of depth inside the same object), so the standard deviation of the statistical distribution is small. On the other hand, if the window contains both elements from foreground and background, their statistical distribution should reveal a large standard deviation, since we would have a distribution with two peaks, centered at the background mean depth and the foreground mean depth. By comparing the standard deviation with a pre-determined threshold, we can identify if the window contains an edge or not. We then mark the central pixel of the window as an edge pixel, but at this point we still don't know whether the pixels belongs to the foreground or to the background.

Then, afterwards, we determine if the edge pixel belongs to the background (it has a larger depth) or the foreground (it has a smaller depth), by comparing the pixel value with the distribution mean. We then mark all the foreground pixels that belong to an edge and analyze their 8 closest neighbors, adding neighbors with similar depths to the foreground object boundary mask. In this manner, we increase the mask by one pixel, but only in the foreground direction.



(a) Mask for view 0 of the Ballet sequence, frame 0 (Threshold used was 10)

(b) Mask for view 0 of the Breakdancers sequence, frame 0 (Threshold used was 5)

(c) Mask for view 10 of the Book Arrival sequence, frame 0 (Threshold used was 5)

(d) Mask for view 37 of the Champagne Tower sequence, frame 0 (Threshold used was 7)

Figure F.14: Edge masks obtained with the proposed edge detection algorithm.

Figure F.14 shows the mask generated by the edge detection algorithm and the captions indicate the respective threshold used to obtain the masks. The threshold was selected as a compromise between the number of detected objects and the rate spent to encode them.

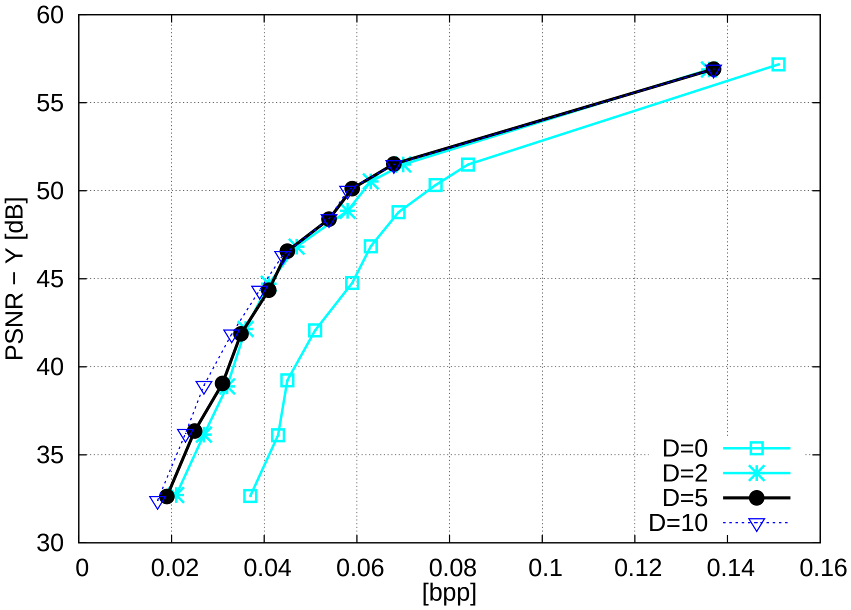
F.2.2 Edge coding restriction

Once the objects' edges are identified, the pixel positions that mark those edges will have an added restriction in the rate-distortion cost calculation. In those regions, distortion values larger than a given threshold are forbidden, forcing the algorithm to encode a block either using a codeword that introduces distortion up to an allowed level at the marked positions, or segmenting the block and coding each half with the according allowed distortion level.

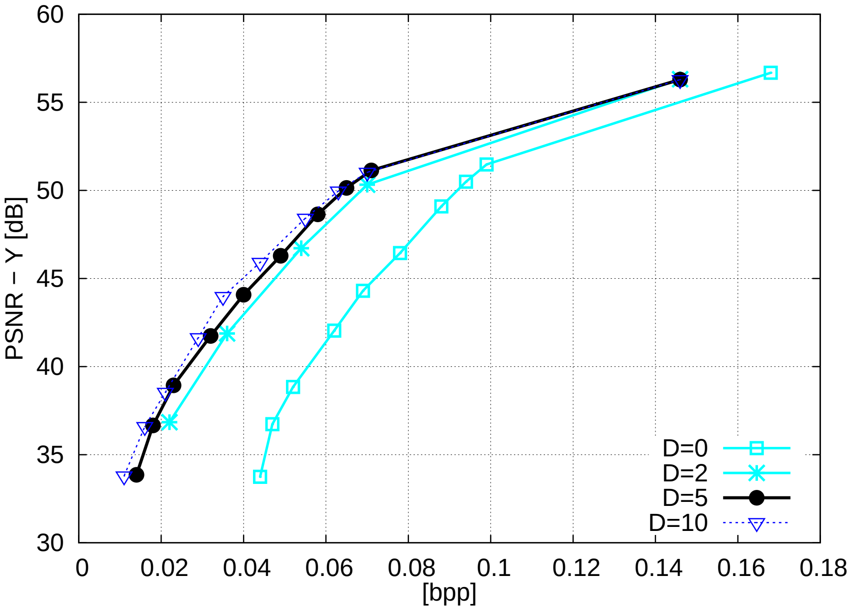
We performed several experiments with different thresholds. By setting the allowed distortion to 0, we perform lossless coding of the edges. However, as can be seen in Figures F.15 and F.16, the added restriction demands an increased bitrate, and a decrease in the overall quality of the depth map. Furthermore, strong blocking artifacts can be seen in the coded depth maps. This condition is alleviated if the allowed distortion threshold is increased. However, thresholds too high do not preserve the borders of foreground objects as desired. We found the difference between the coded value and the original value at the marked edges equal to 5 as a good compromise between added bitrate and allowed distortion.

F.2.3 Experimental results

Subjective results shown in Figure F.17 confirm that views generated with such coded depth maps present very few boundary artifacts, considerably improving the synthesized view. However, the added bitrate does not make up for the use of the technique, in a rate-distortion sense. Figures F.18 and F.19 show the rate-distortion curves for both methods. The low PSNR values from the edge aware procedure are mainly caused by the blocking artifacts of the coded depth maps. Slanted surfaces are coded with smooth blocks with different mean value. This causes an uneven block shift when performing warping with the coded depth values, which results in parts of the image slightly displaced. Although subjectively this might not be a problem, objectively, the PSNR values are affected by the block shift. Nevertheless, one interesting conclusion can be drawn by analyzing the graphics in Figures F.18 and F.19. The performance of both methods are equivalent for high rates. This means that the original algorithm is preserving the edge information, given the amount of bitrate available. It is yet another evidence of the high frequency preservation property of the MMP algorithm, and also that MMP is an appropriate encoder for depth map coding.

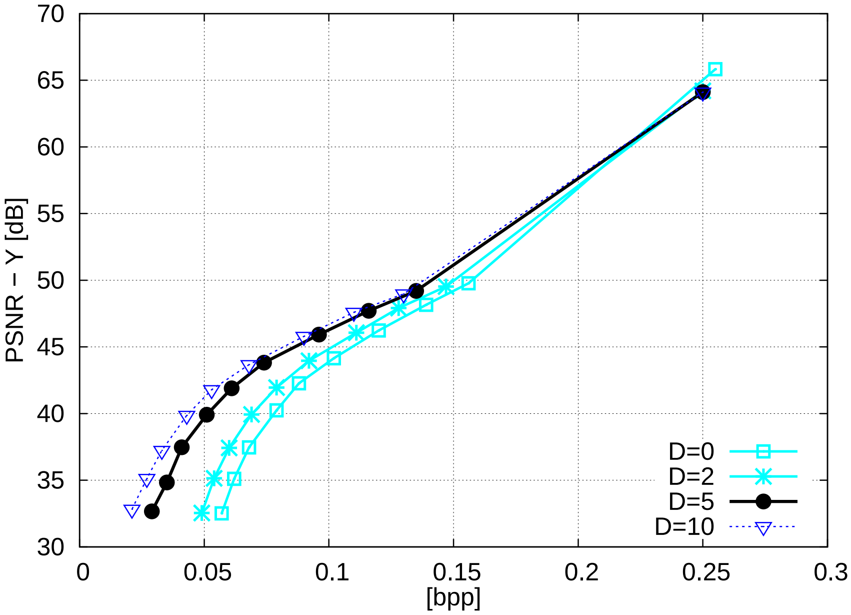


(a) Coded depth maps from the Ballet sequence, view 3, frame 0

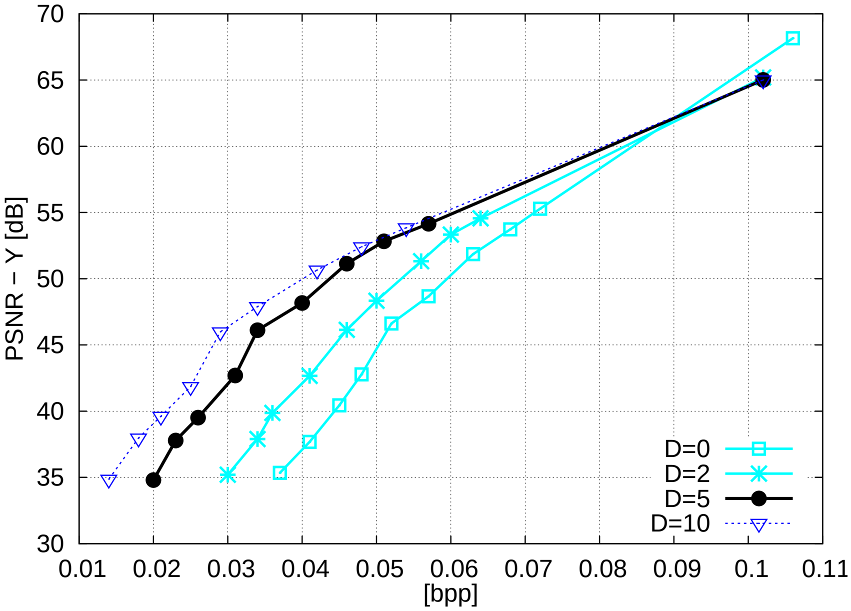


(b) Coded depth maps from the Breakdancers sequence, view 0, frame 0

Figure F.15: Optimization of the allowed distortion for the marked edge pixels, in the edge aware coding of depth maps.

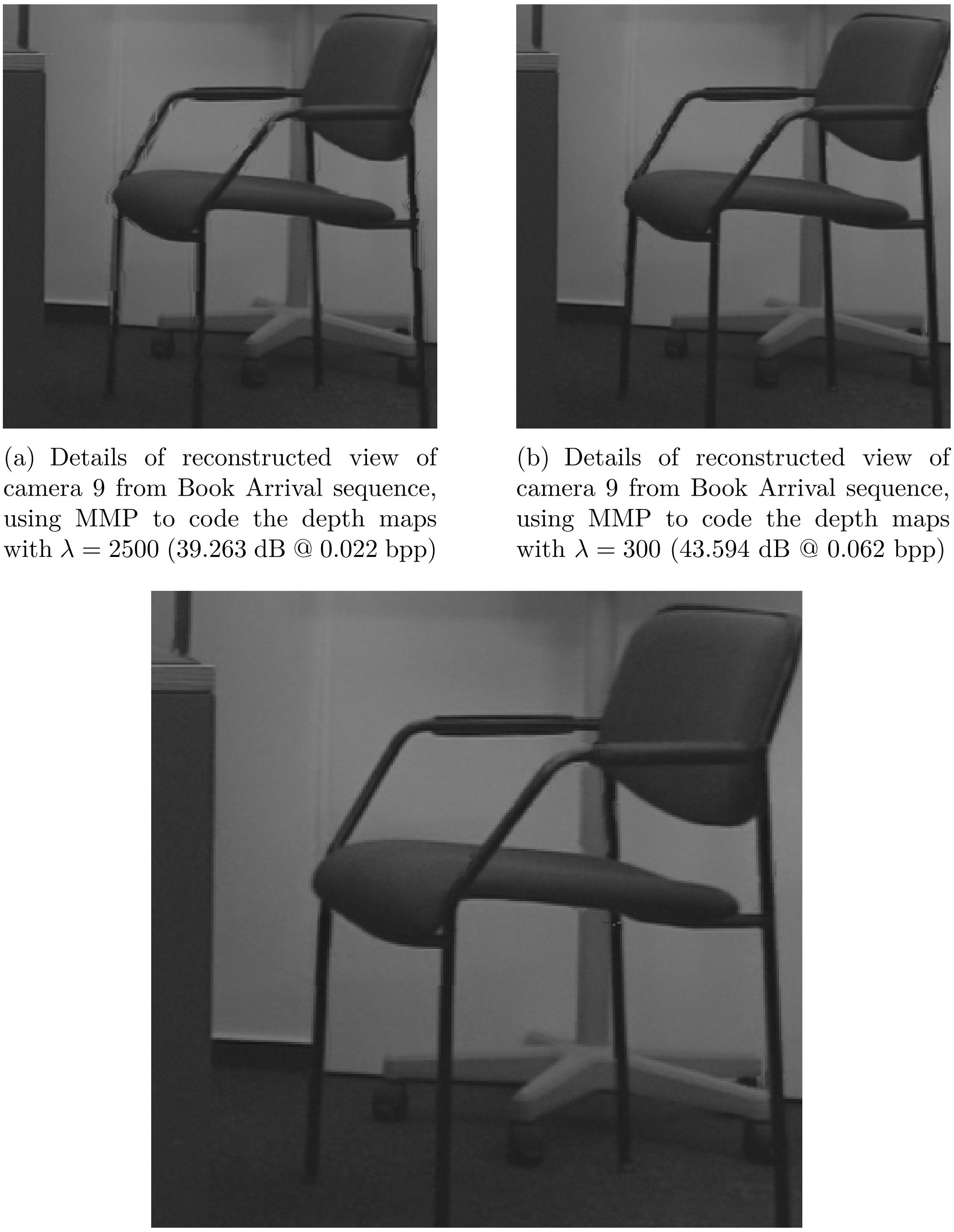


(a) Coded depth maps from the Book Arrival sequence, view 10, frame 0



(b) Coded depth maps from the Champagne Tower sequence, view 0, frame 0

Figure F.16: Optimization of the allowed distortion for the marked edge pixels, in the edge aware coding of depth maps.

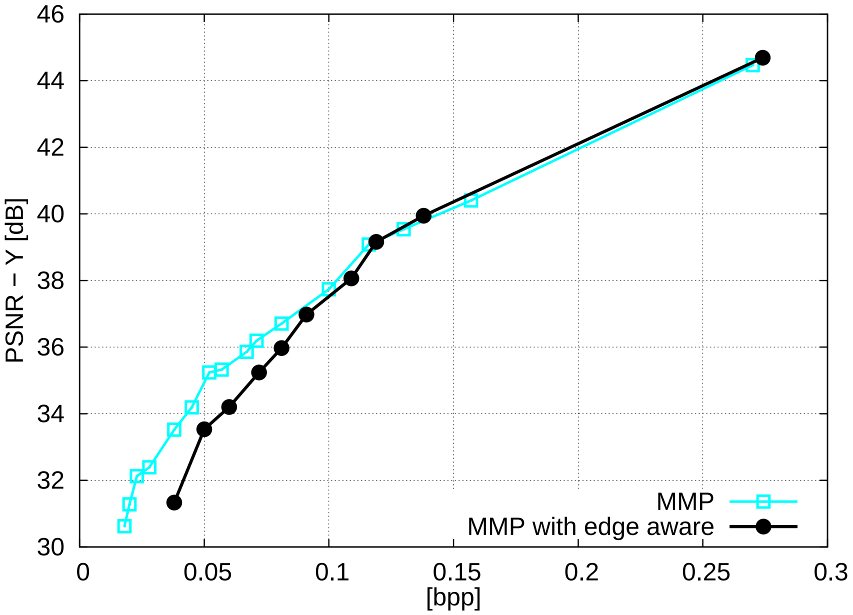


(a) Details of reconstructed view of camera 9 from Book Arrival sequence, using MMP to code the depth maps with $\lambda = 2500$ (39.263 dB @ 0.022 bpp)

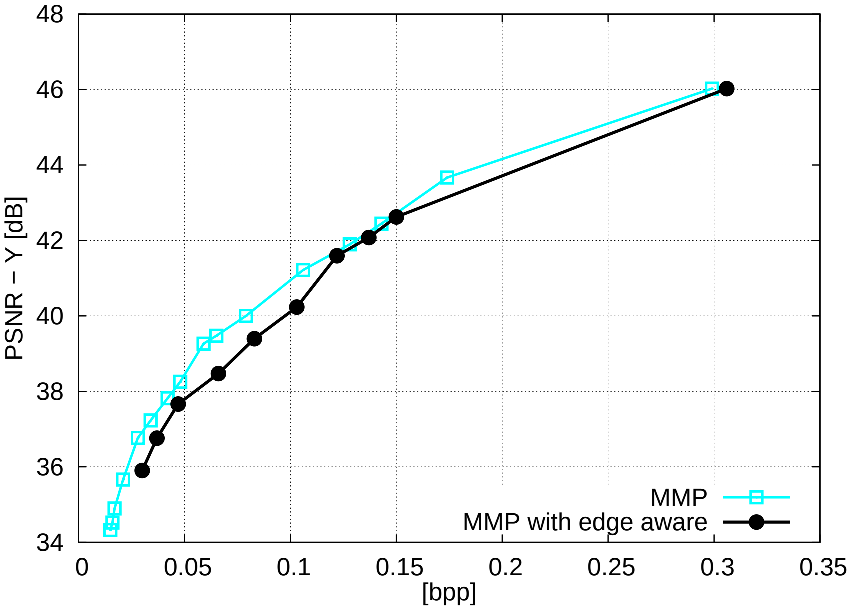
(b) Details of reconstructed view of camera 9 from Book Arrival sequence, using MMP to code the depth maps with $\lambda = 300$ (43.594 dB @ 0.062 bpp)

(c) Details of reconstructed view of camera 9 from Book Arrival sequence, using the edge aware version of MMP to code the depth maps with the same λ (42.665 dB @ 0.061 bpp)

Figure F.17: Comparison of reconstructed frames using the edge aware coding procedure. Notice that the edges of the objects in the view reconstructed with edge aware coded depth data are well preserved.

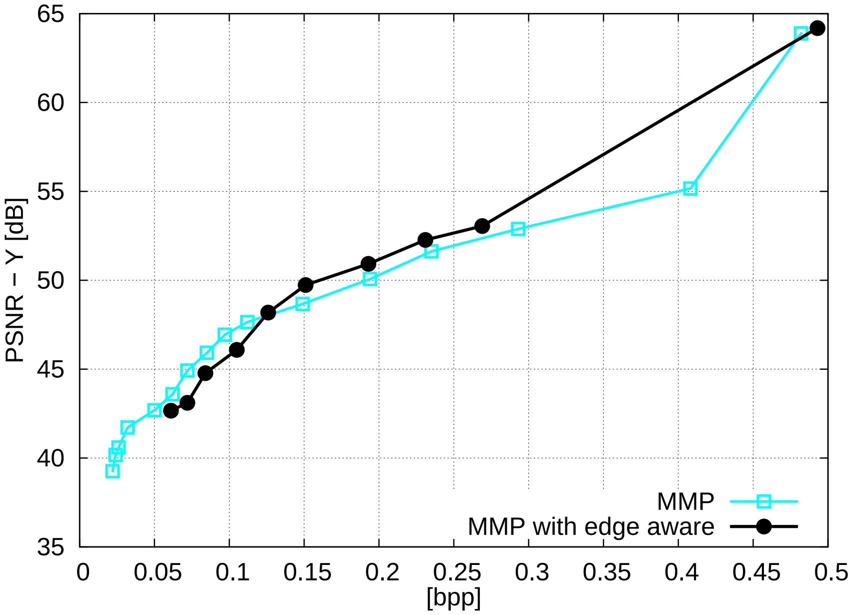


(a) Reconstructed view 4 from Ballet sequence, frame 0

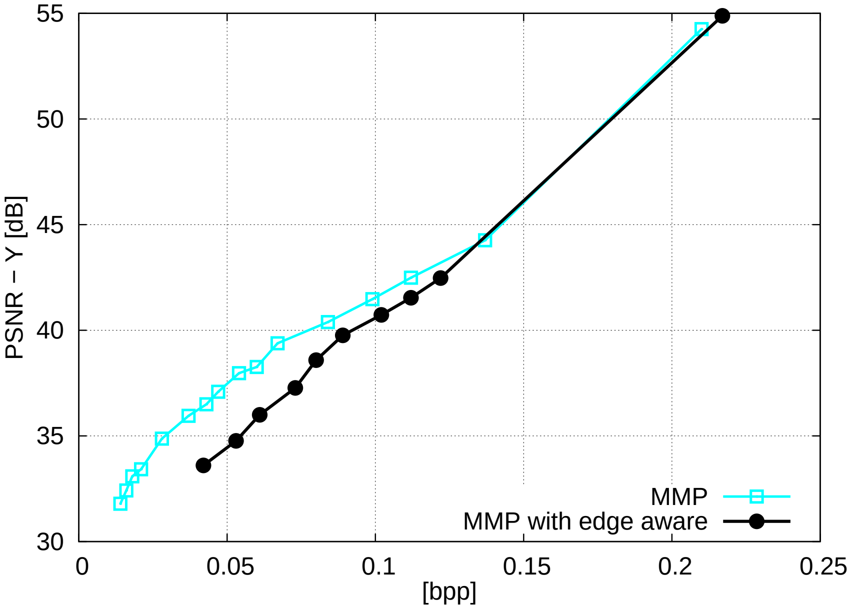


(b) Reconstructed view 1 from Breakdancers sequence, frame 0

Figure F.18: Rate distortion performance of edge aware MMP for the Microsoft sequences.



(a) Reconstructed view 9 from Book Arrival sequence, frame 0



(b) Reconstructed view 38 from Champagne Tower sequence, frame 0

Figure F.19: Rate distortion performance of edge aware MMP for the MPEG sequences.

F.3 Conclusions

In this appendix, MMP coding results for depth maps were presented. MMP has proved to be an effective tool for coding depth data. It is very efficient for middle to high rates, showing an acceptable reconstruction. However, for low bitrates it presents some reconstruction problems due to blockiness. Nevertheless, we advocate that most of the tested algorithms severely compromise the reconstruction at such rates, and that the image synthesis procedure is more effective encoding depth data at higher bitrates.

The proposed technique for edge aware coding of depth maps using MMP presented good subjective results. On the other hand, the edge aware encoding was not rate-distortion efficient, since it spent too much bits coding the edge information. Nevertheless, the results pointed out the importance of edge information of depth maps for the reconstruction algorithm, and is a strong indication that encoders that preserve edge information are well suited for depth map coding. The results confirmed also that the original algorithm indeed preserves the edge information from middle to high bitrates, reassuring MMP's position as an efficient depth map encoder.

MMP also has the advantage of its flexible encoding scheme, being ready to code both texture and depth. In the next appendix, we will conceive a full encoding system for 3D images, where MMP is used to code the depth maps and the texture views. Due to its universal character, MMP is ready to be used for base view and depth map coding, and at all bit rates, from lossy to lossless.

Apêndice G

Texture and depth coding using MMP

SUMMARY: This chapter deals with a multiview coding scenario using only MMP. Section G.2 explores the independent coding of texture and depth map using the MMP. The advantages of using MMP and an optimization scheme for the coding parameters is also described in the corresponding subsections. Section G.3 presents the architecture of a 3D encoder based on MMP that jointly encodes texture and depth, providing an analysis of the advantage of using the warped frame as prediction. The experimental results are presented in Section G.4. Section G.5 concludes this appendix.

G.1 Multiview image coding

Regarding the MPEG "Call for Proposals on 3D Video Coding" (3DVC) technology [11], the present standards such as 2D+depth (MPEG-C Part 3, [70, 71]) and Multiview Video Coding (MVC, as specified by ISO/IEC 14496-10 | ITU-T Recommendation H.264 [9]), do not provide a solution as they cannot give support for N-view auto-stereoscopic displays. They either lack the ability to resolve occlusions or the bitrate requirement for such a large number of views is unacceptable [69].

The new 3D format must be able to transmit an increased number of views and supplementary material (depth maps) through a constrained channel, thus demanding high compression rates. In addition, it has to guarantee that eventual compression artifacts should inflict minimal visual distortion on the final quality of the synthesized views [187]. The development of this new video format will enable the realization of Free Viewpoint TV (FTV), considered one of the most challenging 3D applications [154]. The use of multiple texture and depth signals, together with metadata (e.g., camera parameters), and possibly some additional data (e.g. specular and transparency surfaces), is considered for the new 3D data format reference model by the MPEG [193].

A large number of new algorithms have been proposed for depth-map coding [77–81], and several adaptations in current video standards, such as H.264/AVC or H.264/MVC, have been investigated [74–76]. They all aim at the coding of video and depth maps, targeting the high quality synthesis of virtual views. Most proposals for depth map coding are suited for depth maps only and can not be easily applied to texture coding. On the other hand, adaptations of the common texture coding algorithms, such as in H.264/AVC or in H.264/MVC, for depth map coding do not fully avoid the introduction of coding artifacts that affect the rendering algorithm.

Proposals that try to jointly encode video and depth data have also been published. YEA e VETRO [86] used depth data to warp pixels from neighboring views and to create a synthetic view that is added to the frame buffer structure and used for prediction by the H.264/MVC encoder. The proposed prediction is called View Synthesis Prediction (VSP) mode. Gains have been reported, especially for low to middle bitrates. A different scheme was presented by KITAHARA *et al.* [87], that codes just the residual frame obtained from the difference between the original view and the synthesized view, generated by an anchor view and its corresponding depth data. A structure that uses a similar approach was even proposed by the MPEG group, referred to as the FTV Data Unit [154]. In [171], multiple views are warped to a central view and then encoded using a 3D-DCT scheme, in order to exploit the inter-view correlation using the depth data and view synthesis.

G.2 Independent multiview coding using the MMP algorithm

Multiview sequences based on the upcoming 3D format contain multiple texture and associated depth maps. Regarding independent texture coding, the flexible segmentation and sophisticated intra prediction scheme of MMP produces results that outperform state-of-the-art image coders, such as JPEG2000 or H.264/AVC, in a rate-distortion sense [26]. The high frequency preservation feature of MMP is a great advantage for depth map coding, as shown in Appendix F. MMP's flexible segmentation and advanced intra prediction provide state-of-the-art results for depth map compression. In [30], the effectiveness of reconstructing intermediate views with coded depth data and uncoded texture is shown. Reconstructed views using MMP's coded depth data were objectively and subjectively superior to those of the other tested encoders.

In this section we propose a full MMP-based 3D encoder for both texture and depth, where multiple views and associated depth maps are coded independently with MMP. It will be referred as MMP-standalone. Notice that only static images are used, not video sequences. The exploitation of temporal correlation will be a subject of further investigation. We evaluate the compression artifacts using the same target rates defined by the MPEG group for its anchor sequences [188] and compare our proposal with an MVC-based solution.

G.2.1 Texture coding with MMP

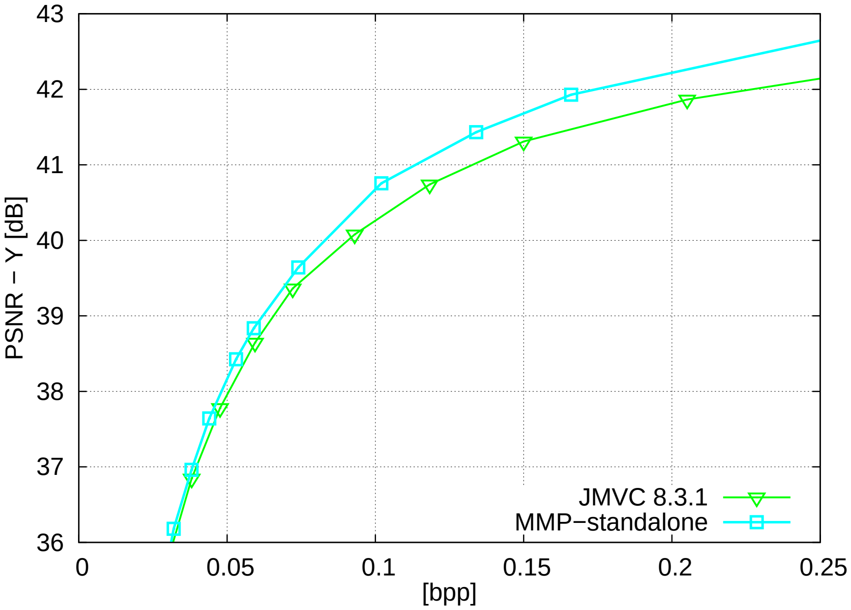
For multiview sequences, compression can be achieved by exploiting not only the temporal redundancy, but also the inter-view redundancy. The inter-view decorrelation is performed using a view as reference to the other views and coding only the residual information. Since the views tend to be similar to each other, a high compression ratio can be achieved by inter-view decorrelation. Due to MMP's state-of-the-art rate-distortion performance for image coding [26], it is expected that the use of MMP in a multiview coding scenario will provide higher compression gains, since inter-view decorrelation can benefit from the use of a higher quality reference view.

Table G.1 shows the experimental set-up, such as the selected cameras for the reference view and the target bitrates. Figures G.1 and G.2 show MMP's performance when coding one of the views, that can be regarded as reference view, of selected multiview sequences. For medium to high bitrates, MMP outperforms JMVC encoder, achieving gains up to 1 dB. Results for other smooth images can be found in [22, 26].

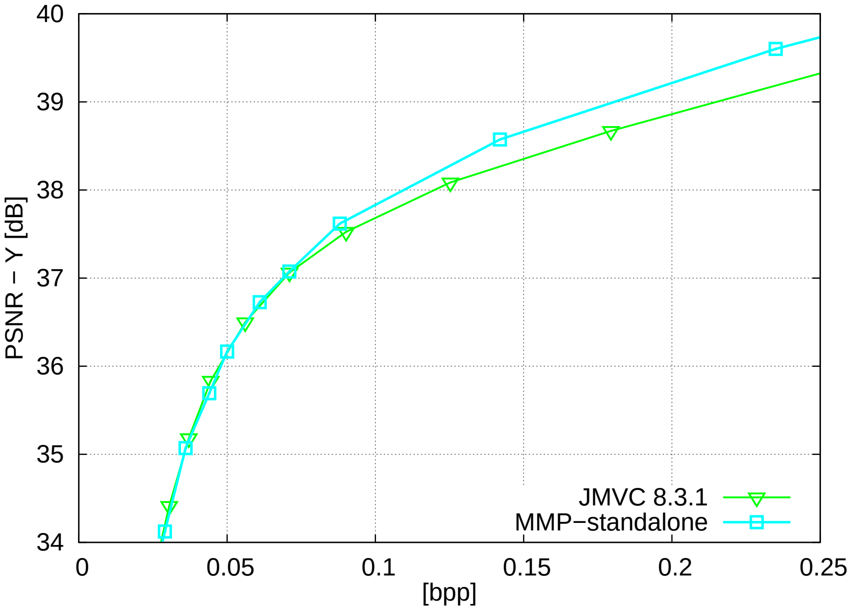
Results are also given for the coding of auxiliary view in Figures G.3 and G.4. Since the MMP-based algorithm does not exploit the inter-view correlation, its rate-distortion performance is usually lower than the one of JMVC. However, for sequences that were obtained with a non-parallel camera arrangement and possess geometric distortion from one view to another, such as the Ballet sequence, the disparity-based prediction used by the JMVC algorithm cannot appropriately model the correlation between views, resulting in a lower rate-distortion performance than MMP independent coding.

G.2.2 Optimal bit allocation for independent depth and texture coding with MMP

For its anchor sequences, MPEG needs to provide the best rate allocation between depth and texture coding for a pre-defined target rate. Under this context, experiments have been conducted in order to determine the optimal combination of quantization parameters (QP) to encode the stereo texture and depth [188].

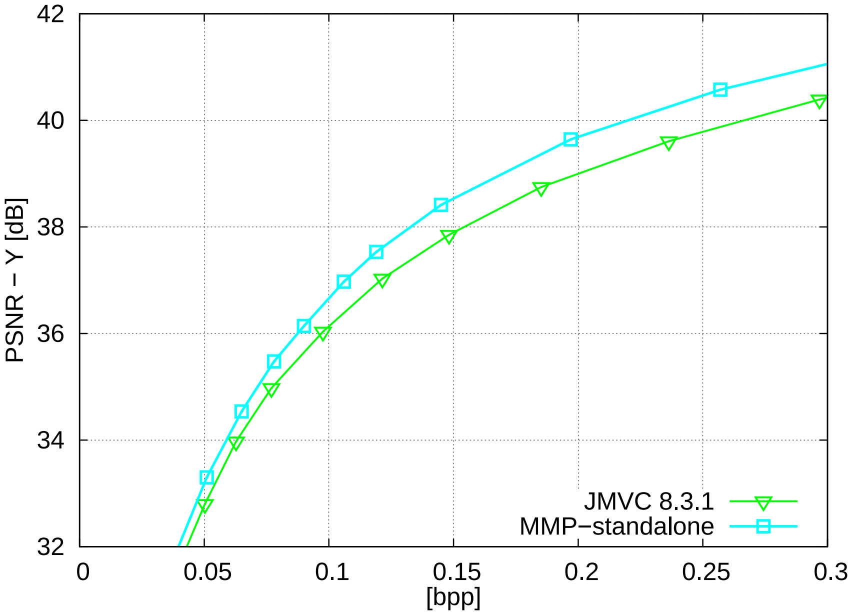


(a) Texture used as reference view from Ballet sequence, camera 3

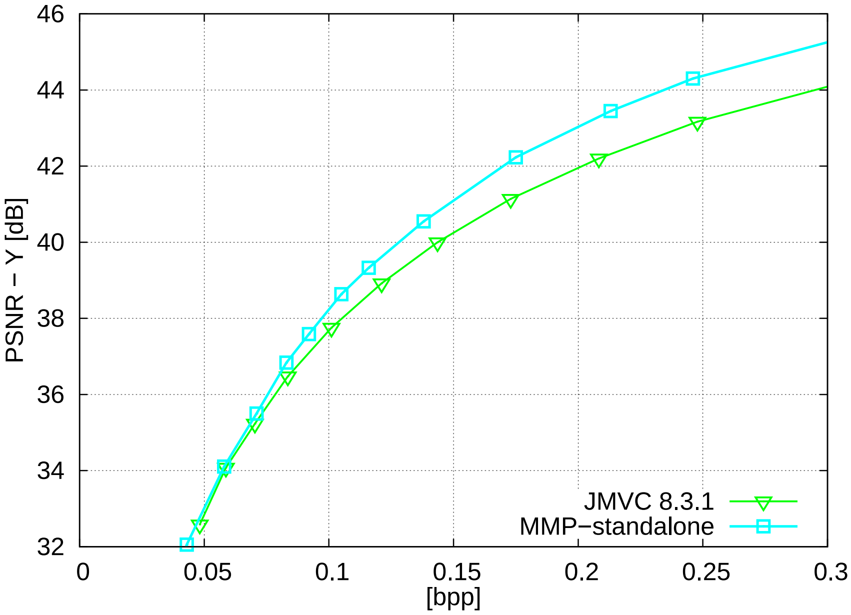


(b) Texture used as reference view from Breakdancers sequence, camera 0

Figure G.1: Rate-distortion curves for the first frame of the stereo pair sequences' reference view.

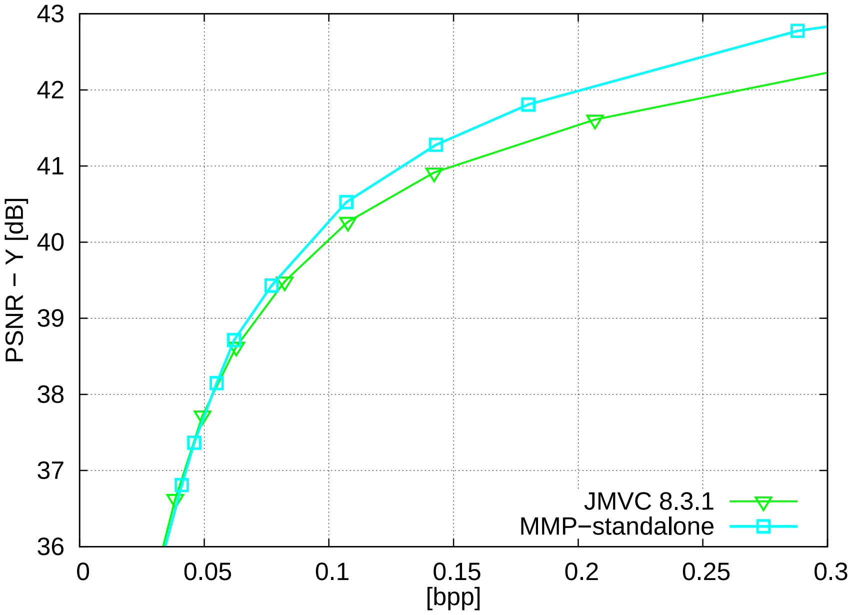


(a) Texture used as reference view from Book Arrival sequence, camera 10

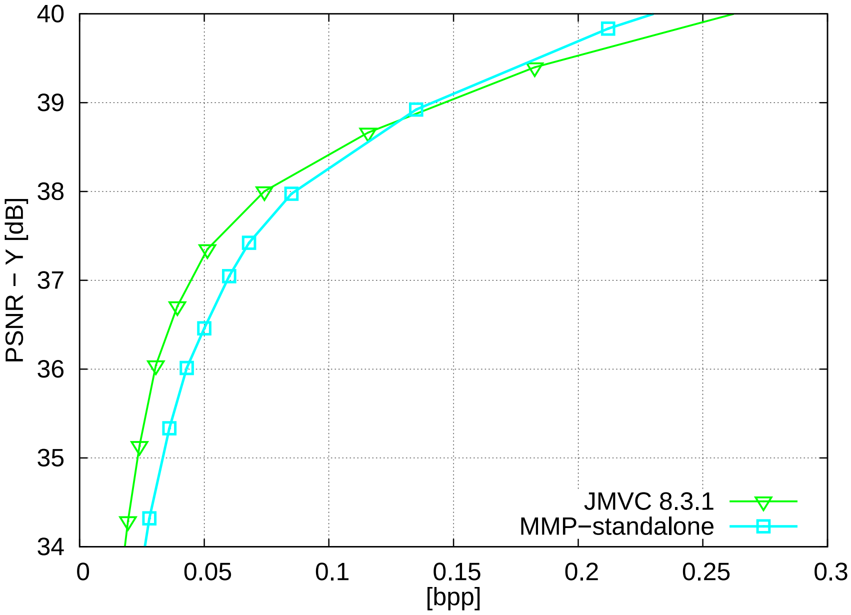


(b) Texture used as reference view from Champagne Tower sequence, camera 39

Figure G.2: Rate-distortion curves for the first frame of the stereo pair sequences' reference view.

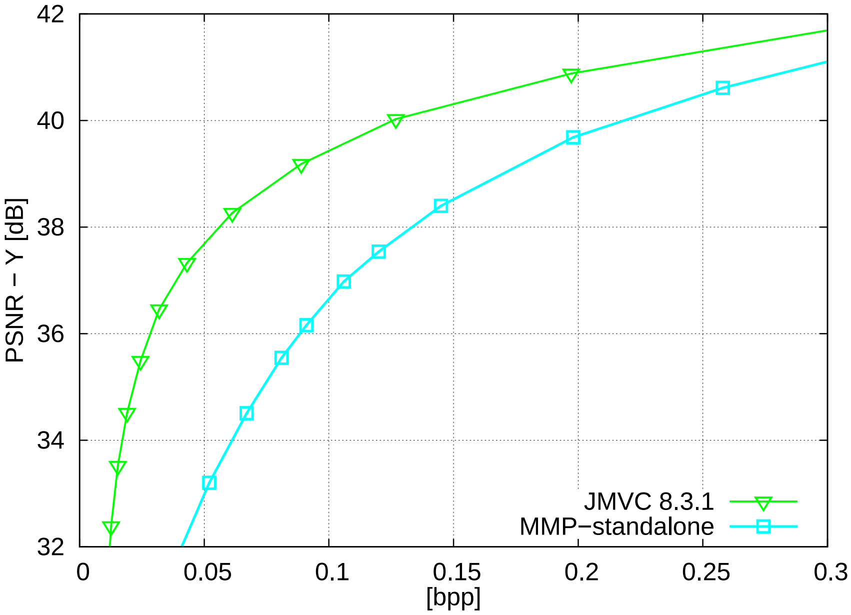


(a) Texture used as auxiliary view from Ballet sequence, first frame of camera 5

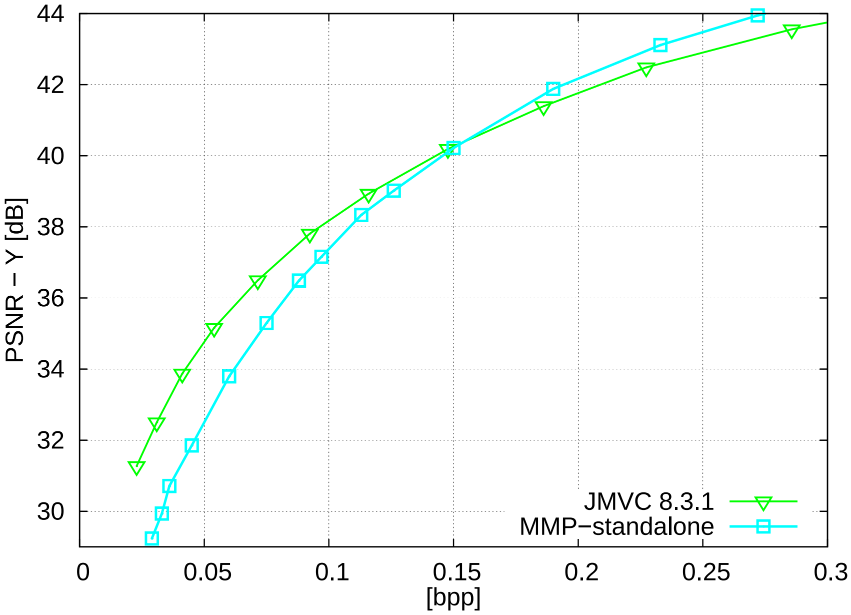


(b) Texture used as auxiliary view from Breakdancers sequence, first frame of camera 2

Figure G.3: Rate-distortion curves for the auxiliary view of the stereo pair sequences from Microsoft.



(a) Texture used as auxiliary view from Book Arrival sequence, first frame of camera 8



(b) Texture used as auxiliary view from Champagne Tower sequence, first frame of camera 41

Figure G.4: Rate-distortion curves for the auxiliary view of the stereo pair sequences

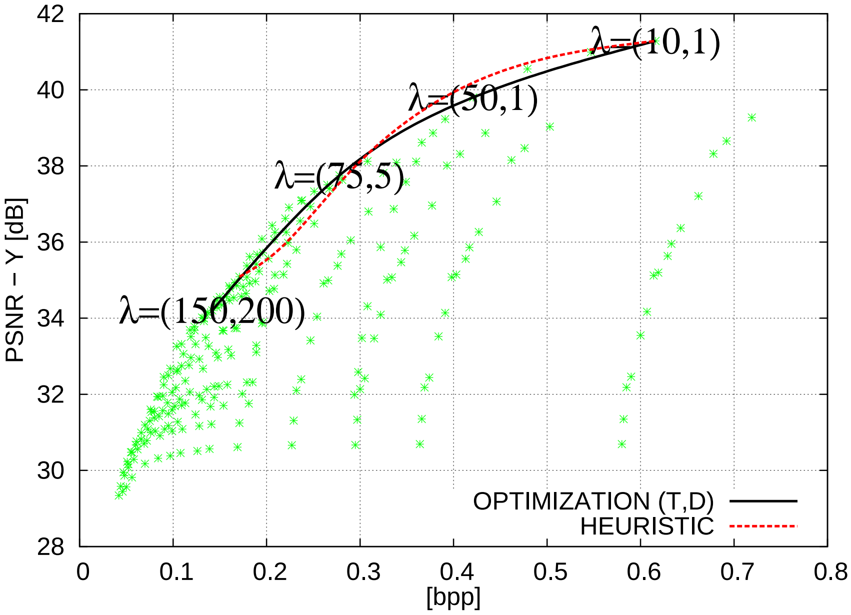
In a similar manner, an optimization of the rate allocation parameter λ (see Appendix B) was performed for the MMP-based encoder. Like the quantization parameter, this value determines the fidelity level achieved by the encoding process. A higher λ will allow more distortion for a lower bitrate, while a smaller λ will not tolerate distortion, at the cost of a higher bitrate usage. The goal is to find the outer convex hull of the RD points generated by coding texture and depth with MMP, at different compression levels. The chosen points, in our case, have similar target rates, as indicated by the MPEG group [188].

Figures G.5 and G.6 show the reconstructed view's PSNR, with the sum of the rates spent to code both texture and depth. The outer curve is marked, and the optimal lambda combinations for the target rates are also indicated in Figures G.5 and G.6. The leftmost value of each pair is the λ used for texture coding, whereas the rightmost is used for depth coding. Similar to [194], we also noticed that the final quality is very sensitive to the depth fidelity. Therefore, the rate allocation provides an indication that is more efficient to code depth data with higher quality than using this bitrate for texture coding.

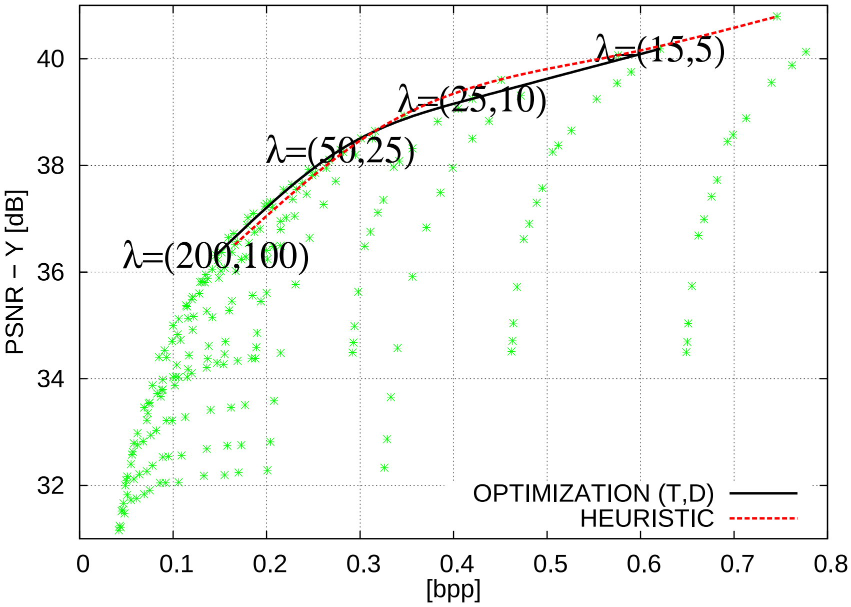
As a result of this extended analysis, a heuristic method for establishing the relationship among λs that give the optimal rate-distortion performance for the reconstructed virtual view was derived by DE OLIVEIRA [195]. This results in a relationship between the λ parameter that should be used when encoding texture (λ_{texture}) and depth (λ_{depth}), given by:

$$\lambda_{\text{depth}} = \begin{cases} 0.25, & \text{if} \quad \lambda_{\text{texture}} \leq 10; \\ 0.75, & \text{if} \quad 10 < \lambda_{\text{texture}} \leq 35; \\ 10, & \text{if} \quad 35 < \lambda_{\text{texture}} \leq 70; \\ 50, & \text{if} \quad 70 < \lambda_{\text{texture}} \leq 250; \\ 100, & \text{if} \quad 250 < \lambda_{\text{texture}} \leq 500; \\ 500, & \text{if} \quad 500 < \lambda_{\text{texture}} \leq 1000; \\ 1000, & \text{otherwise.} \end{cases} \tag{G.1}$$

As can be seen also in Figures G.5 and G.6, the heuristic determined is closely related to the optimal curve at the convex hull of the rate-distortion points. The only exception is the Book Arrival sequence, which has a higher level of details in the texture. Therefore, the bitrate requirements for the texture is higher, demanding more bits for coding.

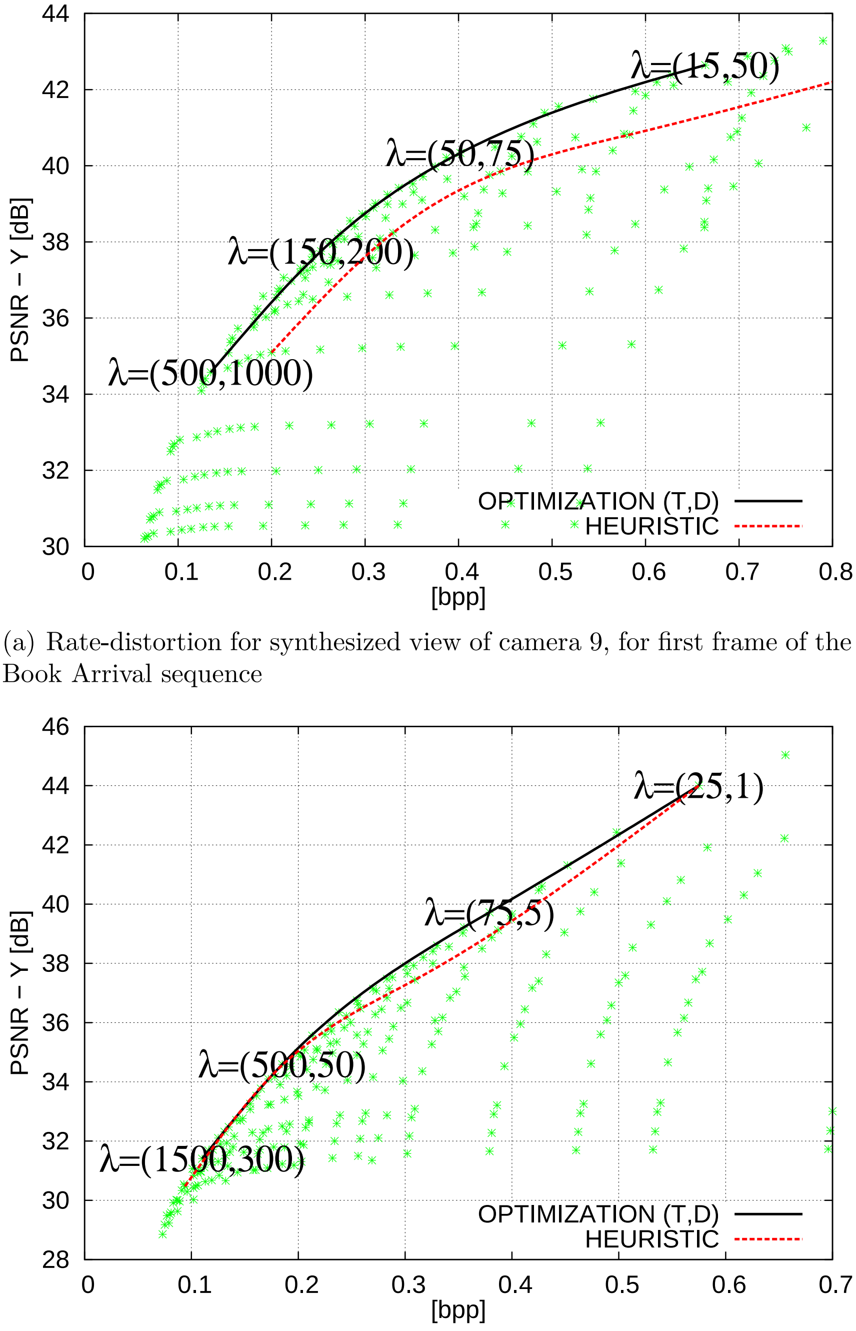


(a) Rate-distortion for synthesized view of camera 4, for first frame of the Ballet sequence



(b) Rate-distortion for synthesized view of camera 1, for first frame of the Breakdancers sequence

Figure G.5: Finding the optimal λ combination for coding depth and texture independently with MMP for the sequences provided by Microsoft.



(a) Rate-distortion for synthesized view of camera 9, for first frame of the Book Arrival sequence

(b) Rate-distortion for synthesized view of camera 40, for first frame of the Champagne Tower sequence

Figure G.6: Finding the optimal λ combination for coding depth and texture independently with MMP for the sequences provided by MPEG.

G.3 Encoding texture and depth jointly through warping

The separate coding of each view does not exploit the inter-dependencies between them. Thus, coding gains can be achieved if information of one view is used to code the other view. The use of texture and depth information of one reference view can generate an estimate of another view, and inter-view decorrelation can be performed, therefore obtaining compression gains for the auxiliary view.

DIBR (Depth Image Based Rendering) algorithms are able to synthesize a view at any desired position, under the limitations of the available texture and depth. Therefore, for multiview coding, it is possible to use a depth/disparity-based synthesized view as prediction for inter-view decorrelation. This inter-prediction is also known as View Synthesis Prediction (VSP). Nonetheless, until now only marginal gains have been reported [72]. In this section, we will use the View Synthesis Prediction concept to shape an MMP-based encoder.

In the following subsections, we propose an architecture for joint texture and depth map coding, named MMP-estimation. The proposed architecture adds the warped reference in the predictive coding loop to optimize the residue coding, an approach similar to [86]. For comparison we also show results for the MMP-standalone architecture, using the optimal λ combinations shown in Figures G.5 and G.6. This independent coding approach shows the potential of our encoder regarding the edge preservation feature.

G.3.1 Joint encoding architecture

In our proposal, we use a prediction frame obtained from warping the reference view camera position to the auxiliary view camera position, using the respective camera parameters, and the coded depth information of the reference camera. This prediction is called view synthesis prediction (VSP). For our experiments we have used the VSRS software [178] to generate the VSP. Since VSRS is able to perform image extrapolation, the generation of the VSP using one view and its respective depth map is straightforward. The depth data used for the warping operation was compressed by MMP, using the same optimal λ combinations obtained for the MMP-standalone case.

We added the warped frame in the prediction loop as an additional prediction mode. The warped prediction is chosen whenever it results in the lowest Lagrangian cost:

$$\text{mode} = \min(J_{\text{mode}} | \text{mode} \in \{\text{H.264/AVC}, \text{LSP}, \text{VSP}\}) \qquad \text{(G.2)}$$

where H.264/AVC are the intra prediction modes based on the homonymous standard, the LSP mode is the prediction mode introduced in [26], and VSP is the new view synthesis prediction mode.

Figures G.7 and G.8 show the rate-distortion curves of the auxiliary view coded using the proposed techniques, and also compare them with the MMP-standalone architecture and JMVC encoder. In camera arrangements that cause geometric distortions, the advantage of using prediction obtained from view synthesis is notable. MMP outperforms JMVC encoder for the Ballet and Breakdancers sequences. For horizontal camera arrangement, where disparity-based prediction is more suitable, MMP results are very similar to JMVC results, indicating that the prediction is effective in reducing inter-view correlation and improving the rate-distortion performance of the auxiliary view coding.

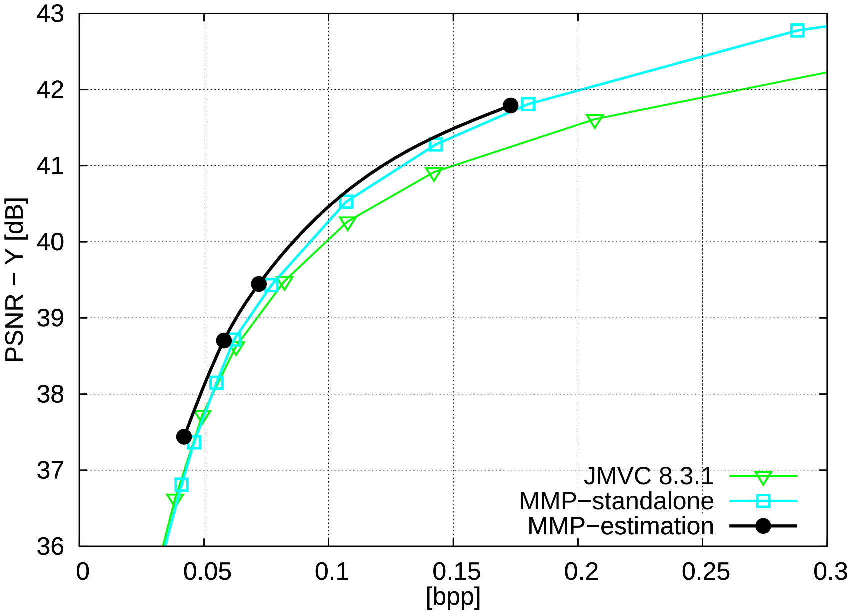
Since the VSP uses only one view for view generation, this may result in large occluded areas that must be inpainted by the VSRS software, generating many rendering artifacts. These artifacts compromise the performance of the prediction process using the VSP frame. However, at these occluded areas, the encoder may avoid the artifacts by choosing one of the intra-prediction modes.

G.3.2 Efficiency of the warped image as a predictor

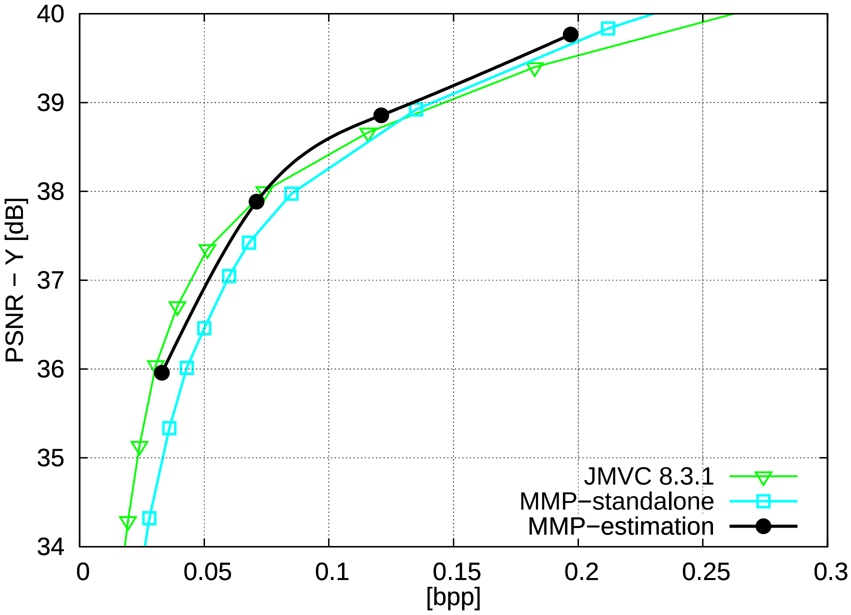
The warping function is an image operation that does not target the reduction of differences between two views. Indeed, it aims the good reconstruction of a target view using information from a reference view, which may not necessarily match what the target camera might be capturing in reality. This task can be very challenging, specially in the presence of problems, such as occluded areas and depth maps inaccuracies.

In some cases, it might be appropriate to create a new depth data that has the purpose of reducing the differences between the auxiliary view and the reference view, as was done in [86]. However, the rate penalty incurred in sending the depth data twice might not make this option viable.

Figures G.9 and G.10 display the PSNR of the images used as prediction considering three cases: only the prediction modes from the original MMP algorithm [26] are used (INTRA ONLY), only the VSP mode is used (VSP) or the combination of MMP's prediction modes and the VSP mode is used (INTRA+VSP). The prediction image constructed with only intra prediction modes uses information from the image itself to form the prediction. The VSP prediction uses inter-view information, that is, information from a different reference view to form the prediction. Finally, the combination of VSP and intra prediction modes may use information from the image itself and from the reference view to form the prediction. Notice that the use of VSP alone has a lower reconstruction quality result, when compared to the

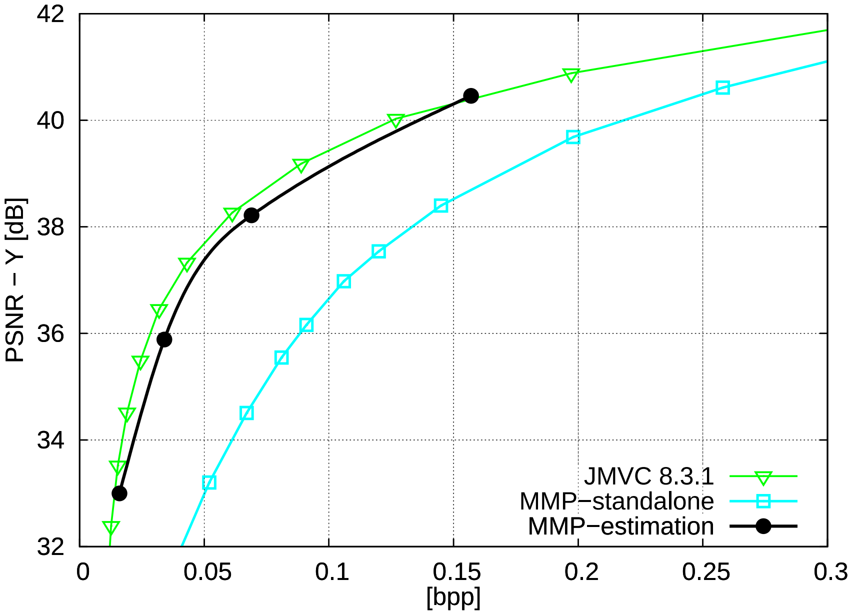


(a) Texture used as auxiliary view from Ballet sequence, first frame of camera 5

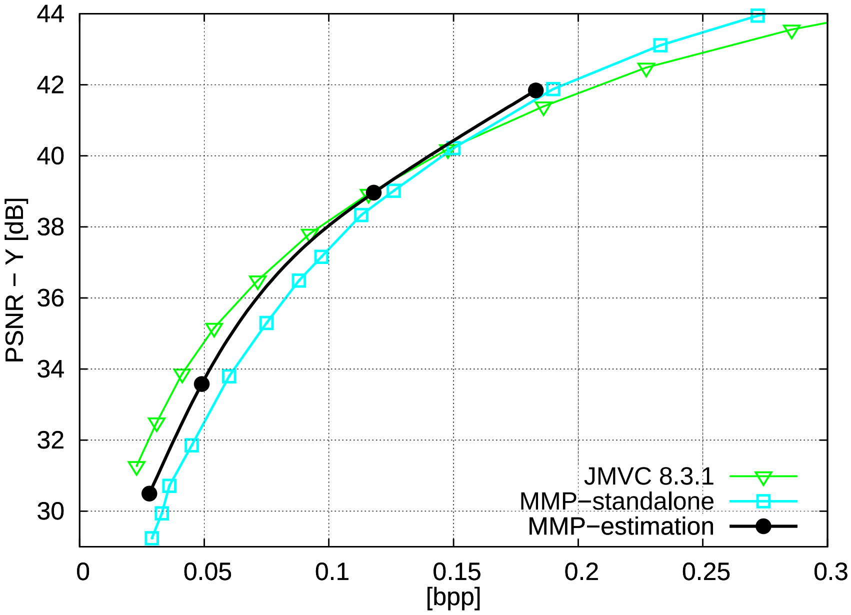


(b) Texture used as auxiliary view from Breakdancers sequence, first frame of camera 2

Figure G.7: Rate-distortion curves for the auxiliary view of the stereo pair sequences provided by Microsoft.

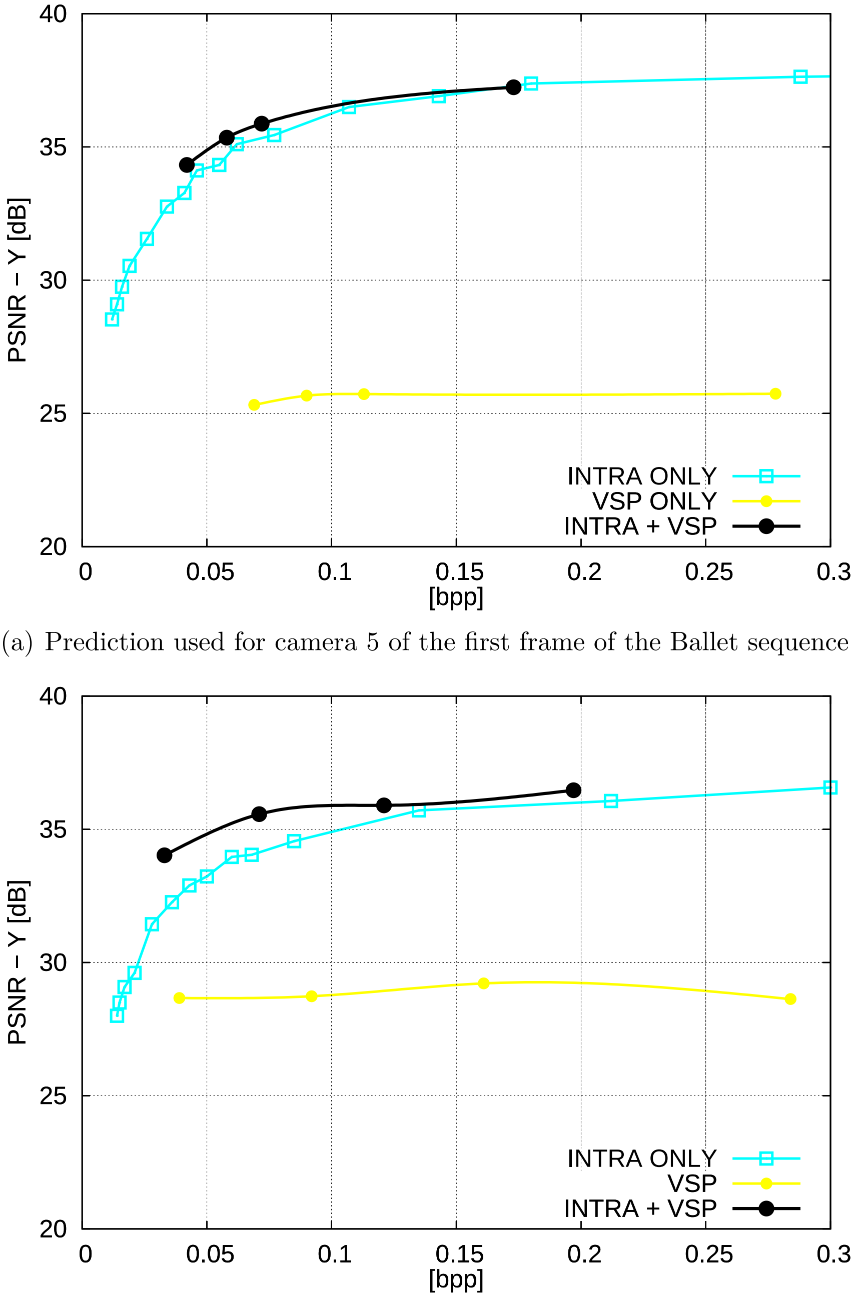


(a) Texture used as auxiliary view from Book Arrival sequence, first frame of camera 8



(b) Texture used as auxiliary view from Champagne Tower sequence, first frame of camera 41

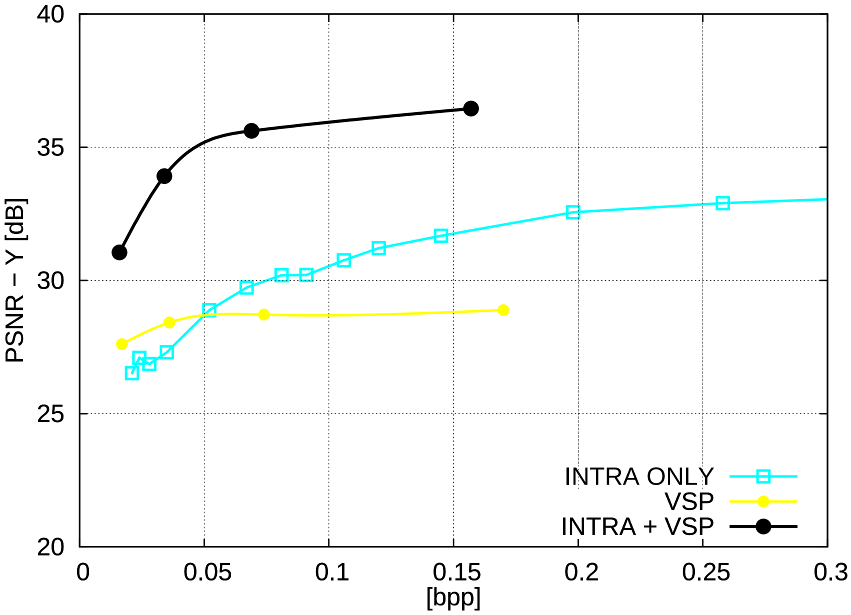
Figure G.8: Rate-distortion curves for the auxiliary view of the stereo pair sequences provided by the MPEG.



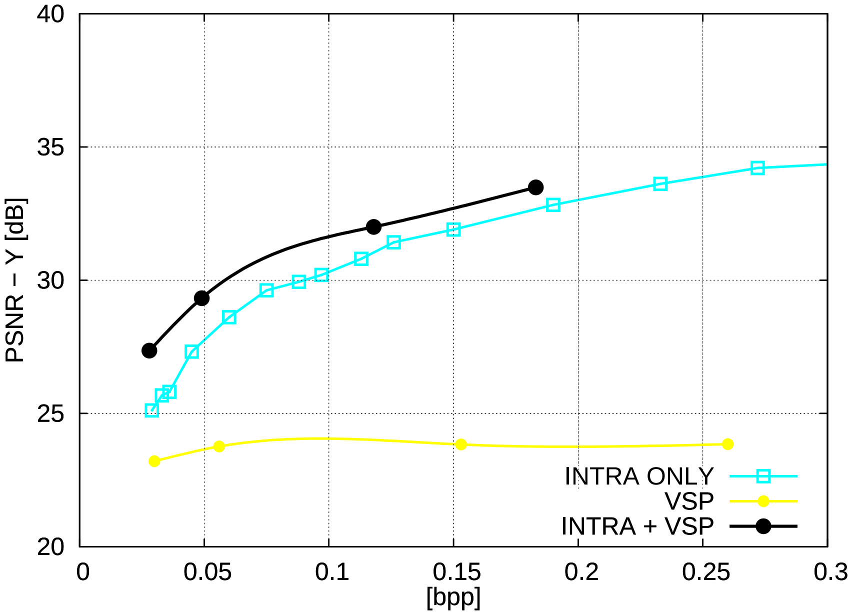
(a) Prediction used for camera 5 of the first frame of the Ballet sequence

(b) Prediction used for camera 2 of the first frame of the Breakdancers sequence

Figure G.9: Rate-distortion curves for the prediction for the auxiliary view of the stereo pair sequences from Microsoft.

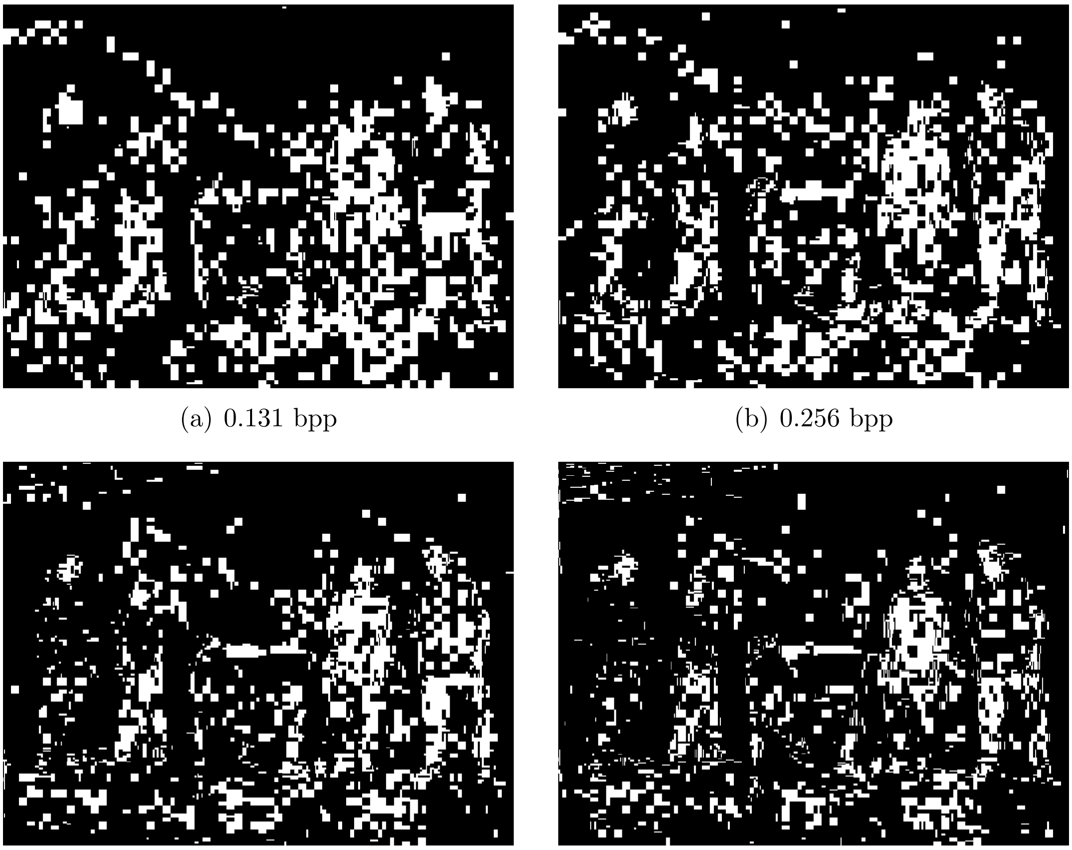


(a) Prediction used for camera 8 of the first frame of the Book Arrival sequence



(b) Prediction used for camera 41 of the first frame of the Champagne Tower sequence

Figure G.10: Rate-distortion curves for the prediction for the auxiliary view of the stereo pair sequences from MPEG.



(a) 0.131 bpp

(b) 0.256 bpp

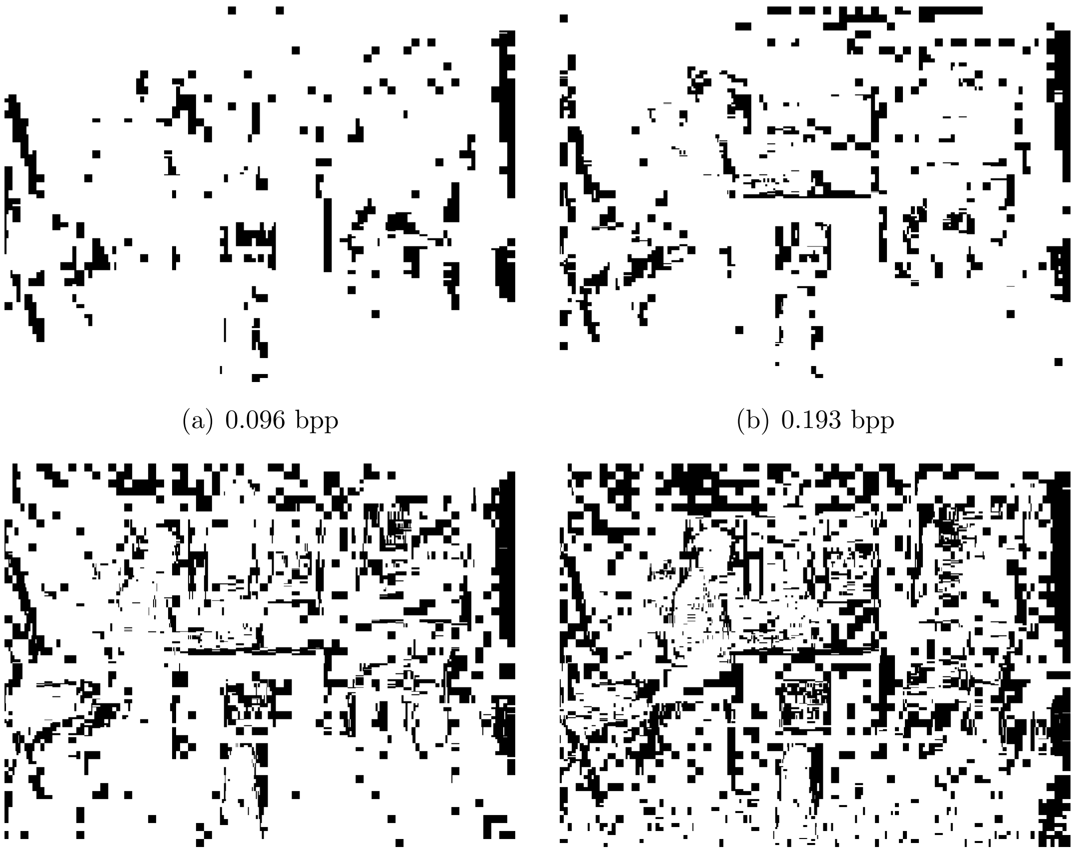
(c) 0.394 bpp

(d) 0.588 bpp

Figure G.11: Prediction efficiency for Breakdancers [G.11(a)-G.11(d)] sequence. The white areas represent the blocks that used the warped view as prediction.

other two prediction methods. The inpainting of large occluded areas contributes to the decrease in the PSNR value, even when using higher fidelity texture and depth data. In the case of combining both intra and inter-view prediction methods, the intra prediction mode can be used at the areas where the warping operation fails.

Figures G.11 and G.12 depict the usage maps of the VSP prediction mode for two different sequences, at several bitrates (lower to higher bitrates are positioned from left to right), indicated by the white area. Since horizontal camera arrangements usually have less geometrical distortion, the warped prediction presents less reconstruction artifacts and is better suited for inter-view decorrelation, justifying the higher adoption of the VSP mode for the Book Arrival sequence in comparison to the Breakdancers sequence. This mode is particularly useful at low bitrates. As no extra information is sent, we can obtain an acceptable prediction that is useful for coding the auxiliary view with no rate penalty. At higher bitrates, other prediction modes are used, specially in the areas where the warped view does not match the real view, for example at object boundaries or occluded areas. One suggestion for improvement of the VSP prediction mode, and also the reconstructed synthesized view, is the use of depth maps with higher precision and better inpainting algorithms


(a) 0.096 bpp

(b) 0.193 bpp

(c) 0.325 bpp

(d) 0.561 bpp

Figure G.12: Prediction efficiency for Book Arrival [G.12(a)-G.12(d)] sequence. The white areas represent the blocks that used the warped view as prediction.

[185].

G.4 Experimental results

In this experiments we used two sequences with a 1-D half-arch camera arrangement: Ballet and Breakdancers sequences from Microsoft [160], and two sequences with horizontal camera arrangement, from MPEG's set of reference anchor sequences: Book Arrival and Champagne Tower sequences [158, 159]. Both texture and related depth maps were encoded with JMVC 8.3.1 software, using the coding parameters and QP combinations obtained from MPEG's Experimental Experiences [191, 192] and also described in Table G.1. The rates reported account for the sum of the bitrates spent for coding texture and depth of both views. The PSNR values were obtained comparing the reconstructed view using coded data with the same reconstructed view using original non-coded data. The camera position where views are to be synthesized was also obtained from MPEG's experiments [188].

The optimal bitrate allocation, that determined the relationship between λ parameters for depth and texture, was done only for the independent coding of texture

Table G.1: Experimental set-up for multiview image coding, taken from the MPEG documents [188, 191, 192]. Since Ballet and Breakdancers Sequence have the same resolution as the Book Arrival sequence, the same rate-distortion points were used, and a similar optimization for the best QP combination was performed for those sequences. The table also shows the chosen cameras, left and right views, and the central view, which will be used for reconstruction

Sequences	Reference View(L)	Auxiliary View(R)	Virtual View(C)	Target Bitrate (Mbps)				QPT				QPD			
Ballet	3	4	5	0.3	0.5	0.75	1.25	38	34	30	24	36	26	22	22
Breakdancers	0	1	2	0.3	0.5	0.75	1.25	36	32	30	26	34	26	22	22
Book Arrival	10	9	8	0.3	0.5	0.75	1.25	36	34	28	26	42	34	34	26
Champagne Tower	39	40	41	0.3	0.5	0.8	1.5	42	38	34	28	44	36	24	22

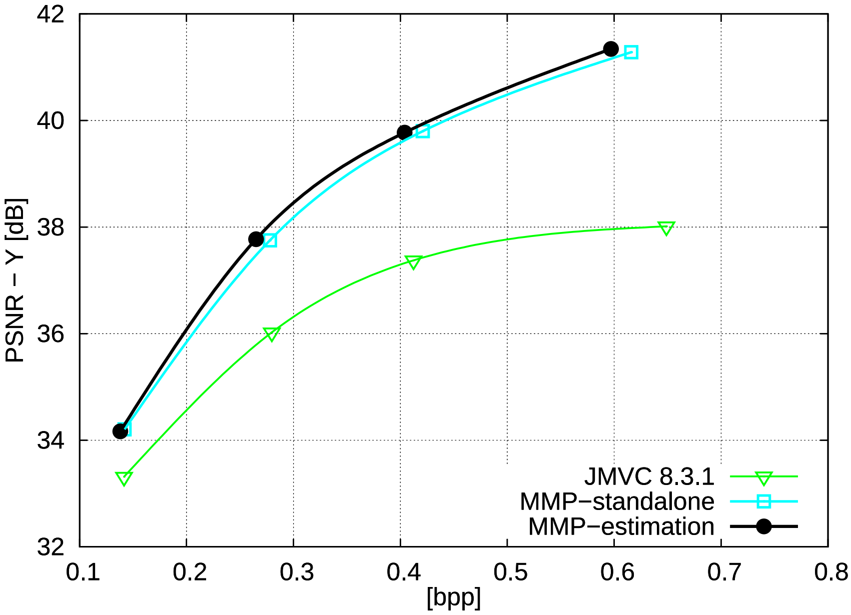
and depth. To reduce the scope of our work, the same optimal lambda combination used for MMP-standalone (relative to the independent coding described in Section G.2, see Figures G.5 and G.6) was also used for joint coding. Note that a better rate-distortion point can be found for the MMP-estimation (relative to the joint coding described in Section G.3), namely by performing a local search only at nearby points in the RD plane. On the account of the smooth monotonic properties of the rate-distortion surface, an extensive search would not be necessary [196].

G.4.1 Coding results

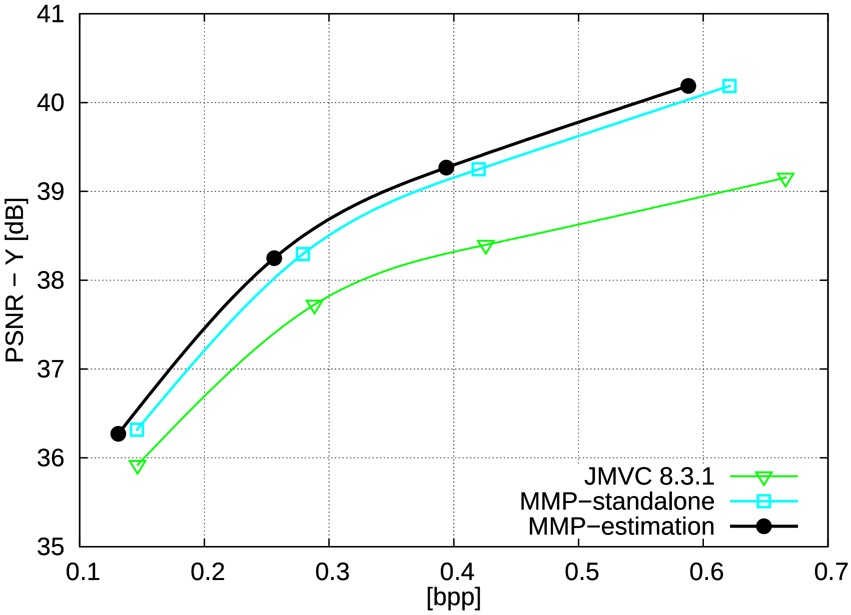
Figures G.13 and G.14 show the rate-distortion curves of the view synthesized from data coded by JMVC 8.3.1 software and the two MMP encoding architectures. MMP efficiently codes the depth data, as well as texture information. Hence, for all sequences, the high fidelity of the coded depth map, despite the low bitrate requirement, allows the generation of the highest quality virtual views. We can notice up to 4 dB gain for luminance in the Champagne Tower Sequence at high bitrates (see Figure G.14(b)). In addition, the PSNR gains vary from 1 dB to 3 dB for the Microsoft sequences at similar rates (see Figures G.13(a) and G.13(b)). Both tested versions of MMP (MMP-estimation and MMP-standalone algorithms) outperform JMVC encoder at all bitrates, for all sequences tested.

G.4.2 Subjective analysis

We show the reconstructed virtual views from Ballet (Figure G.15) and Book Arrival (Figure G.16) sequences to illustrate the subjective performance. The former has a non-aligned camera arrangement, causing several geometrical distortions in the stereo pair. This contributes to the decrease of efficiency of the H.264/MVC scheme, which performs only disparity-based inter-view decorrelation, and cannot

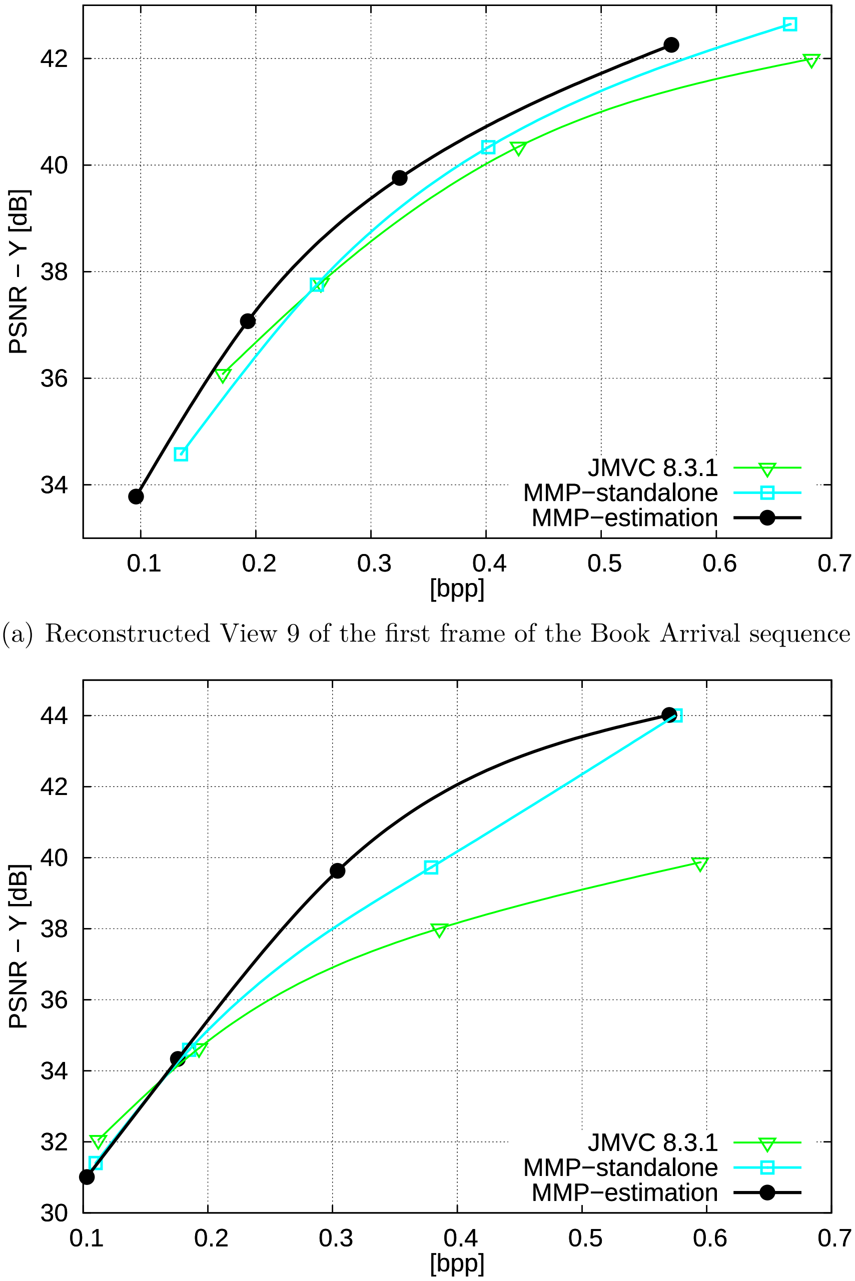


(a) Reconstructed View 4 of the first frame of the Ballet sequence



(b) Reconstructed View 1 of the first frame of the Breakdancers sequence

Figure G.13: Rate-distortion curves for the reconstructed virtual view of the stereo pair sequences provided by Microsoft.

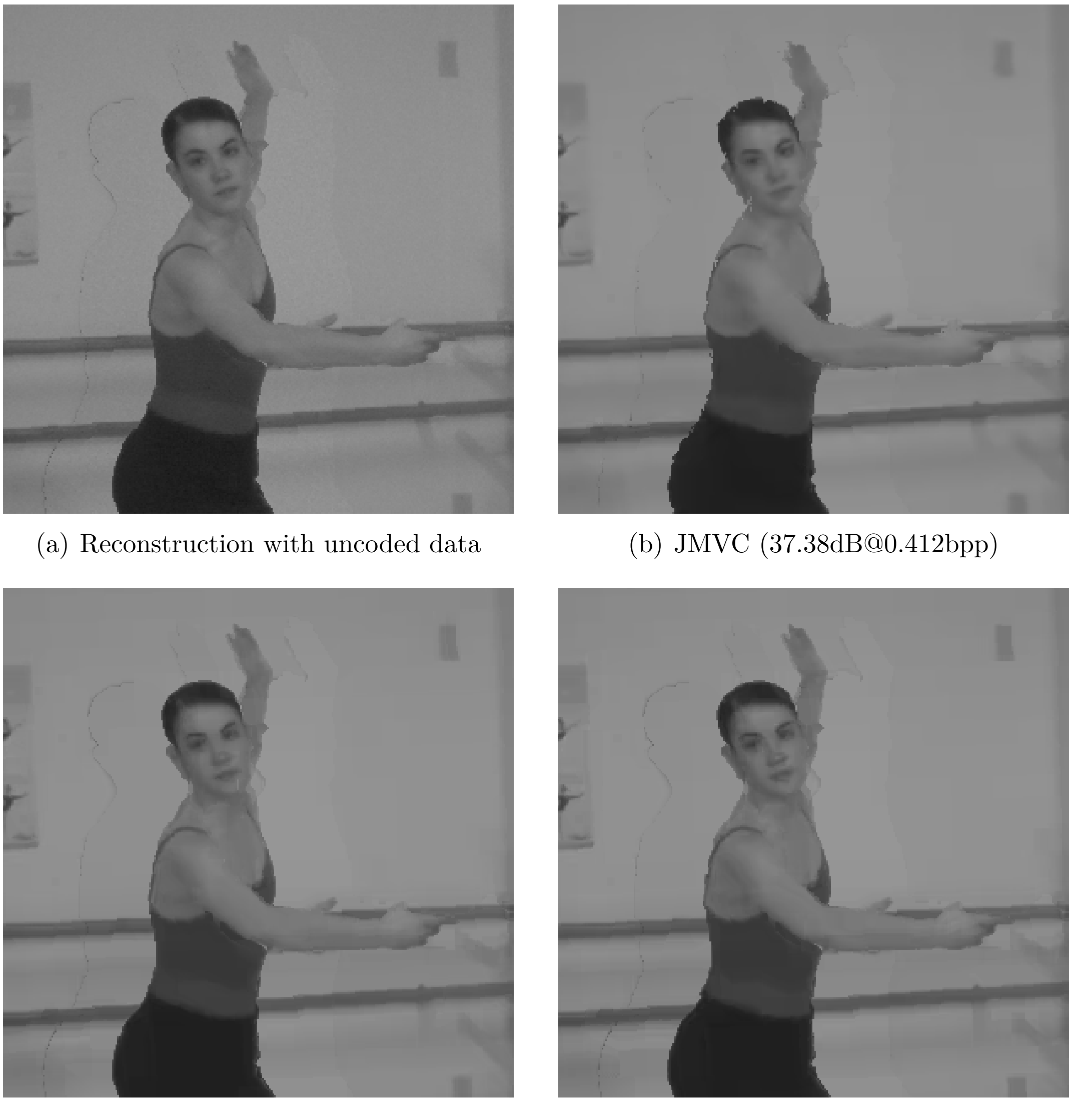


(a) Reconstructed View 9 of the first frame of the Book Arrival sequence

(b) Reconstructed View 40 of the first frame of the Champagne Tower sequence

Figure G.14: Rate-distortion curves for the reconstructed virtual view of the stereo pair sequences provided by MPEG.

cope with the non-translational motion between cameras. Hence, the advantage of using the warping function becomes clear, since the view synthesis approach can compensate non-translational disparities by using the projection model. For the latter sequence, a horizontal camera alignment is used, reducing geometry-related effects. Nevertheless, it is advantageous to jointly code texture and depth, since inter-view prediction can drastically reduce the required bitrate.



(a) Reconstruction with uncoded data

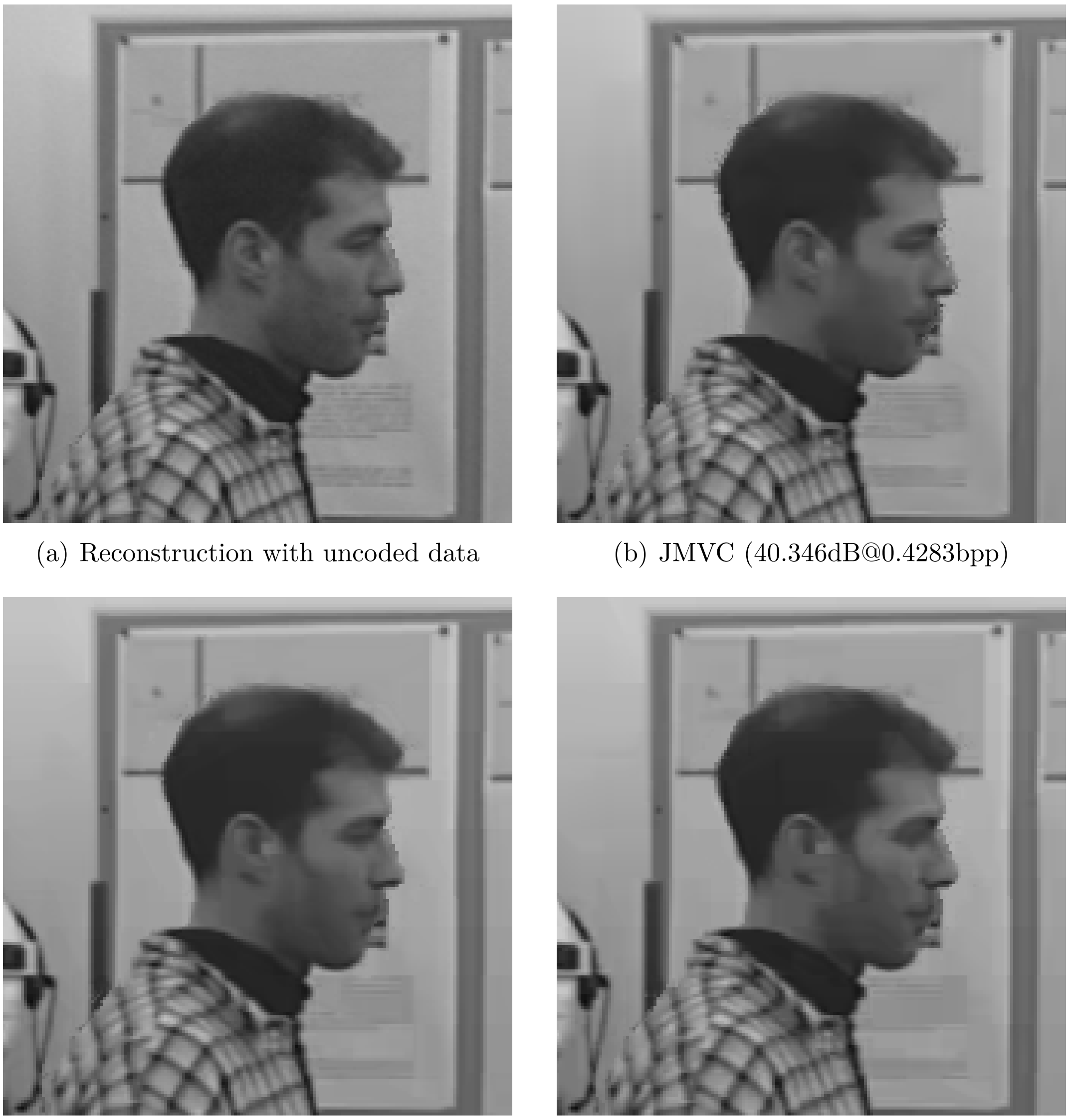
(b) JMVC (37.38dB@0.412bpp)

(c) MMP-standalone (39.80dB@0.421bpp)

(d) MMP-estimation (39.77dB@0.404bpp)

Figure G.15: Subjective comparison for the reconstructed virtual view of the first frame of the Ballet Sequence (Camera 4, frame 000).

In Figure G.15, it is possible to spot artifacts of depth and texture coding for the JMVC software and the proposed MMP encoders. The edges of the objects are specially affected by the H.264/MVC coding, noticeable on the mixture of foreground and background texture around the ballerina. As for the MMP-based encoders, we can see a more accurate preservation of the edges, which may be noted specially in finer detailed areas, such as the ballerina's hand. With the MMP-estimation encoder, the final quality of the reconstructed view is similar to the case when all the elements are coded independently, but its encoding is more rate-efficient, spending less bits to code both views and related depth maps.



(a) Reconstruction with uncoded data

(b) JMVC (40.346dB@0.4283bpp)

(c) MMP-standalone (40.335db@0.402bpp)

(d) MMP-estimation (39.758dB@0.325)

Figure G.16: Subjective comparison of the reconstructed virtual view for the first frame of Book Arrival sequence (Camera 9).

For the Book Arrival sequence, the usage of the warped frame as prediction for the MMP demands less bits to generate intermediate views with equivalent fidelity than the usage of intra modes alone in prediction, as depicted in Figures G.16(c) and G.16(d). Visual inspection of the details of the reconstructed view in Figure G.16 shows that the ringing artifacts of depth data encoded with the JMVC encoder generate sample scattering at the boundaries of the back of the man's head, as well as the front part of his nose. These reconstruction artifacts are not present at the uncoded reconstructed view, nor at the MMP-encoded reconstructed views. The preservation of depth edges or discontinuities presented at images reconstructed after MMP algorithms, assures that synthesized views shall not suffer from typical artifacts inflicted by transform-based encoders. However, blocking artifacts are visible in the reconstructed views using texture coded with the MMP. This effect can be minimized with the usage of deblocking filters, as proposed in [18].

G.5 Conclusions

In this appendix, the performance of a 3D encoder based on the MMP algorithm was analyzed. At first, results for coding both texture and depth independently were presented, and also an optimal rate-distortion allocation for both texture and depth was found. Then, in order to exploit the inter-view dependency, an architecture was proposed, which uses a warping function and predictive coding for inter-view decorrelation.

The proposed architecture uses MMP to encode the residues that are obtained from the difference between the original image and either the warped view or an intra-predicted image. This architecture is very tolerant to synthesis errors, such as artifacts resulting from inpainting occluded areas. The view synthesis prediction mode is used quite frequently, mainly at low bitrates, causing the reduction of the bitrate burden to code the auxiliary view.

At last, we compared the reconstructed view using the MMP-based encoders and the JMVC encoder, and we showed that MMP-based encoders outperform JMVC at all target rates, for all sequences. The MMP algorithm presents less artifacts in the synthesized views due to its edge preservation property, and views reconstructed with MMP-coded data show better objective and subjective quality.

This research still has some open topics that can be dealt with in the future. Methods for enhancing the View Synthesis Prediction, focusing on the auxiliary view coding, can contribute to the improvement of the encoder. The use of alternative view synthesis methods instead of the VSRS, such as the ones presented in [153, 177], may lead to improved predicted views and more efficient inter-view decorrelation. The inpainting algorithm has also shown to be able to influence the result of the residual encoder. Thus, schemes that implement better hole-filling algorithms, specially for large occluded areas, such as the ones proposed in [186, 197], may also improve this particular encoder architecture. Finally, it is foreseen also an MMP-based encoder for video sequences, that includes all the evolutions mentioned so far and also explores temporal correlations.

Apêndice H

Conclusions and perspectives

SUMMARY: This thesis has presented several proposals for image coding using the Multiscale Multidimensional Parser algorithm. The techniques presented here successfully accomplished their tasks in either decreasing the algorithms complexity, enhancing its rate-distortion performance or even developing new coding techniques for other areas, such as lossless image coding and 3D multiview coding. In this appendix, the achievements are summarized and possible extensions of the proposed algorithms are mentioned. Finally, some perspectives of image coding using the MMP will be discussed.

H.1 Conclusions and discussion for each individual contribution

H.1.1 Multiscale multidimensional parser

Latest developments in the MMP algorithm have provided it with a better rate-distortion performance, at the cost of increased computational complexity. As was shown in Appendix B, the addition of flexible segmentation in conjunction with an hierarchical prediction has dramatically increased the encoding time.

The computational cost of the algorithm was addressed in this thesis. A proposal for fast decision making of the choice of best prediction mode considering only the distortion of the residual block, instead of its coding cost was implemented. The modification allowed a decrease in computational complexity of about 80%. However, the rate-distortion performance also suffered losses of around 0.25 dB for smooth images, or even larger losses for compound images.

MMP computational complexity is a critical point and must be addressed. The proposed technique was one effort in this direction, but many other methods can be implemented or adapted for the MMP encoding scheme. For example, dictionary clean-up approaches and fast search methods for early block decision can also be incorporated into the MMP framework. Nevertheless, the trade-off between computational gains and rate-distortion performance losses needs to be evaluated. One promising area for encoding time reduction is the use of parallel processing and GPUs (*Graphic Processing Unit*). However, still a great effort in research needs to be done, in order to determine which routines in MMP are critical and may be parallelized.

H.1.2 Least-squares prediction in MMP

One feature that provided enhancement in MMP's rate-distortion performance was the modification of the algorithm, in order to make it more adaptive, like for example the flexible segmentation scheme. One element of the MMP-FP algorithm that was still rigid was the prediction mode. The 9 prediction modes use a fixed neighborhood weighting function, and cannot adapt its prediction structure to variations inside the block.

The Least Squares Prediction mode was first proposed for lossless coding, and assumed a raster scan encoding order of the image. The neighborhood used for prediction and training used the closest pixels from a non-symmetric half plane, and presented excellent edge adaptation properties, appropriate for image modeling.

In Appendix C we proposed the use of an adaptive least-squares prediction mode in a multiscale recurrent pattern image encoding framework. At first, a non-trivial adaptation of the prediction mode for block coding was done. The prediction calculation was modified in order to use pixels positions closer to the position to be predicted, instead of always using pixels from neighboring blocks.

The technique proved to be very efficient, especially for images with spatial high frequency content. Gains reported for such images were more than 1 dB, in the range of middle to high rates. For images with less high frequency content, the new prediction mode still reported some gains, and the LSP mode was one of the most used modes for smooth image coding. MMP with LSP prediction also presented no rate-distortion performance losses for text and compound image. The proposed method outperforms state-of-the-art, transform-based compression algorithms for all image types, from smooth to text and compound images.

Despite the increased rate-distortion performance, adding one more prediction mode also increased the computational complexity, since now one more mode needs to be tested, involving several dictionary searches. LSP adds another concerning factor to this problem. Unlike the other fixed prediction modes, the adaptation of the predictors coefficients has a non-negligible complexity, since it involves an inverse matrix operation. A future topic for investigation is the use of faster techniques for coefficient adaptation.

H.1.3 Lossless image compression using MMP

For the first time, MMP algorithm was evaluated for lossless coding. In Appendix D, a theoretical bound on its redundancy was derived, proving that MMP lossless coding rate achieves the entropy of any stationary, ergodic, memoryless source with finite alphabet. This also indicates how important the prediction stage is, in accelerating MMP dictionary adaptation and improving MMP's compression performance.

Several lossless coding techniques were adapted and validated in the MMP framework. As indicated by the theoretical results, the improvement of the prediction stage is beneficial for the algorithms compression capability, and a modification was adopted in order to use original values closer to the position to be predicted, instead of far away values from block neighborhood. Techniques to improve residue coding were also tested, but not adopted, since MMP's residue encoding, using arithmetic encoder and an adaptive dictionary, is already efficient enough.

The proposed enhancements were incorporated into the same algorithm used for lossy coding. Now MMP is able to perform lossless and lossy coding, achieving state-of-the-art results at all rates, for several types of images.

H.1.4 Depth coding using MMP

The new 3D image format is an interesting challenge for image compression. The depth data of a scene can be viewed as a luminance only signal, and also encoded as such. However, the specific characteristics of this type of image make it especially difficult for encoding. The image presents a mixture of very low frequency signals with high frequency edge information.

Due to its edge preservation feature, result of all the improvements added to the MMP algorithm (flexible segmentation and edge adaptive prediction mode), MMP is an adequate tool for depth map coding. Results provided in Appendix F show that MMP is very efficient for coding depth maps at middle to high rates and also allows an acceptable reconstruction.

A proposition to add an edge-aware feature in MMP's encoding loop was made. Although not rate-distortion effective, the edge-aware technique added showed an interesting property of the original MMP algorithm: the high quality edge reproduction at middle to high rates. An interesting conclusion on this topic is the importance of edges for depth maps. Coding artifacts at edges compromise view reconstruction. By preserving edge information, MMP is suited for coding depth information that will be used for view synthesis.

However, for low bitrates MMP presents some reconstruction problems due to blockiness. Since the algorithm does not have enough rate for dictionary adaptation, the encoded depth images are too heavily quantized, and the view synthesis stage is particularly affected by the coding artifacts. However, no viable solution is known at such low bitrates.

MMP also has the advantage of its flexible encoding scheme, being ready to code both texture and depth. A natural extension of the work presented here is the development of a full encoding system for 3D images, where MMP is used to code the depth maps and the texture views. Due to its universal character, MMP is ready to be used for base view and depth map coding, and at all bit rates, from lossy to lossless.

H.1.5 Joint texture and depth coding using MMP

In Appendix G, a proposal for a full MMP-based 3D image encoder was made. Coding both texture and depth with MMP demanded an efficient bitrate allocation between the two, and this was also done under the framework of this topic. Our results have shown the importance of depth coding, and the benefits of using MMP for coding both elements of the new 3D format. Efficient texture coding and state-of-the-art results for depth coding were obtained, and view reconstruction using MMP-coded data was also superior than other encoders.

The first results presented coded texture views independently. However, inter-view correlation can be exploited by the coding stage of multiple views. In Appendix G we also presented an architecture that jointly exploits the inter-view dependency using the warping function and predictive coding. The proposed architecture coded the residues that were obtained from the difference between the original image and either the warped view or an intra-predicted image. Moreover, we also showed that the bitrate spent to code multiple views was reduced and that the reconstructed view using the MMP-based encoder presented fewer artifacts than the ones obtained with coded data using the JMVC encoder. We showed that rate-distortion performance of the reconstructed virtual views using the MMP-coded data outperforms rate-distortion performance of views using data coded with JMVC at all target rates. The MMP algorithm presents less artifacts in the synthesized views due to its edge preservation property, and views reconstructed with MMP-coded data have better objective and subjective quality.

Future developments of this topic include the use of improved warped views, to increase view decorrelation, and the extension of the proposed technique for video.

H.2 Key issues and open questions

Though MMP has already demonstrated to be a very effective tool for image coding and has achieved state-of-the-art performance in several coding scenarios, a number of open questions still remains.

Do we really need a new image coding procedure, or should we stick to transform-based schemes? The point of using a different technique for a known-problem is the ability to think “out of the box”, which allows a disruptive way of dealing with a problem. This has a great potential of achieving new solutions or even improve the current ones. The knowledge acquired with MMP can be useful for transform-based coding techniques as well. MMP can help us understand how images are formed and how we can represent them more efficiently. Therefore, investigation on methods such as the ones proposed by the MMP should continue.

Nevertheless, a trade-off between theory and praxis should be reached for the MMP algorithm. At the actual stage, MMP is still far away from being a practical solution for image coding, due to its large complexity. However, the developments of machines with more computational power is a certainty. Computational complexity burden seems to be less and less important as time goes by. Also, the development of hardware specific for image processing, such as the GPUs, might pave the way for making MMP a practical solution too.

This leads us also to another open issue, which is to analyze the impact of the proposed encoding algorithm using different hardware. How MMP can be modified to leverage on the potential of hardware specific solutions is a very interesting open question. Due to MMP’s flexibility, it can be used with different types of images. The first results published showed a state-of-the-art performance for compound images.

Due to its high frequency preservation feature, MMP is able to encode such images with more efficiency than transform-based encoders. We have also seen that the new 3D format benefits tremendously with the MMP approach. With the ever increasing use of computer graphics mixed with real-life information, with increased resolution and detailed information, new image formats could arise. MMP has the potential to be a viable solution for encoding new types of images, or even helping to understand the critical issues when encoding them.

H.3 Future perspectives for image coding

It is a certainty that in the future we will experience images with greater detail, videos with higher resolution, 3D images, all due to the evolution of new displays and the wide spread digital education that we experience nowadays. The easy-access to digital information through the use of laptops, handheld devices and many other technological new gadgets that are coming into the market year by year will enable access to image and video content by the consumer. This also puts an increased pressure on image encoding algorithms, that need to cope with the higher standards for quality acceptance that are being developed by the users.

One typical solution for increasing quality is to decrease the compression, which may not be a viable solution for some cases. The other solution is to develop algorithms that are more efficient, that achieve higher compression gains at better final quality levels. Such proposed algorithms also incur in increased computational complexity. However, gadgets with increased computational capability will be ever more available, with accessible prices and tremendous computational power. Parallel processing is the hardware evolutionary tendency. Multiple cores, specific graphic processing units with shared memory and multiprocessors will enable solutions that could not even be realized some years ago.

Finally, it is the author's opinion that the MMP algorithm can be of great use for new image formats. The presence of digital manipulation of images and video through computer vision creates new challenges and also new possibilities for image coding. The increased mixture of digital content and real content, images and computer graphics, is propitious for an algorithm that does not assume any typical behavior of the source's statistic.

Apêndice I

Pseudo-codes

SUMMARY: This appendix outlines pseudo-codes for the MMP algorithm used in this thesis. The following subsections analyze each routine of the algorithm with greater details.

I.1 Main function of the MMP encoder

MMP's main function can be divided into three local routines:

- Block optimization (block 1). In this routine the optimal coding parameters will be chosen. After a complete analysis of all the coding combinations, the segmentation flags, prediction modes and dictionary indexes that result in the lowest cost will be saved in a tree-like structure, to be passed on to the next routine. More details are given in Section I.2
- Block coding (block 2). The optimal parameter combination passed on by the previous routine will be entropy encoded here. The generated bitstream, that will be later interpreted by the decoder, is the output of this routine. Further details are discussed in Section I.3.
- Dictionary update (block 3). The coded segments are concatenated and this new pattern is added to the dictionary in this routine. More details can be found in Section I.4.

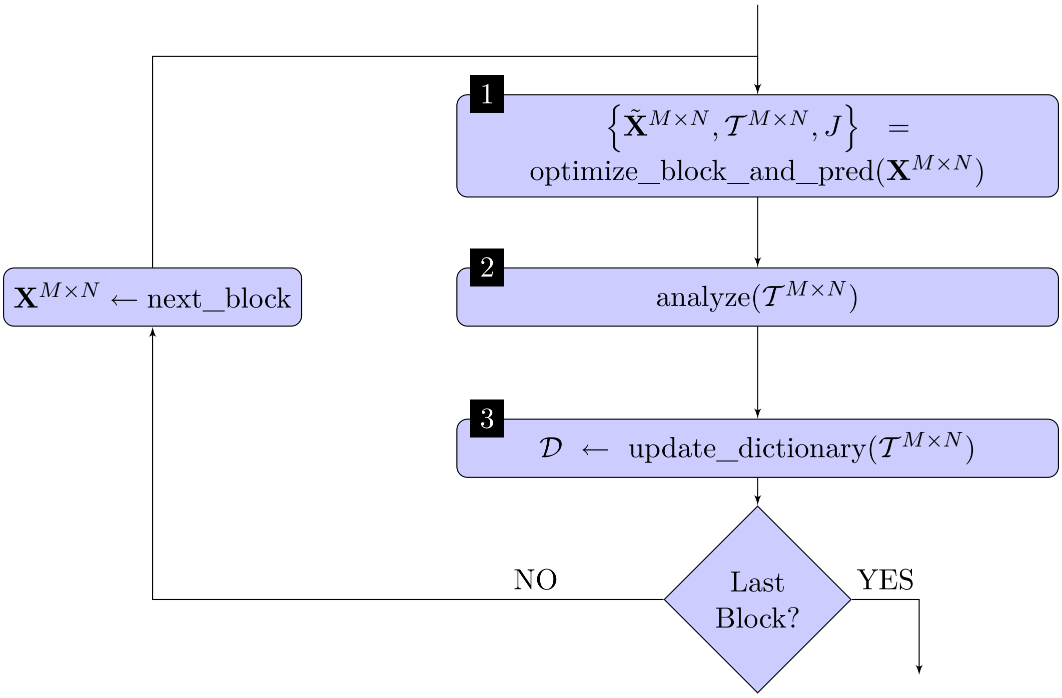

1
$\left\{\tilde{\mathbf{X}}^{M\times N}, \mathcal{T}^{M\times N}, J\right\}$ =
optimize_block_and_pred($\mathbf{X}^{M\times N}$)
2
analyze($\mathcal{T}^{M\times N}$)
3
$\mathcal{D}$ ← update_dictionary($\mathcal{T}^{M\times N}$)
$\mathbf{X}^{M\times N}$ ← next_block
Last Block?
NO
YES

Figure I.1: Main function diagram of the MMP encoder.

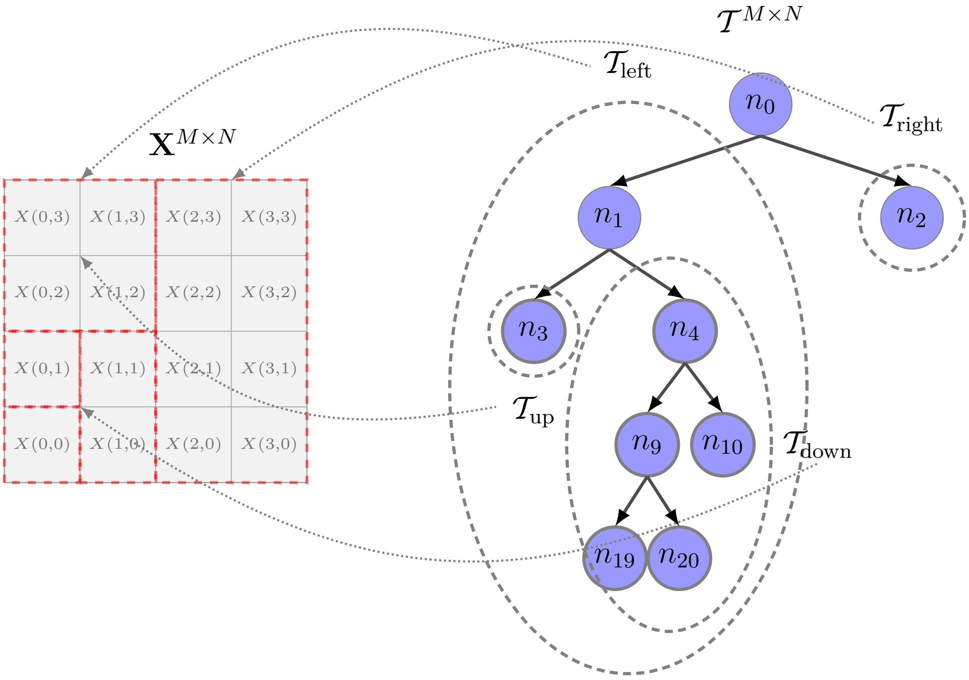


Figure I.2: Example for clarifying the notation used in the pseudo-code.

I.2 Block optimization

$$\left\{\tilde{\mathbf{X}}^{M\times N}, \mathcal{T}^{M\times N}, J\right\} = \text{optimize\_block\_and\_pred}(\mathbf{X}^{M\times N})$$

Step 1 Estimate the $\mathcal{M}$ possible block predictions for block $\mathbf{X}^{M\times N}$. Notice that due to lack of neighborhood, some prediction modes might not be allowed.

Step 2 For each prediction $m \in \mathcal{M}$, obtain the residual block, that is, the difference between the original block and the corresponding prediction, $\mathbf{R}_m^{M\times N} = \mathbf{X}^{M\times N} - \hat{\mathbf{X}}_m^{M\times N}$.

Step 3 Call the optimize_block($\mathbf{R}_m^{M\times N}$) function to code the residual block. The output will be the reconstructed residue $\tilde{\mathbf{R}}_m^{M\times N}$, the analogous segmentation tree $\mathcal{T}^{M\times N}$ and the corresponding Lagrangian residue coding cost J_{res}.

Step 4 The total coding cost $J_{\text{pred}}(m, \text{noseg}|\mathbf{X}^{M\times N})$ will be given by the sum of the Lagrangian residue coding cost, $J_{\text{res}}(\mathcal{T}^{M\times N}|\mathbf{R}_m^{M\times N})$, with λ times the rate spent to code the prediction mode.

Step 5 If the recently calculated Lagrangian cost is lower than the best Lagrangian cost so far, then this will assume the new value, $J_{\text{best}} = J_{\text{pred}}(m, \text{noseg}|\mathbf{X}^{M\times N})$ and the reconstructed block will be given by the sum of the coded residue and the predicted block $\tilde{\mathbf{X}}^{M\times N} = \tilde{\mathbf{R}}_m^{M\times N} + \hat{\mathbf{X}}_m^{M\times N}$

Step 6 If there is still one prediction mode to evaluate, go back to Step 2.

Step 7 If the block can be divided vertically (that is, either the block vertical dimension is divisible by two, or the next dimension is larger than the minimum dimension allowed for prediction, $M \times N > 4 \times 4$), go to Step 8, otherwise go to Step 12.

Step 8 Call the optimize_block_and_pred($\mathbf{X}_{\text{left}}^{M\times\frac{N}{2}}$) function for the left subblock, and keep the Lagrangian cost $J_{\text{pred}}(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}})$, the reconstructed subblock $\tilde{\mathbf{X}}_{\text{left}}^{M\times\frac{N}{2}}$ and the segmentation tree $\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}$

Step 9 Call the optimize_block_and_pred($\mathbf{X}_{\text{right}}^{M\times\frac{N}{2}}$) function for the right subblock, and keep the Lagrangian cost $J_{\text{pred}}(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}})$, the reconstructed subblock $\tilde{\mathbf{X}}_{\text{right}}^{M\times\frac{N}{2}}$ and the segmentation tree $\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}$

Step 10 Calculate the Lagrangian cost for vertical prediction segmentation $J_{\text{pred}}(\text{vert}|X^{M\times N})$ as the sum of the Lagrangian of each half, $J_{\text{pred}}(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}) +$

$J_{\text{pred}}(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}})$, with λ times the rate spent to indicate that the block (and its prediction) was divided vertically (flag_V_pred).

Step 11 If the recently calculated Lagrangian cost is lower than the best Lagrangian cost so far, update the best Lagrangian cost with this value $J_{\text{best}} = J_{\text{pred}}(m, \text{vert}|X^{M\times N})$, and keep the reconstructed block and the segmentation tree as the vertical concatenation of both segments ($\tilde{\mathbf{X}}^{M\times N} = [\tilde{\mathbf{X}}_{\text{left}}^{M\times\frac{N}{2}} : \tilde{\mathbf{X}}^{M\times\frac{N}{2}\,\text{right}}]; \mathcal{T}^{M\times N} = [\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}} : \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}]$).

Step 12 If the block can be divided horizontally (that is, either the block horizontal dimension is divisible by two, or the next dimension is larger than the minimum dimension allowed for prediction, $M \times N > 4 \times 4$), go to Step 13, otherwise go to Step 17.

Step 13 Call the optimize_block_and_pred($\mathbf{X}_{\text{up}}^{\frac{M}{2}\times N}$) function for the upper subblock, and keep the Lagrangian cost $J_{\text{pred}}(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N})$, the reconstructed subblock $\tilde{\mathbf{X}}_{\text{up}}^{\frac{M}{2}\times N}$ and the segmentation tree $\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}$

Step 14 Call the optimize_block_and_pred($\mathbf{X}_{\text{down}}^{\frac{M}{2}\times N}$) function for the lower subblock, and keep the Lagrangian cost $J_{\text{pred}}(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N})$, the reconstructed subblock $\tilde{\mathbf{X}}_{\text{down}}^{\frac{M}{2}\times N}$ and the segmentation tree $\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}$

Step 15 Calculate the Lagrangian cost for horizontal prediction segmentation $J_{\text{pred}}(\text{hor}|\mathbf{X}^{M\times N})$ as the sum of the Lagrangian cost of each half, $J_{\text{pred}}(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}) + J_{\text{pred}}(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N})$, with λ times the rate spent to indicate that the block (and its prediction) was divided horizontally (flag_H_pred).

Step 16 If the recently calculated Lagrangian cost is the lowest achieved so far, update the best Lagrangian cost with this value $J_{\text{best}} = J_{\text{pred}}(m, \text{hor}|\mathbf{X}^{M\times N})$, and keep the reconstructed block and the segmentation tree as the horizontal concatenation of both segments ($\tilde{\mathbf{X}}^{M\times N} = [\tilde{\mathbf{X}}_{\text{up}}^{\frac{M}{2}\times N} : \tilde{\mathbf{X}}_{\text{down}}^{\frac{M}{2}\times N}]; \mathcal{T}^{M\times N} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$).

Step 17 Return the best cost $J = J_{\text{best}}$, the reconstructed block $\tilde{\mathbf{X}}^{M\times N}$ and the segmentation tree $\mathcal{T}^{M\times N}$. Notice that the reconstructed block may be used as neighborhood of the next block to be coded.

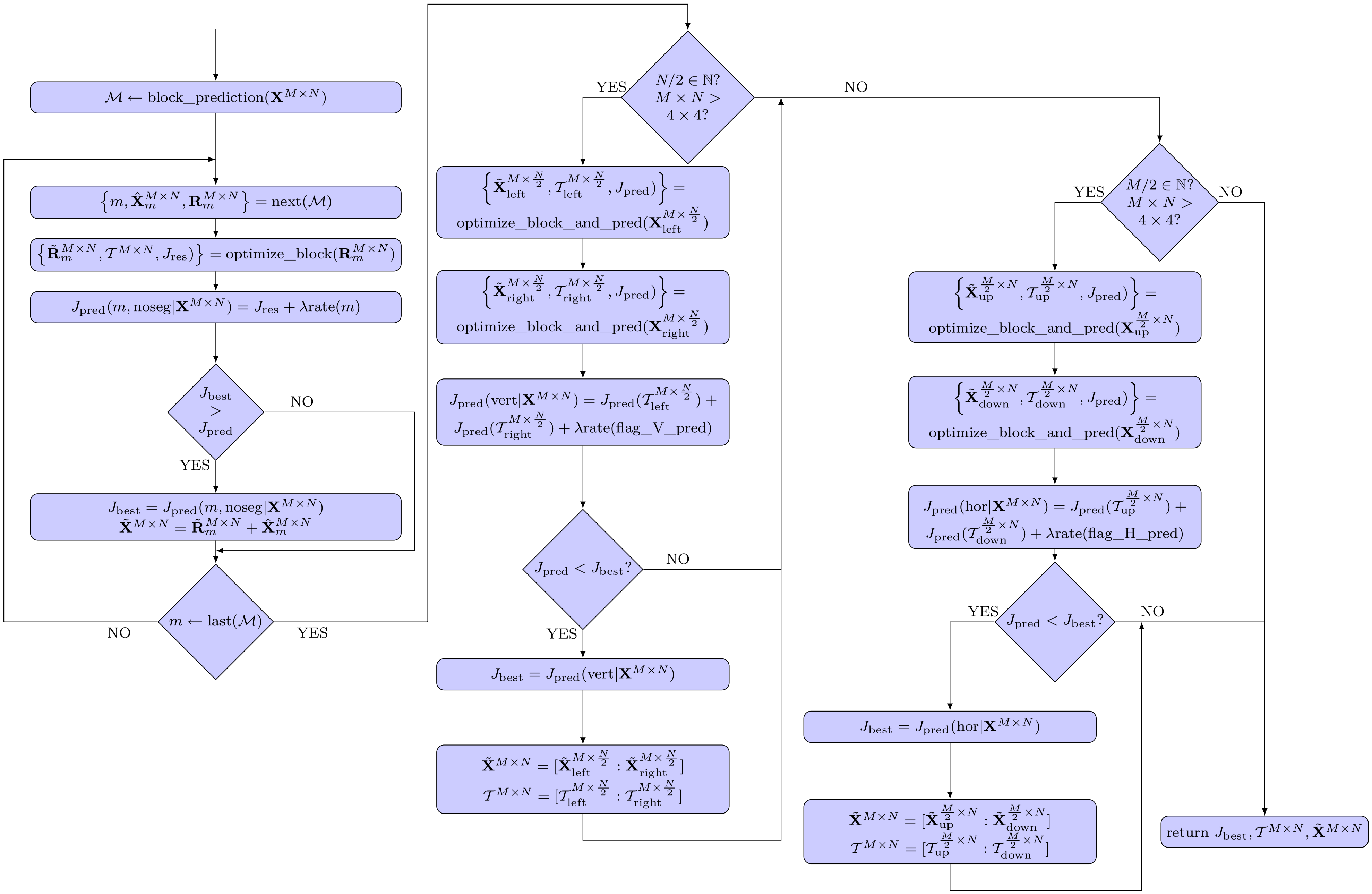

$\mathcal{M} \leftarrow$ block_prediction$(\mathbf{X}^{M\times N})$
$\left\{m, \hat{\mathbf{X}}_m^{M\times N}, \mathbf{R}_m^{M\times N}\right\} =$ next$(\mathcal{M})$
$\left\{\tilde{\mathbf{R}}_m^{M\times N}, \mathcal{T}^{M\times N}, J_{\text{res}}\right\} =$ optimize_block$(\mathbf{R}_m^{M\times N})$
$J_{\text{pred}}(m, \text{noseg}|\mathbf{X}^{M\times N}) = J_{\text{res}} + \lambda\text{rate}(m)$
$J_{\text{best}} > J_{\text{pred}}$
NO
YES
$J_{\text{best}} = J_{\text{pred}}(m, \text{noseg}|\mathbf{X}^{M\times N})$
$\tilde{\mathbf{X}}^{M\times N} = \tilde{\mathbf{R}}_m^{M\times N} + \hat{\mathbf{X}}_m^{M\times N}$
$m \leftarrow$ last$(\mathcal{M})$
NO
YES
$N/2 \in \mathbb{N}$? $M \times N >$ 4×4?
YES
NO
$\left\{\tilde{\mathbf{X}}_{\text{left}}^{M\times\frac{N}{2}}, \mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}, J_{\text{pred}}\right\} =$ optimize_block_and_pred$(\mathbf{X}_{\text{left}}^{M\times\frac{N}{2}})$
$\left\{\tilde{\mathbf{X}}_{\text{right}}^{M\times\frac{N}{2}}, \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}, J_{\text{pred}}\right\} =$ optimize_block_and_pred$(\mathbf{X}_{\text{right}}^{M\times\frac{N}{2}})$
$J_{\text{pred}}(\text{vert}|\mathbf{X}^{M\times N}) = J_{\text{pred}}(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}) +$ $J_{\text{pred}}(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}) + \lambda\text{rate}(\text{flag\_V\_pred})$
$J_{\text{pred}} < J_{\text{best}}$?
NO
YES
$J_{\text{best}} = J_{\text{pred}}(\text{vert}|\mathbf{X}^{M\times N})$
$\tilde{\mathbf{X}}^{M\times N} = [\tilde{\mathbf{X}}_{\text{left}}^{M\times\frac{N}{2}} : \tilde{\mathbf{X}}_{\text{right}}^{M\times\frac{N}{2}}]$
$\mathcal{T}^{M\times N} = [\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}} : \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}]$
$M/2 \in \mathbb{N}$? $M \times N >$ 4×4?
YES
NO
$\left\{\tilde{\mathbf{X}}_{\text{up}}^{\frac{M}{2}\times N}, \mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}, J_{\text{pred}}\right\} =$ optimize_block_and_pred$(\mathbf{X}_{\text{up}}^{\frac{M}{2}\times N})$
$\left\{\tilde{\mathbf{X}}_{\text{down}}^{\frac{M}{2}\times N}, \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}, J_{\text{pred}}\right\} =$ optimize_block_and_pred$(\mathbf{X}_{\text{down}}^{\frac{M}{2}\times N})$
$J_{\text{pred}}(\text{hor}|\mathbf{X}^{M\times N}) = J_{\text{pred}}(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}) +$ $J_{\text{pred}}(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}) + \lambda\text{rate}(\text{flag\_H\_pred})$
$J_{\text{pred}} < J_{\text{best}}$?
YES
NO
$J_{\text{best}} = J_{\text{pred}}(\text{hor}|\mathbf{X}^{M\times N})$
$\tilde{\mathbf{X}}^{M\times N} = [\tilde{\mathbf{X}}_{\text{up}}^{\frac{M}{2}\times N} : \tilde{\mathbf{X}}_{\text{down}}^{\frac{M}{2}\times N}]$
$\mathcal{T}^{M\times N} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$
return $J_{\text{best}}, \mathcal{T}^{M\times N}, \tilde{\mathbf{X}}^{M\times N}$

Figure I.3: Block diagram of the optimization function for prediction.

$$\left\{\tilde{\mathbf{R}}^{M\times N}, \mathcal{T}^{M\times N}, J_{\text{res}}\right\} = \text{optimize\_block}(\mathbf{R}^{M\times N})$$

Step 1 If the block dimension to be optimized is 1 × 1, go to Step 2, otherwise go to Step 4

Step 2 Search for the best index in the dictionary of patterns with dimension 1 × 1, that results in the minimum Lagrangian cost, given by the distortion between the residue value and the chosen codeword (that is, the squared difference between the residue and the dictionary codeword), added with λ lambda times the rate spent to send the index from the dictionary ($J_{\text{res}} = ||\mathbf{R}^{1\times1} - \mathcal{C}_i||_2 + \lambda\text{rate}(\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{1\times1})$)

Step 3 Return the reconstructed residue, the segmentation tree and the associated cost ($\tilde{\mathbf{R}}_m^{M\times N}, \mathcal{T}^{M\times N}, J_{\text{res}}$). Note that the residue will be used for block reconstruction, adding its value to the prediction in the prediction optimization function.

Step 4 Search the best index in the dictionary of scale $M \times N$, that is divided in J segments, where each dictionary segment represents the level of origin of the codeword. To represent the index, it is necessary to indicate the codewords respective dictionary segment (j), and which element belonging to that segment was chosen ($\mathcal{C}_i^{(j)}$). The best index is the one that results in the lowest cost, that is given by the distortion of the residue being represented by the dictionary pattern (that is, the L_2 norm of the difference between the residue pattern and the chosen dictionary element, added with λ times the rate necessary to send the index (that now will be the sum of rates spent to send the segment of the dictionary to which the codeword belongs along with the respective index of the dictionary segments and the flag signalizing that the block will not be further divided ($J_{\text{res}} = ||\mathbf{R}^{M\times N} - \mathcal{C}_i||_2 + \lambda\text{rate}(j) + \lambda\text{rate}(\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{M\times N}(j)) + \lambda\text{rate}(\text{flag\_NOSEG})$).

Step 5 If the block can be divided vertically (that is, if the block vertical dimension is divisible by two), go to Step 6, otherwise go to Step 12.

Step 6 Call the optimize_block($\mathbf{R}^{M\times\frac{N}{2}}$) function for the segment residue to the left, and keep the coding cost of the subblock $J(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}})$, the reconstructed residue subblock ($\tilde{\mathbf{R}}_{\text{left}}^{M\times\frac{N}{2}}$ and its segmentation tree $\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}$

Step 7 Call the optimize_block($\mathbf{R}^{M\times\frac{N}{2}}$) function for the segment residue to the right, and keep the coding cost of the subblock $J(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}})$, the reconstructed residue subblock ($\tilde{\mathbf{R}}_{\text{right}}^{M\times\frac{N}{2}}$ and its segmentation tree $\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}$

Step 8 Calculate the Lagrangian cost for vertical segmentation J_{ver} as the sum of the Lagrangian cost of each half, $J(\mathcal{T}^{\text{left}}) + J(\mathcal{T}^{\text{right}})$, added to λ times the rate necessary to indicate block vertical segmentation (flag_V), valid for the residue only, not for the prediction).

Step 9 If the recently calculated Lagrangian cost is the lowest achieved so far, update the best Lagrangian cost with this value $J_{\text{best}} = J_{\text{ver}}$, and keep the reconstructed residual block and the segmentation tree as the vertical concatenation of both segments ($\tilde{\mathbf{R}}^{M\times N} = [\tilde{\mathbf{R}}_{\text{left}}^{M\times\frac{N}{2}} : \tilde{\mathbf{R}}_{\text{right}}^{M\times\frac{N}{2}}]; \mathcal{T}^{M\times N} = [\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}} : \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}]$)

Step 10 If the block can be divided horizontally (that is, if the block horizontal dimension is divisible by two), go to Step 11, otherwise go to Step 15.

Step 11 Call the optimize_block($\mathbf{R}^{\frac{M}{2}\times N}$) function for the upper segment residue, and keep the coding cost of the subblock $J(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N})$, the reconstructed residue subblock ($\tilde{\mathbf{R}}_{\text{up}}^{\frac{M}{2}\times N}$ and its segmentation tree $\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}$

Step 12 Call the optimize_block($\mathbf{R}^{\frac{M}{2}\times N}$) function for the lower segment residue, and keep the coding cost of the subblock $J(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N})$, the reconstructed residue subblock ($\tilde{\mathbf{R}}_{\text{down}}^{\frac{M}{2}\times N}$ and its segmentation tree $\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}$

Step 13 Calculate the Lagrangian cost for horizontal segmentation J_{hor} as the sum of the Lagrangian cost of each half, $J(\mathcal{T}^{\text{up}}) + J(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N})$, added to λ times the rate necessary to indicate block horizontal segmentation (flag_H, valid for the residue only, not for the prediction).

Step 14 If the recently calculated Lagrangian cost is the lowest achieved so far, update the best Lagrangian cost with this value $J_{\text{best}} = J_{\text{hor}}$, and keep the reconstructed residual block and the segmentation tree as the horizontal concatenation of both segments ($\tilde{\mathbf{R}}^{M\times N} = [\tilde{\mathbf{R}}_{\text{up}}^{\frac{M}{2}\times N} : \tilde{\mathbf{R}}_{\text{down}}^{\frac{M}{2}\times N}]; \mathcal{T}^{M\times N} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$)

Step 15 Return the best Lagrangian cost $J_{\text{res}} = J_{\text{best}}$, the reconstructed residue block $\hat{\mathbf{R}}^{M\times N}$ and its respective segmentation tree $\mathcal{T}^{M\times N}$.

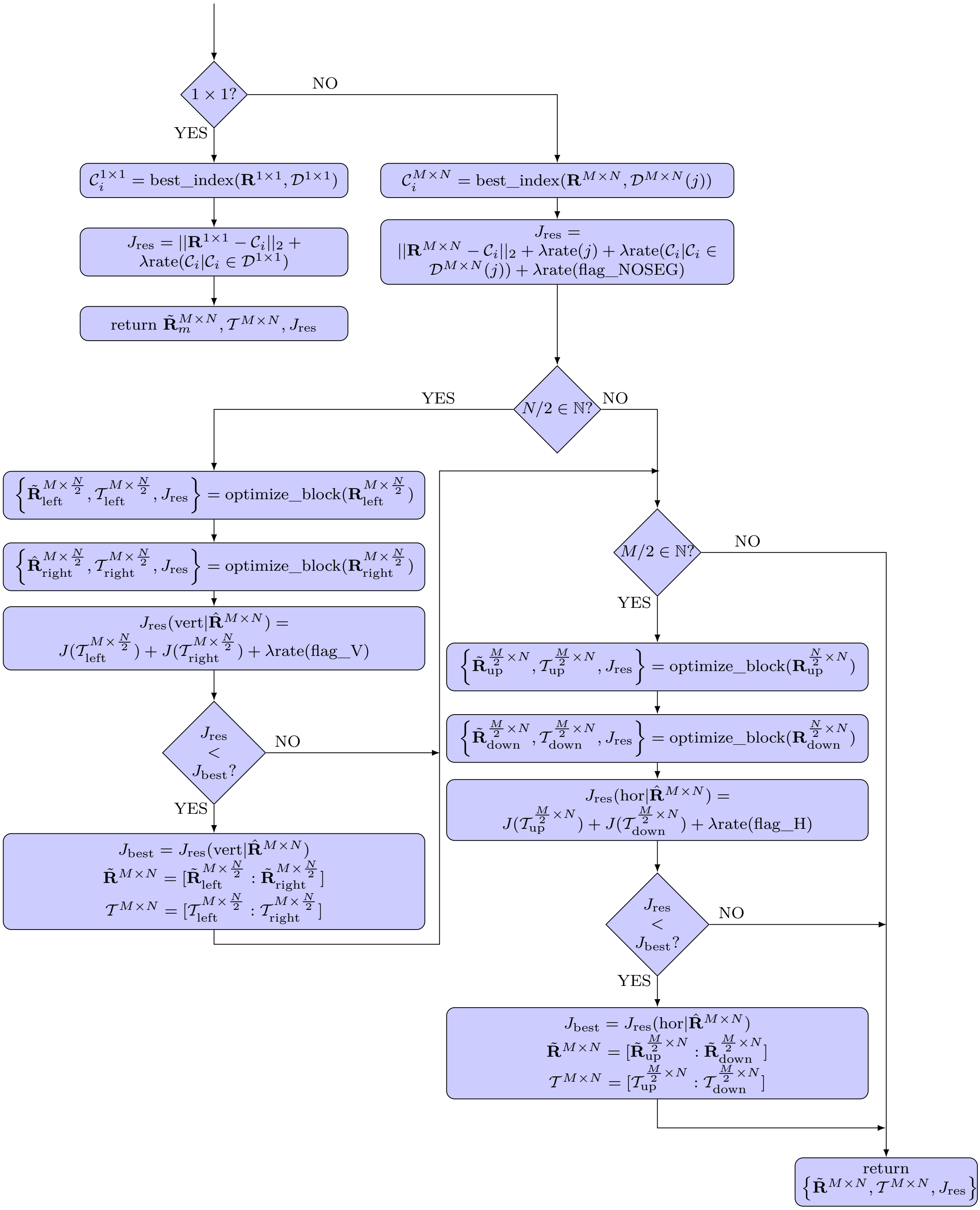

1×1?
NO
YES
$\mathcal{C}_i^{1\times 1} = \text{best\_index}(\mathbf{R}^{1\times 1}, \mathcal{D}^{1\times 1})$
$J_{\text{res}} = ||\mathbf{R}^{1\times 1} - \mathcal{C}_i||_2 + \lambda\text{rate}(\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{1\times 1})$
return $\tilde{\mathbf{R}}_m^{M\times N}, \mathcal{T}^{M\times N}, J_{\text{res}}$
$\mathcal{C}_i^{M\times N} = \text{best\_index}(\mathbf{R}^{M\times N}, \mathcal{D}^{M\times N}(j))$
$J_{\text{res}} = ||\mathbf{R}^{M\times N} - \mathcal{C}_i||_2 + \lambda\text{rate}(j) + \lambda\text{rate}(\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{M\times N}(j)) + \lambda\text{rate}(\text{flag\_NOSEG})$
YES
$N/2 \in \mathbb{N}$?
NO
$\left\{\tilde{\mathbf{R}}_{\text{left}}^{M\times\frac{N}{2}}, \mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}, J_{\text{res}}\right\} = \text{optimize\_block}(\mathbf{R}_{\text{left}}^{M\times\frac{N}{2}})$
$\left\{\tilde{\mathbf{R}}_{\text{right}}^{M\times\frac{N}{2}}, \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}, J_{\text{res}}\right\} = \text{optimize\_block}(\mathbf{R}_{\text{right}}^{M\times\frac{N}{2}})$
$J_{\text{res}}(\text{vert}|\hat{\mathbf{R}}^{M\times N}) = J(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}) + J(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}) + \lambda\text{rate}(\text{flag\_V})$
$J_{\text{res}} < J_{\text{best}}$?
NO
YES
$J_{\text{best}} = J_{\text{res}}(\text{vert}|\hat{\mathbf{R}}^{M\times N})$
$\tilde{\mathbf{R}}^{M\times N} = [\tilde{\mathbf{R}}_{\text{left}}^{M\times\frac{N}{2}} : \tilde{\mathbf{R}}_{\text{right}}^{M\times\frac{N}{2}}]$
$\mathcal{T}^{M\times N} = [\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}} : \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}]$
$M/2 \in \mathbb{N}$?
NO
YES
$\left\{\tilde{\mathbf{R}}_{\text{up}}^{\frac{M}{2}\times N}, \mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}, J_{\text{res}}\right\} = \text{optimize\_block}(\mathbf{R}_{\text{up}}^{\frac{N}{2}\times N})$
$\left\{\tilde{\mathbf{R}}_{\text{down}}^{\frac{M}{2}\times N}, \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}, J_{\text{res}}\right\} = \text{optimize\_block}(\mathbf{R}_{\text{down}}^{\frac{N}{2}\times N})$
$J_{\text{res}}(\text{hor}|\hat{\mathbf{R}}^{M\times N}) = J(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}) + J(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}) + \lambda\text{rate}(\text{flag\_H})$
$J_{\text{res}} < J_{\text{best}}$?
NO
YES
$J_{\text{best}} = J_{\text{res}}(\text{hor}|\hat{\mathbf{R}}^{M\times N})$
$\tilde{\mathbf{R}}^{M\times N} = [\tilde{\mathbf{R}}_{\text{up}}^{\frac{M}{2}\times N} : \tilde{\mathbf{R}}_{\text{down}}^{\frac{M}{2}\times N}]$
$\mathcal{T}^{M\times N} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$
return $\left\{\tilde{\mathbf{R}}^{M\times N}, \mathcal{T}^{M\times N}, J_{\text{res}}\right\}$

Figure I.4: Diagram for optimization of the residue encoding.

I.3 Entropy coding

analize($\mathcal{T}^{M\times N}$)

Step 1 If the block level to be analyzed is equal to 1×1, go to Step 2, otherwise go to Step 3

Step 2 Encode the dictionary index code_index($\mathcal{C}_i | \mathcal{C}_i \in \mathcal{D}^{1\times 1}$) and return.

Step 3 In case the block has been divided vertically, go to Step 4, otherwise go to Step 9

Step 4 Encode the flag that indicates vertical segmentation of the block. In case prediction hasn't been sent yet, use the flag that indicates both segmentation of the prediction and the block (flag_V_pred), otherwise use the flag that indicates segmentation of the block only ($flag\_V$)

Step 5 In case prediction hasn't been sent yet, encode the prediction mode.

Step 6 Recursively analyze the subblock to the left, encoding all flags, possible prediction modes, and subblock indexes (analize($\mathcal{T}_{\text{left}}^{M\times \frac{N}{2}}$))

Step 7 Recursively analyze the subblock to the right, encoding all flags, possible prediction modes, and subblock indexes (analize($\mathcal{T}_{\text{right}}^{M\times \frac{N}{2}}$))

Step 8 Return

Step 9 In case the block has been divided horizontally, go to Step 10, otherwise go to Step 15

Step 10 Encode the flag that indicates horizontal segmentation of the block. In case prediction hasn't been sent yet, use the flag that indicates both segmentation of the prediction and the block (flag_H_pred), otherwise use the flag that indicates segmentation of the block only ($flag\_H$)

Step 11 In case prediction hasn't been sent yet, encode the prediction mode.

Step 12 Recursively analyze the upper subblock, encoding all flags, possible prediction modes, and subblock indexes (analize($\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}$))

Step 13 Recursively analyze the lower subblock, encoding all flags, possible prediction modes, and subblock indexes (analize($\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}$))

Step 14 Return

Step 15 Encode the flag that indicates that the block (and also the prediction) has not been divided (flag_NOSEG)

Step 16 In case prediction hasn't been sent yet, encode the prediction mode.

Step 17 Encode the dictionary segment (j) that indicates the level of origin of the block (code_dic_seg$\mathcal{D}^{M\times N}(j)$)

Step 18 Encode the index of the chosen codeword (code_index($\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{M\times N}(j)$))

Step 19 Return

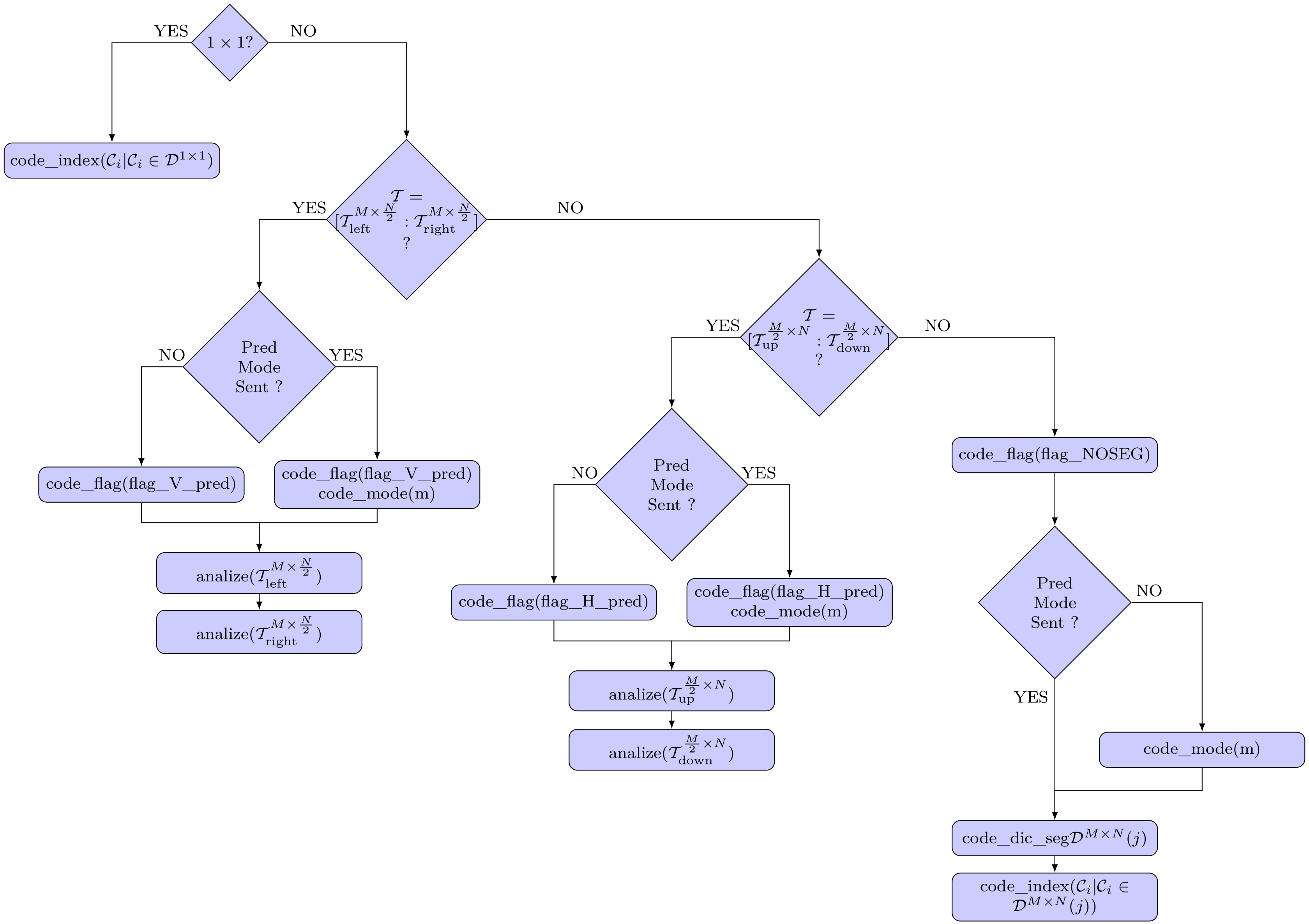

1 × 1?
YES
NO
code_index($\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{1\times1}$)
$\mathcal{T} = [\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}} : \mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}]$?
YES
NO
Pred Mode Sent ?
NO
YES
code_flag(flag_V_pred)
code_flag(flag_V_pred) code_mode(m)
analize($\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}}$)
analize($\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}}$)
$\mathcal{T} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$?
YES
NO
Pred Mode Sent ?
NO
YES
code_flag(flag_H_pred)
code_flag(flag_H_pred) code_mode(m)
analize($\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N}$)
analize($\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}$)
code_flag(flag_NOSEG)
Pred Mode Sent ?
NO
YES
code_mode(m)
code_dic_seg$\mathcal{D}^{M\times N}(j)$
code_index($\mathcal{C}_i|\mathcal{C}_i \in \mathcal{D}^{M\times N}(j)$)

Figure I.5: Diagram for the analysis stage. Here the coded parameters will be incorporated into the output bitstream.

I.4 Dictionary update

$$\mathcal{D} \leftarrow \text{update\_dictionary}(\mathcal{T}^{M \times N})$$

Step 1 In case the block has been divided vertically, go to Step 2, otherwise go to Step 5

Step 2 Update the dictionary with the subblock to the left (update_dictionary($\mathcal{T}_{\text{left}}^{M \times \frac{N}{2}}$)), and keep the codeword index that indicates the subblock that was added to the dictionary ($\mathcal{C}_{left}$)

Step 3 Update the dictionary with the subblock to the right (update_dictionary($\mathcal{T}_{\text{right}}^{M \times \frac{N}{2}}$)), and keep the codeword index that indicates the subblock that was added to the dictionary ($\mathcal{C}_{right}$)

Step 4 Update the dictionary with the concatenation of both subblocks (dic_update($\mathcal{C}_{left}, \mathcal{C}_{right}$)), and return the new index ($\mathcal{D}_i$). Note that this function is related to the update of several scales, and also the super update routine, adding symmetric blocks ($-\mathcal{C}_i$), rotated blocks ($\mathrm{T}\{\mathcal{C}_i\}$) or dislocated blocks ($\mathcal{C}_i(x + \delta)$).

Step 5 In case the block has been divided horizontally, go to Step 6, otherwise go to Step 9

Step 6 Update the dictionary with the upper subblock (update_dictionary($\mathcal{T}_{\text{up}}^{\frac{M}{2} \times N}$)), and keep the codeword index that indicates the subblock that was added to the dictionary ($\mathcal{C}_{up}$)

Step 7 Update the dictionary with the lower subblock (update_dictionary($\mathcal{T}_{\text{down}}^{\frac{M}{2} \times N}$)), and keep the codeword index that indicates the subblock that was added to the dictionary ($\mathcal{C}_{down}$)

Step 8 Update the dictionary with the concatenation of both subblocks (dic_update($\mathcal{C}_{up}, \mathcal{C}_{down}$)), and return the new index ($\mathcal{C}_i$). Note that this function is related to the update of several scales, and also the super update routine, adding symmetric blocks ($-\mathcal{C}_i$), rotated blocks ($\mathrm{T}\{\mathcal{C}_i\}$) or dislocated blocks ($\mathcal{C}_i(x + \delta)$).

Step 9 Return the chosen index that represents the block ($\mathcal{C}_i$)

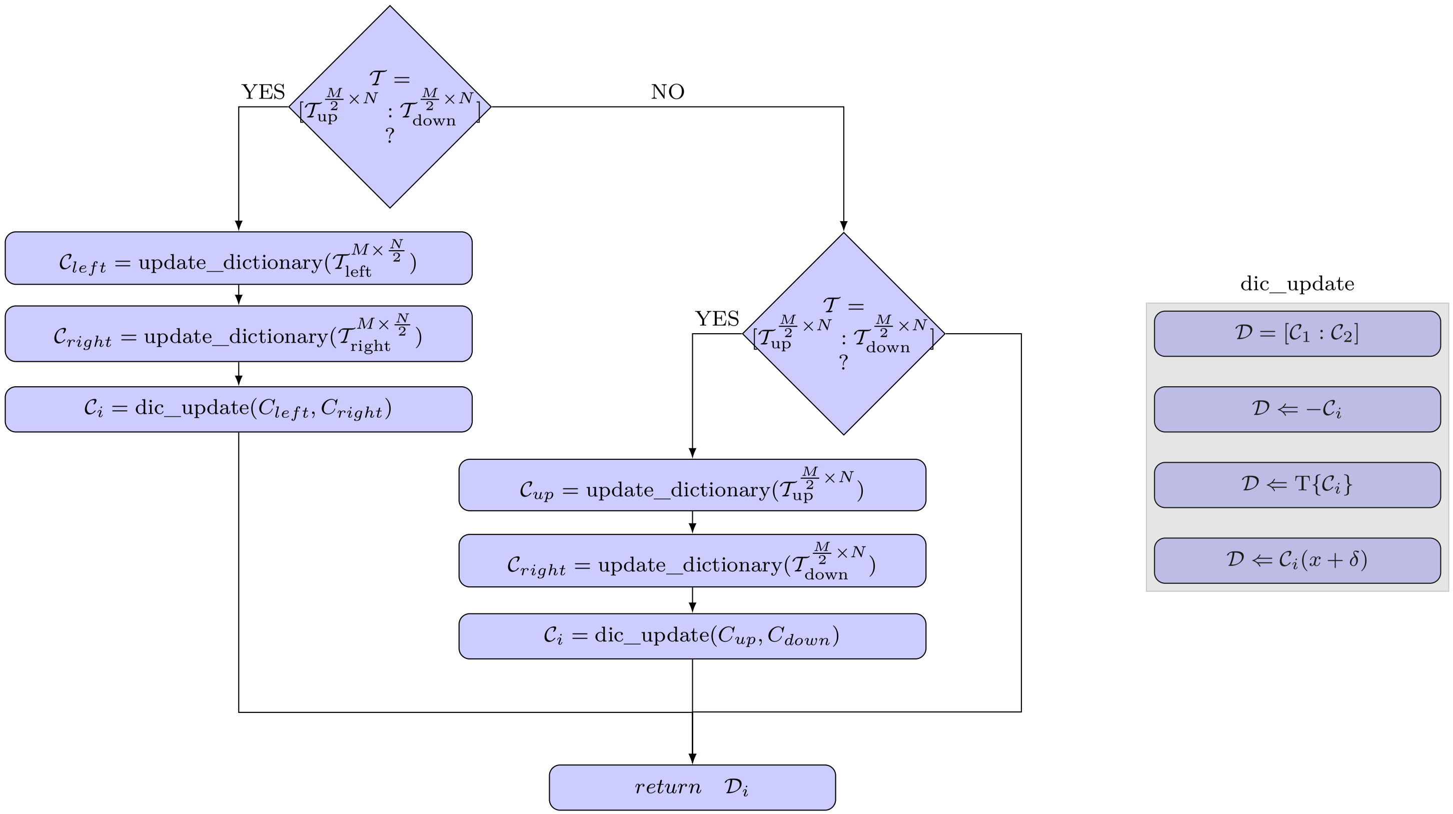

YES
$\mathcal{T} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$?
NO
$\mathcal{C}_{left} =$ update_dictionary$(\mathcal{T}_{\text{left}}^{M\times\frac{N}{2}})$
$\mathcal{C}_{right} =$ update_dictionary$(\mathcal{T}_{\text{right}}^{M\times\frac{N}{2}})$
$\mathcal{C}_i =$ dic_update(C_{left}, C_{right})
YES
$\mathcal{T} = [\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N} : \mathcal{T}_{\text{down}}^{\frac{M}{2}\times N}]$?
$\mathcal{C}_{up} =$ update_dictionary$(\mathcal{T}_{\text{up}}^{\frac{M}{2}\times N})$
$\mathcal{C}_{right} =$ update_dictionary$(\mathcal{T}_{\text{down}}^{\frac{M}{2}\times N})$
$\mathcal{C}_i =$ dic_update(C_{up}, C_{down})
$return \quad \mathcal{D}_i$
dic_update
$\mathcal{D} = [\mathcal{C}_1 : \mathcal{C}_2]$
$\mathcal{D} \Leftarrow -\mathcal{C}_i$
$\mathcal{D} \Leftarrow \mathrm{T}\{\mathcal{C}_i\}$
$\mathcal{D} \Leftarrow \mathcal{C}_i(x+\delta)$

Figure I.6: Diagram for the dictionary update routine.

Apêndice J

List of publications

SUMMARY: This appendix presents the published work, resulted from the investigation done during this thesis. The scientific contributions appeared in international and national conference proceedings and international journals.

- Proceedings in Conferences

  i) GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., DE FARIA, S. M. M., SILVA, V. M. M., "Fast Implementation for Multiscale Recurrent Pattern Image Coding", In: *7th Conference on Telecommunications*, Santa Maria da Feira, Portugal, May 2009

  ii) GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., DE FARIA, S. M. M., DE CARVALHO, M. B., "Improving Multiscale Recurrent Pattern Image Coding with Least-squares prediction mode", In: *Proceedings of the IEEE International Conference on Image Processing*, Cairo, Egypt, pp. 2813-2816, November 2009

  iii) GRAZIOSI, D. B., N. M. M., DA SILVA, E. A. B., DE FARIA, S. M. M., DE CARVALHO, M. B., SILVA, V. M. M., "Codificação de Imagens com Predição Adaptativa Baseada no Critério de Mínimos Quadrados", In: *Anais XXVII do Simpósio Brasileiro de Telecomunicações*, Blumenau, SC, Brasil, September 2009

  iv) GRAZIOSI, D. B., RODRIGUES, N. M. M., PAGLIARI, C. L., DA SILVA, E. A. B., DE FARIA, S. M. M., PEREZ, M. M., DE CARVALHO, M. B., "Multiscale Recurrent Pattern Matching Approach for Depth Map Coding", In: *Proceedings of the 29th Picture Coding Symposium - PCS2010*, Nagoya, Japan, pp. 294-297, December 2010

- Articles in Journals

i) GRAZIOSI, D. B., RODRIGUES, N. M. M., PAGLIARI, C. L., DA SILVA, E. A. B., DE FARIA, S. M. M., DE CARVALHO, M. B.,"Compressing Depth Maps using Multiscale Recurrent Pattern Image Coding", In: *Electronics Letters*, Vol. 46, No. 5, pp. 340-341, March, 2010

- In preparation or submitted

  i) GRAZIOSI, D. B., RODRIGUES, N. M. M., PAGLIARI, C. L., DA SILVA, E. A. B., DE FARIA, S. M. M., PEREZ, M. M., DE CARVALHO, M. B.,"Joint coding of texture and depth using multiscale recurrent pattern matching", in preparation for submission

  ii) GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., DE FARIA, S. M. M., DE CARVALHO, M. B., SILVA, V. M. M., "On the performance of lossless image compression using multiscale recurrent pattern matching", in preparation for submission

  iii) GRAZIOSI, D. B., PAGLIARI, C. L., RODRIGUES, N. M. M., DA SILVA, E. A. B., DE FARIA, S. M. M., DE CARVALHO, M. B.,"Codificação de mapas de profundidade usando casamento de padrões multiescalas", submitted to : SBrT 2011

Apêndice K

Test set

SUMMARY: Available in this appendix are the original images used in simulations throughout the thesis. The same images can be downloaded from MMP's website (`http://www.lps.ufrj.br/profs/eduardo/mmp`).

K.1 Smooth images

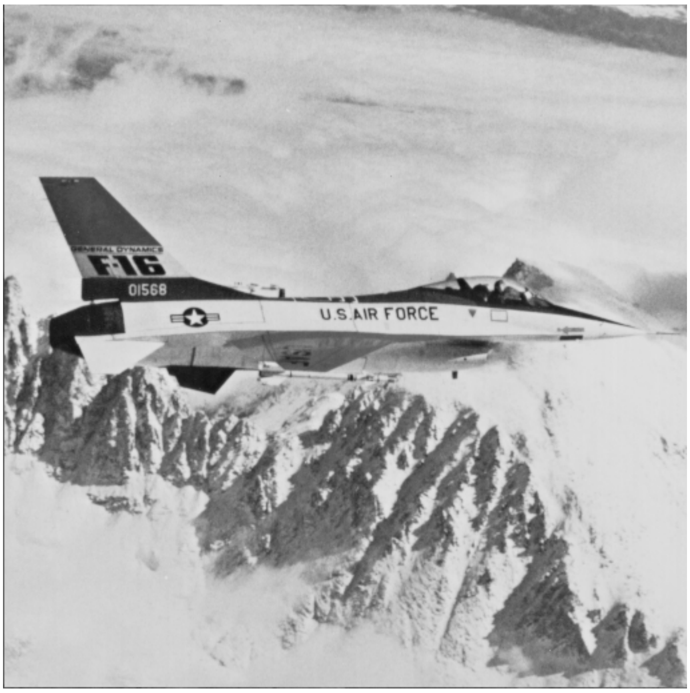


Figure K.1: Airplane (512 × 512).

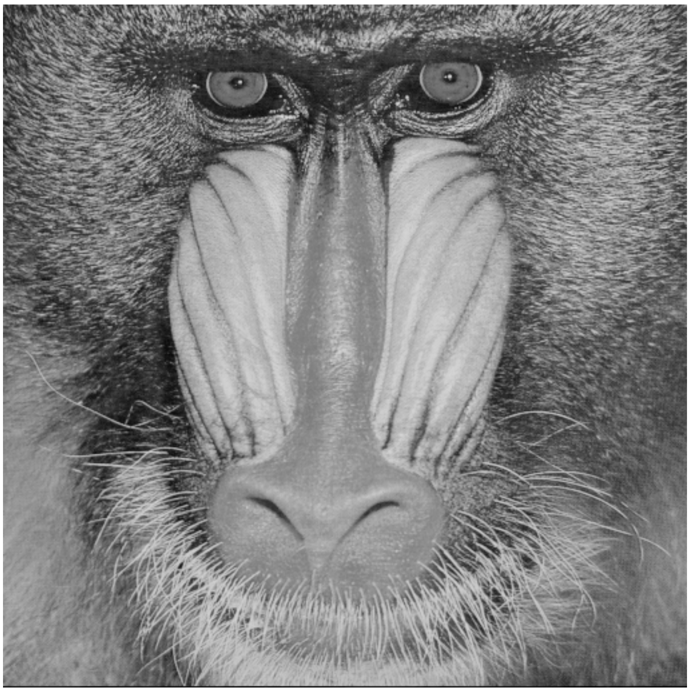


Figure K.2: Baboon (512 × 512).

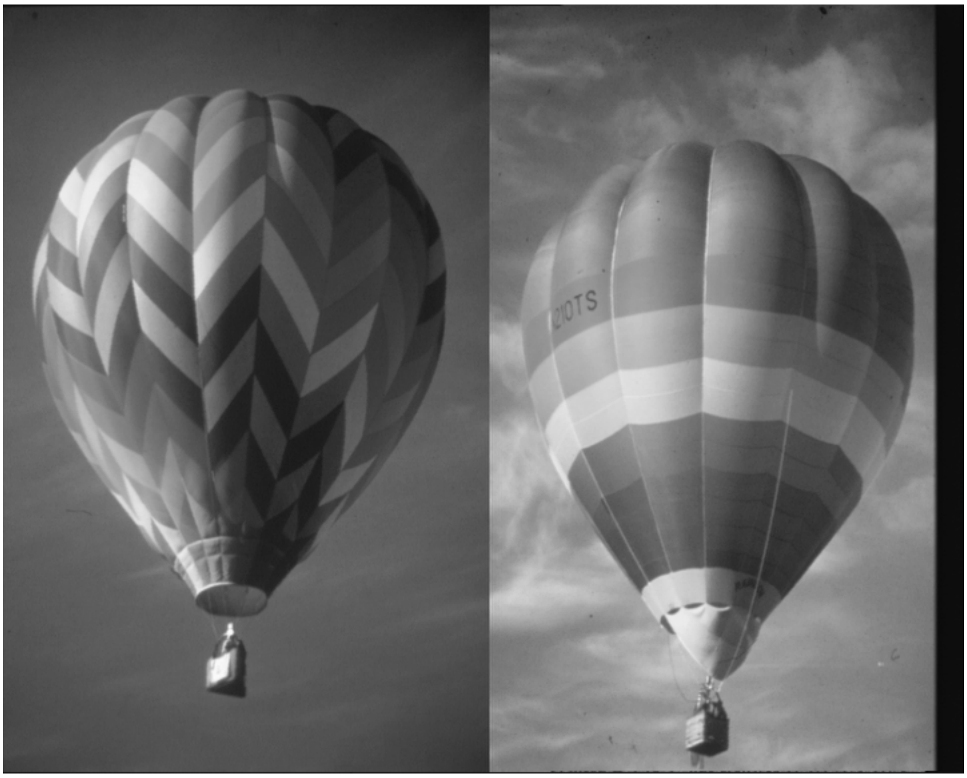


Figure K.3: Balloon (720 × 576).

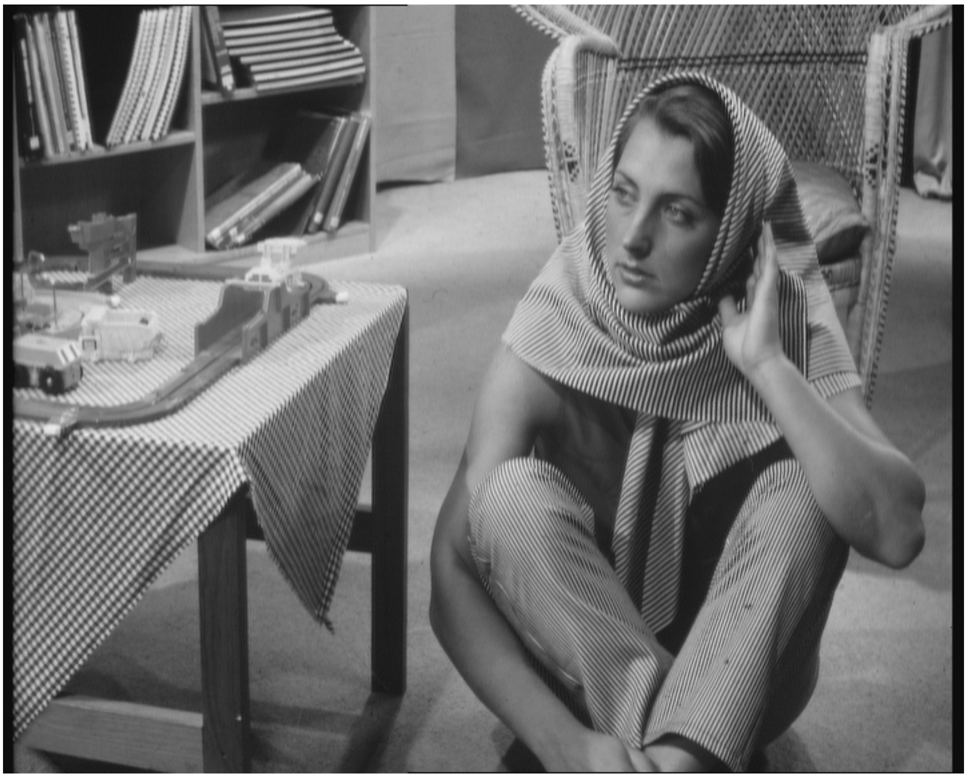


Figure K.4: Barb (720 × 576).

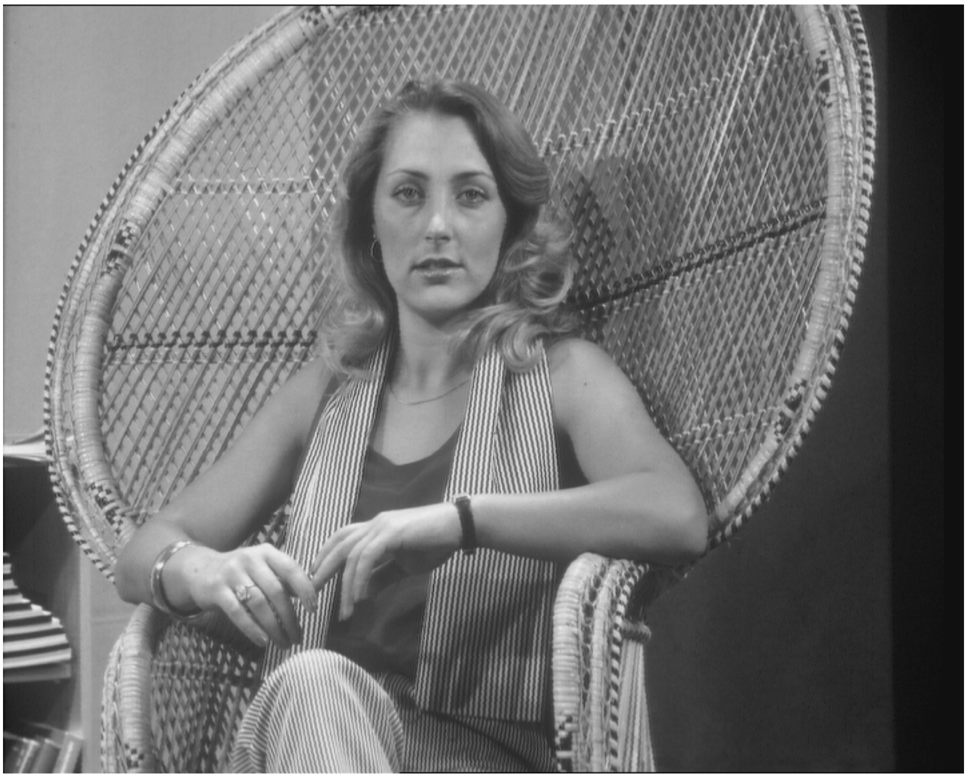


Figure K.5: Barb2 (720 × 576).

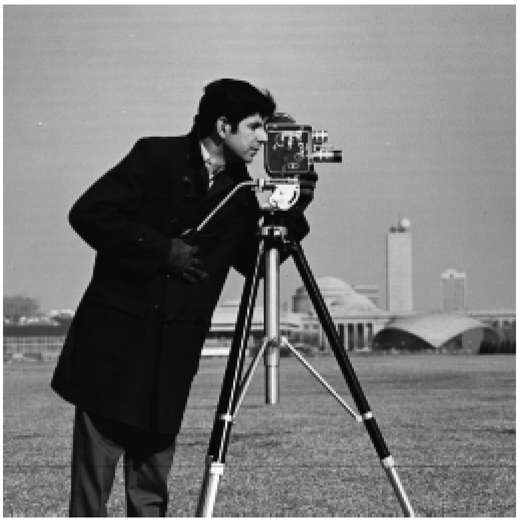


Figure K.6: Cameraman (256 × 256).

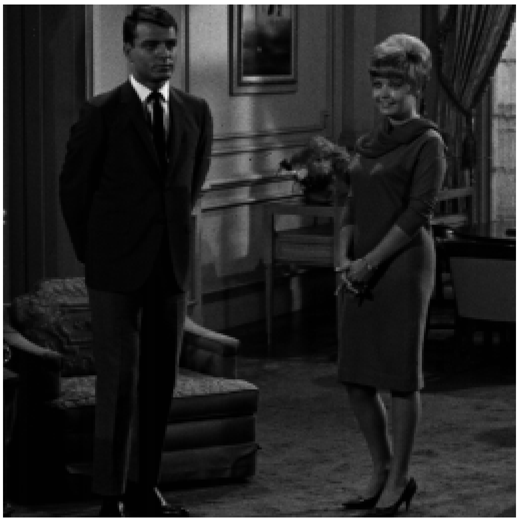


Figure K.7: Couple (256 × 256).

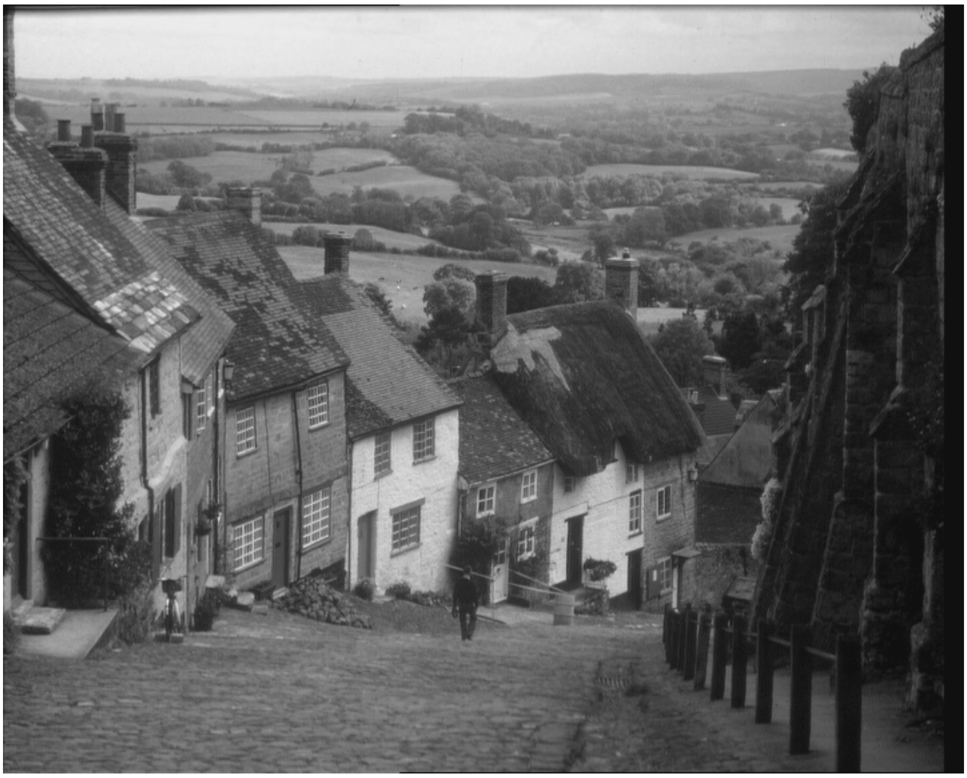


Figure K.8: Goldhill (720 × 576).

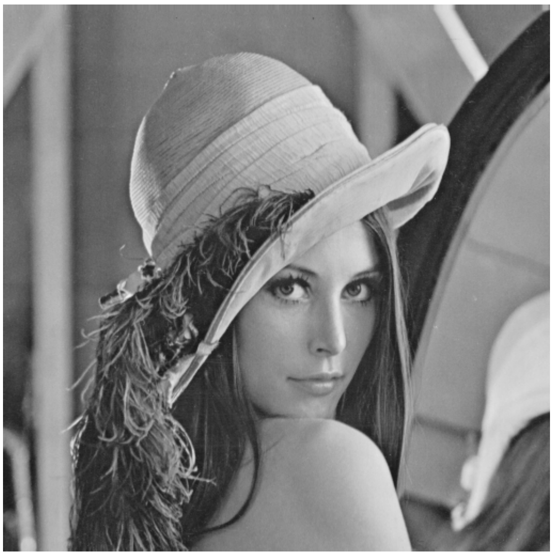


Figure K.9: Lena (512 × 512).

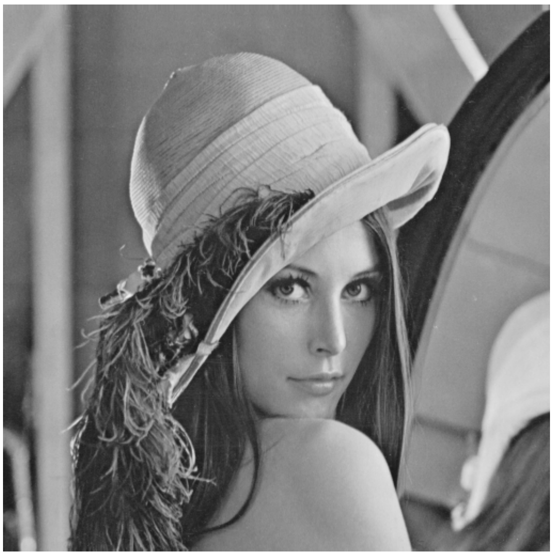


Figure K.10: Lennagrey (512 × 512).

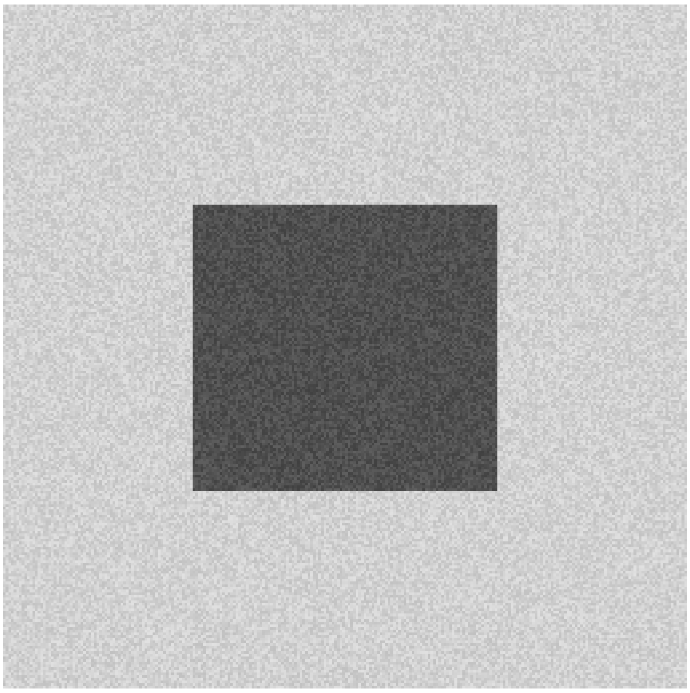


Figure K.11: Noisesquare (256 × 256).

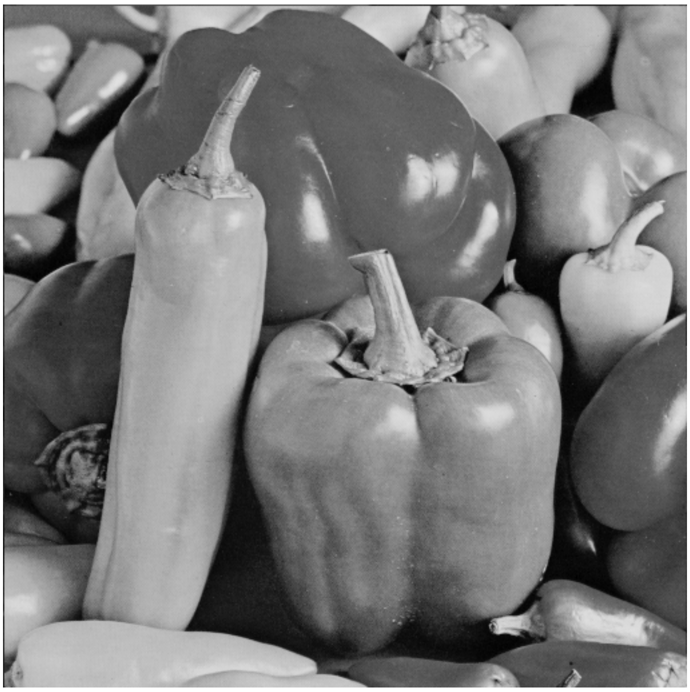


Figure K.12: Peppers (512 × 512).

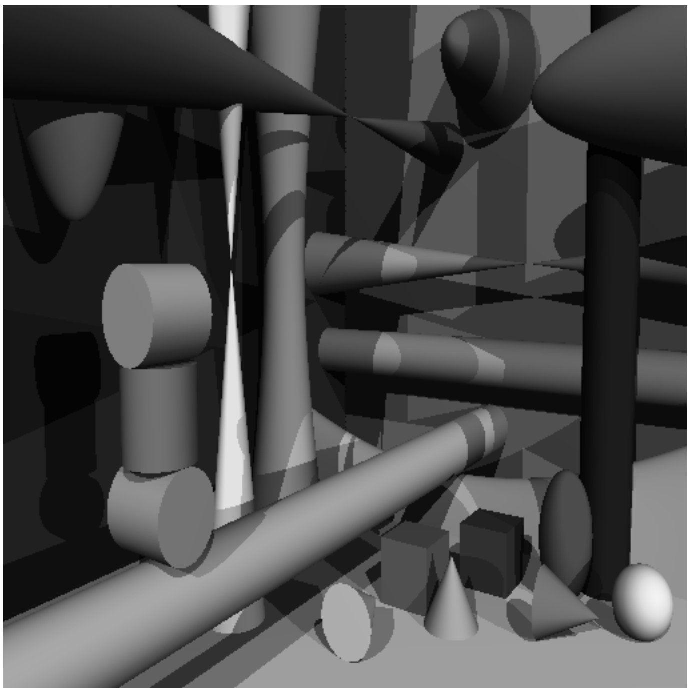
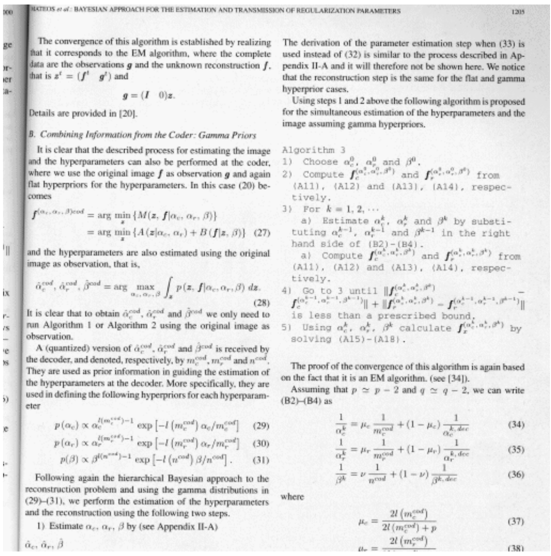


Figure K.13: Shapes (512 × 512).

K.2 Compound images



MATEOS *et al.*: BAYESIAN APPROACH FOR THE ESTIMATION AND TRANSMISSION OF REGULARIZATION PARAMETERS 1205

The convergence of this algorithm is established by realizing that it corresponds to the EM algorithm, where the complete data are the observations $\boldsymbol{g}$ and the unknown reconstruction $\boldsymbol{f}$, that is $\boldsymbol{z}^t = (\boldsymbol{f}^t \quad \boldsymbol{g}^t)$ and

$$\boldsymbol{g} = (I \quad 0)\boldsymbol{z}.$$

Details are provided in [20].

B. Combining Information from the Coder: Gamma Priors

It is clear that the described process for estimating the image and the hyperparameters can also be performed at the coder, where we use the original image $\boldsymbol{f}$ as observation $\boldsymbol{g}$ and again flat hyperpriors for the hyperparameters. In this case (20) becomes

$$\begin{aligned}\boldsymbol{f}^{(\alpha_c, \alpha_r, \beta)cod} &= \arg\min_{\boldsymbol{z}} \{M(\boldsymbol{z}, \boldsymbol{f}|\alpha_c, \alpha_r, \beta)\} \\ &= \arg\min_{\boldsymbol{z}} \{A(\boldsymbol{z}|\alpha_c, \alpha_r) + B(\boldsymbol{f}|\boldsymbol{z}, \beta)\} \quad (27)\end{aligned}$$

and the hyperparameters are also estimated using the original image as observation, that is,

$$\hat{\alpha}_c^{cod}, \hat{\alpha}_r^{cod}, \hat{\beta}^{cod} = \arg \max_{\alpha_c, \alpha_r, \beta} \int_{\boldsymbol{z}} p(\boldsymbol{z}, \boldsymbol{f}|\alpha_c, \alpha_r, \beta)\, d\boldsymbol{z}. \quad (28)$$

It is clear that to obtain $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ and $\hat{\beta}^{cod}$ we only need to run Algorithm 1 or Algorithm 2 using the original image as observation.

A (quantized) version of $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ and $\hat{\beta}^{cod}$ is received by the decoder, and denoted, respectively, by m_c^{cod}, m_r^{cod} and n^{cod}. They are used as prior information in guiding the estimation of the hyperparameters at the decoder. More specifically, they are used in defining the following hyperpriors for each hyperparameter

$$p(\alpha_c) \propto \alpha_c^{l(m_c^{cod})-1} \exp\left[-l\left(m_c^{cod}\right)\alpha_c/m_c^{cod}\right] \quad (29)$$

$$p(\alpha_r) \propto \alpha_r^{l(m_r^{cod})-1} \exp\left[-l\left(m_r^{cod}\right)\alpha_r/m_r^{cod}\right] \quad (30)$$

$$p(\beta) \propto \beta^{l(n^{cod})-1} \exp\left[-l\left(n^{cod}\right)\beta/n^{cod}\right]. \quad (31)$$

Following again the hierarchical Bayesian approach to the reconstruction problem and using the gamma distributions in (29)–(31), we perform the estimation of the hyperparameters and the reconstruction using the following two steps.

1) Estimate α_c, α_r, β by (see Appendix II-A)

$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}$

The derivation of the parameter estimation step when (33) is used instead of (32) is similar to the process described in Appendix II-A and it will therefore not be shown here. We notice that the reconstruction step is the same for the flat and gamma hyperprior cases.

Using steps 1 and 2 above the following algorithm is proposed for the simultaneous estimation of the hyperparameters and the image assuming gamma hyperpriors.

```
Algorithm 3
1) Choose $\alpha_c^0$, $\alpha_r^0$ and $\beta^0$.
2) Compute $\boldsymbol{f}_c^{(\alpha_c^0, \alpha_r^0, \beta^0)}$ and $\boldsymbol{f}_r^{(\alpha_c^0, \alpha_r^0, \beta^0)}$ from
   (A11), (A12) and (A13), (A14), respectively.
3) For $k = 1, 2, \cdots$
   a) Estimate $\alpha_c^k$, $\alpha_r^k$ and $\beta^k$ by substituting $\alpha_c^{k-1}$, $\alpha_r^{k-1}$ and $\beta^{k-1}$ in the right
   hand side of (B2)-(B4).
   a) Compute $\boldsymbol{f}_c^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ and $\boldsymbol{f}_r^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ from
   (A11), (A12) and (A13), (A14), respectively.
4) Go to 3 until $\|\boldsymbol{f}_c^{(\alpha_c^k, \alpha_r^k, \beta^k)} - \boldsymbol{f}_c^{(\alpha_c^{k-1}, \alpha_r^{k-1}, \beta^{k-1})}\| + \|\boldsymbol{f}_r^{(\alpha_c^k, \alpha_r^k, \beta^k)} - \boldsymbol{f}_r^{(\alpha_c^{k-1}, \alpha_r^{k-1}, \beta^{k-1})}\|$
   is less than a prescribed bound.
5) Using $\alpha_c^k$, $\alpha_r^k$, $\beta^k$ calculate $\boldsymbol{f}_x^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ by
   solving (A15)-(A18).
```

The proof of the convergence of this algorithm is again based on the fact that it is an EM algorithm. (see [34]).

Assuming that $p \simeq p - 2$ and $q \simeq q - 2$, we can write (B2)–(B4) as

$$\frac{1}{\alpha_c^k} = \mu_c \frac{1}{m_c^{cod}} + (1 - \mu_c)\frac{1}{\alpha_c^{k,\,dec}} \quad (34)$$

$$\frac{1}{\alpha_r^k} = \mu_r \frac{1}{m_r^{cod}} + (1 - \mu_r)\frac{1}{\alpha_r^{k,\,dec}} \quad (35)$$

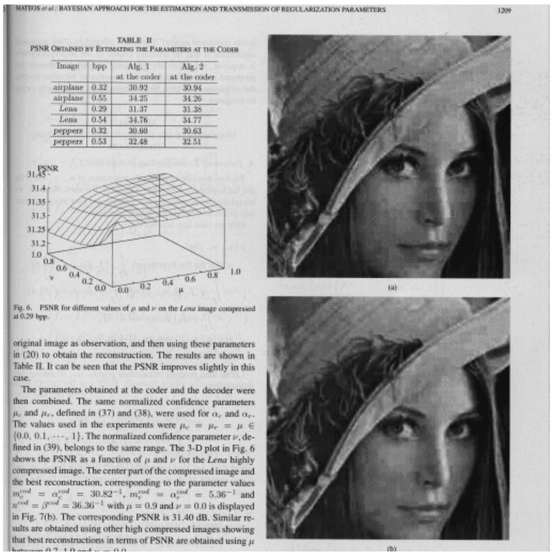
$$\frac{1}{\beta^k} = \nu \frac{1}{n^{cod}} + (1 - \nu)\frac{1}{\beta^{k,\,dec}} \quad (36)$$

where

$$\mu_c = \frac{2l\left(m_c^{cod}\right)}{2l\left(m_c^{cod}\right) + p} \quad (37)$$

$$\mu_r = \frac{2l\left(m_r^{cod}\right)}{\quad} \quad (38)$$

Figure K.14: PP1205 (512 × 512).



MATEOS *et al.*: BAYESIAN APPROACH FOR THE ESTIMATION AND TRANSMISSION OF REGULARIZATION PARAMETERS 1209

TABLE II
PSNR OBTAINED BY ESTIMATING THE PARAMETERS AT THE CODER

Image	bpp	Alg. 1 at the coder	Alg. 2 at the coder
airplane	0.32	30.92	30.94
airplane	0.55	34.25	34.26
Lena	0.29	31.37	31.38
Lena	0.54	34.76	34.77
peppers	0.32	30.60	30.63
peppers	0.53	32.48	32.51

Fig. 6. PSNR for different values of μ and ν on the *Lena* image compressed at 0.29 bpp.

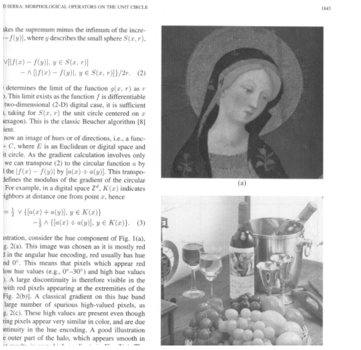
original image as observation, and then using these parameters in (20) to obtain the reconstruction. The results are shown in Table II. It can be seen that the PSNR improves slightly in this case.

The parameters obtained at the coder and the decoder were then combined. The same normalized confidence parameters μ_c and μ_r, defined in (37) and (38), were used for α_c and α_r. The values used in the experiments were $\mu_c = \mu_r = \mu \in \{0.0, 0.1, \cdots, 1\}$. The normalized confidence parameter ν, defined in (39), belongs to the same range. The 3-D plot in Fig. 6 shows the PSNR as a function of μ and ν for the *Lena* highly compressed image. The center part of the compressed image and the best reconstruction, corresponding to the parameter values $m_c^{cod} = \alpha_c^{cod} = 30.82^{-1}$, $m_r^{cod} = \alpha_r^{cod} = 5.36^{-1}$ and $n^{cod} = \beta^{cod} = 36.36^{-1}$ with $\mu = 0.9$ and $\nu = 0.0$ is displayed in Fig. 7(b). The corresponding PSNR is 31.40 dB. Similar results are obtained using other high compressed images showing that best reconstructions in terms of PSNR are obtained using μ between 0.7–1.0 and $\nu = 0.0$.

(a)

(b)

Figure K.15: PP1209 (512 × 512).



D SERRA: MORPHOLOGICAL OPERATORS ON THE UNIT CIRCLE 1843

akes the supremum minus the infimum of the incre- $-f(y)|$, where y describes the small sphere $S(x, r)$,

$$\vee[|f(x) - f(y)|, y \in S(x, r)] - \wedge[|f(x) - f(y)|, y \in S(x, r)]\}/2r. \quad (2)$$

determines the limit of the function $g(x, r)$ as r). This limit exists as the function f is differentiable two-dimensional (2-D) digital case, it is sufficient , taking for $S(x, r)$ the unit circle centered on x exagon). This is the classic Beucher algorithm [8] ient.

now an image of hues or of directions, i.e., a func- C, where E is an Euclidean or digital space and it circle. As the gradient calculation involves only we can transpose (2) to the circular function a by l the $|f(x) - f(y)|$ by $[a(x) \div a(y)]$. This transpo- defines the modulus of the gradient of the circular For example, in a digital space $\mathbb{Z}^d$, $K(x)$ indicates ighbors at distance one from point x, hence

$$= \tfrac{1}{2} \vee \{[a(x) \div a(y)], y \in K(x)\} - \tfrac{1}{2} \wedge \{[a(x) \div a(y)], y \in K(x)\}. \quad (3)$$

stration, consider the hue component of Fig. 1(a), g. 2(a). This image was chosen as it is mostly red d in the angular hue encoding, red usually has hue nd 0°. This means that pixels which appear red low hue values (e.g., 0°–30°) and high hue values). A large discontinuity is therefore visible in the with red pixels appearing at the extremities of the Fig. 2(b)]. A classical gradient on this hue band large number of spurious high-valued pixels, as g. 2(c). These high values are present even though ring pixels appear very similar in color, and are due ntinuity in the hue encoding. A good illustration e outer part of the halo, which appears smooth in

(a)

Figure K.16: SCAN0002 (512 × 512).

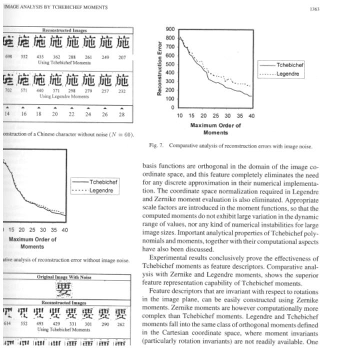


IMAGE ANALYSIS BY TCHEBICHEF MOMENTS 1363

onstruction of a Chinese character without noise ($N = 60$).

ative analysis of reconstruction error without image noise.

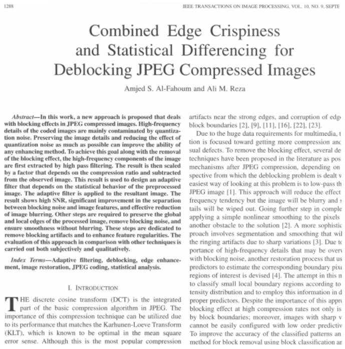
Fig. 7. Comparative analysis of reconstruction errors with image noise.

basis functions are orthogonal in the domain of the image coordinate space, and this feature completely eliminates the need for any discrete approximation in their numerical implementation. The coordinate space normalization required in Legendre and Zernike moment evaluation is also eliminated. Appropriate scale factors are introduced in the moment functions, so that the computed moments do not exhibit large variation in the dynamic range of values, nor any kind of numerical instabilities for large image sizes. Important analytical properties of Tchebichef polynomials and moments, together with their computational aspects have also been discussed.

Experimental results conclusively prove the effectiveness of Tchebichef moments as feature descriptors. Comparative analysis with Zernike and Legendre moments, shows the superior feature representation capability of Tchebichef moments.

Feature descriptors that are invariant with respect to rotations in the image plane, can be easily constructed using Zernike moments. Zernike moments are however computationally more complex than Tchebichef moments. Legendre and Tchebichef moments fall into the same class of orthogonal moments defined in the Cartesian coordinate space, where moment invariants (particularly rotation invariants) are not readily available. One

Figure K.17: SCAN0004 (512 × 512).



1288 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 9, SEPTE

Combined Edge Crispiness and Statistical Differencing for Deblocking JPEG Compressed Images

Amjed S. Al-Fahoum and Ali M. Reza

Abstract—**In this work, a new approach is proposed that deals with blocking effects in JPEG compressed images. High-frequency details of the coded images are mainly contaminated by quantization noise. Preserving the image details and reducing the effect of quantization noise as much as possible can improve the ability of any enhancing method. To achieve this goal along with the removal of the blocking effect, the high-frequency components of the image are first extracted by high pass filtering. The result is then scaled by a factor that depends on the compression ratio and subtracted from the observed image. This result is used to design an adaptive filter that depends on the statistical behavior of the preprocessed image. The adaptive filter is applied to the resultant image. The result shows high SNR, significant improvement in the separation between blocking noise and image features, and effective reduction of image blurring. Other steps are required to preserve the global and local edges of the processed image, remove blocking noise, and ensure smoothness without blurring. These steps are dedicated to remove blocking artifacts and to enhance feature regularities. The evaluation of this approach in comparison with other techniques is carried out both subjectively and qualitatively.**

Index Terms—**Adaptive filtering, deblocking, edge enhancement, image restoration, JPEG coding, statistical analysis.**

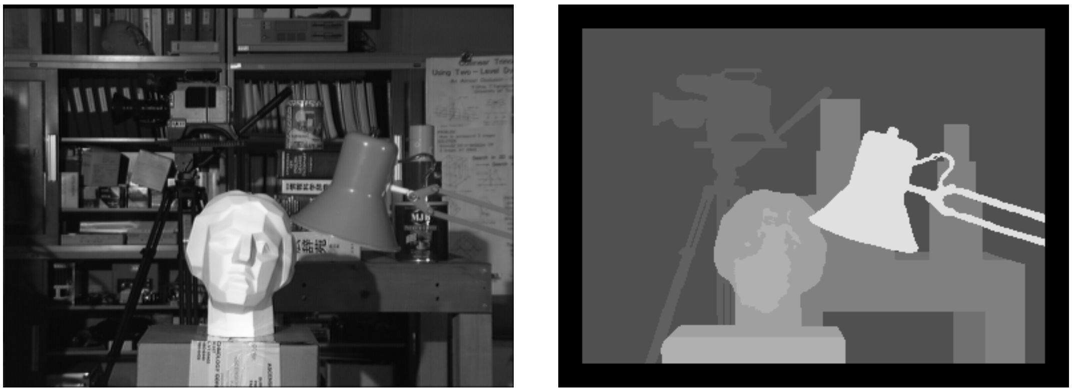
I. INTRODUCTION

THE discrete cosine transform (DCT) is the integrated part of the basic compression algorithm in JPEG. The importance of this compression technique can be utilized due to its performance that matches the Karhunen-Loeve Transform (KLT), which is known to be optimal in the mean square error sense. Although this is the most popular compression artifacts near the strong edges, and corruption of edg block boundaries [2], [9], [11], [16], [22], [23].

Due to the huge data requirements for multimedia, t tion is focused toward getting more compression anc sual defects. To remove the blocking effect, several de techniques have been proposed in the literature as pos mechanisms after JPEG compression, depending on spective from which the deblocking problem is dealt v easiest way of looking at this problem is to low-pass th JPEG image [1]. This approach will reduce the effect frequency tendency but the image will be blurry and s tails will be wiped out. Going further step in comple applying a simple nonlinear smoothing to the pixels another obstacle to the solution [2]. A more sophistic proach involves segmentation and smoothing that wil the ringing artifacts due to sharp variations [3]. Due to portance of high-frequency details that may be overv with blocking noise, another restoration process that us predictors to estimate the corresponding boundary pix regions of interest is devised [4]. The attempt in this n to classify small local boundary regions according to tensity distribution and to employ this information in d proper predictors. Despite the importance of this appr blocking effect at high compression rates not only is by block boundaries; moreover, images with sharp v cannot be easily configured with low order predictiv To improve the accuracy of the classified patterns an method for block removal using block classification an

Figure K.18: SCAN0006 (512 × 512).

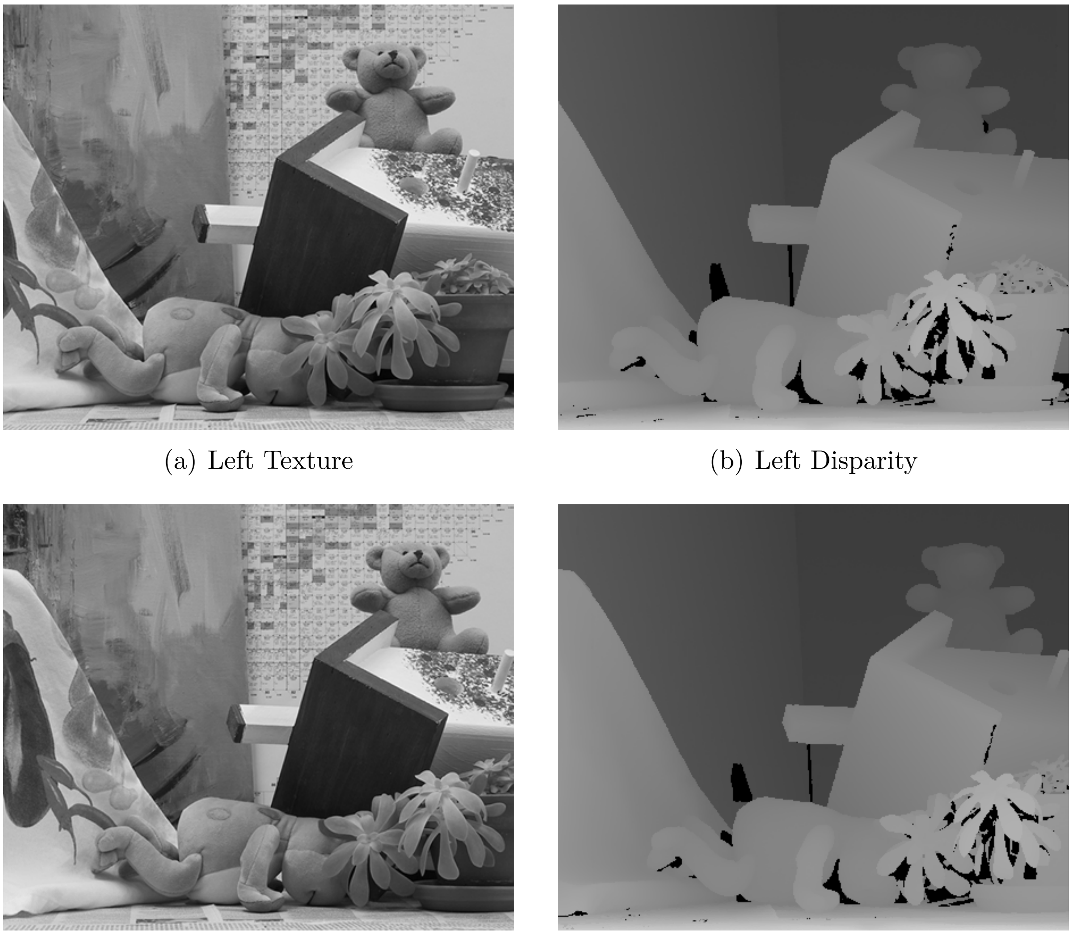
K.3 3D images



(a) Texture

(b) Disparity

Figure K.19: Tsukuba (384 × 288).



(a) Left Texture

(b) Left Disparity

(c) Right Texture

(d) Right Disparity

Figure K.20: Teddy (450 × 375).

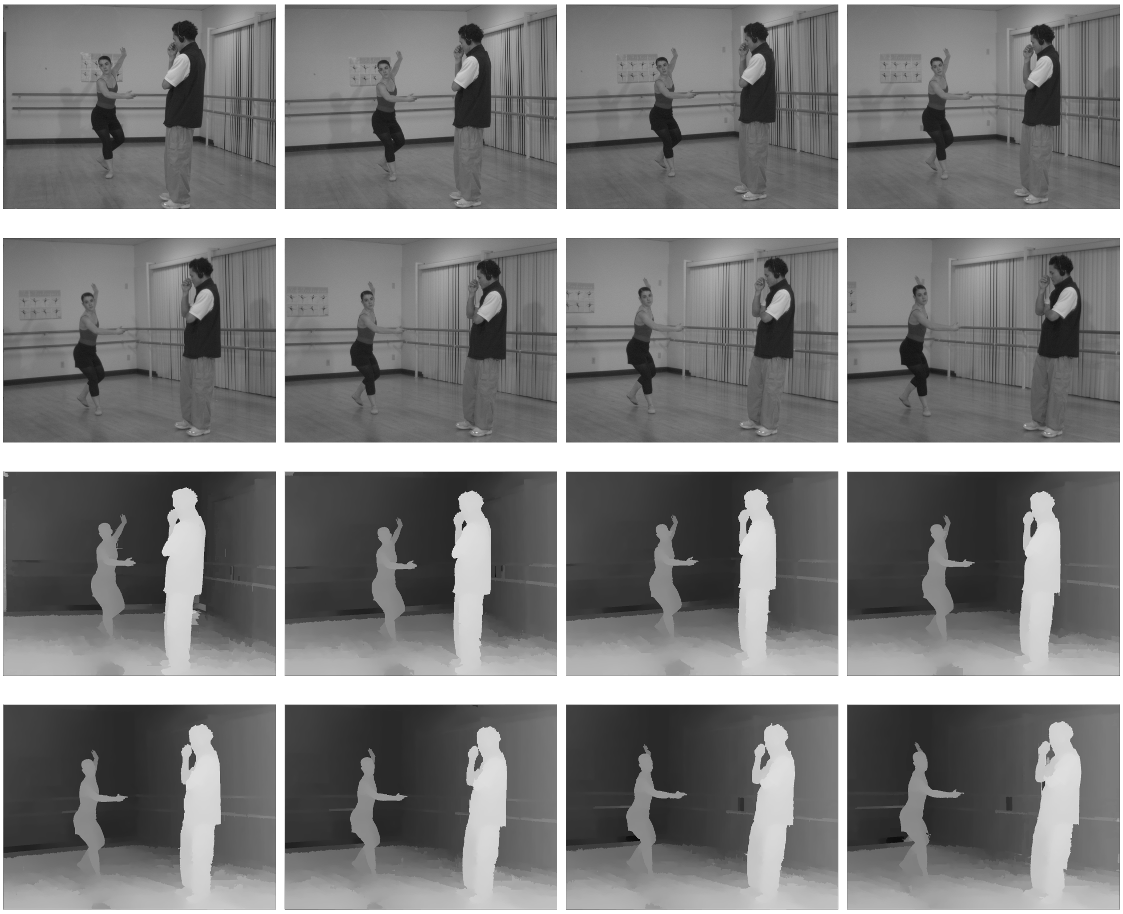


Figure K.21: First Frame of the Sequence Ballet (Luminance-only, 1024 × 768) and respective depth maps, from cameras 0 to 7, from left to right.

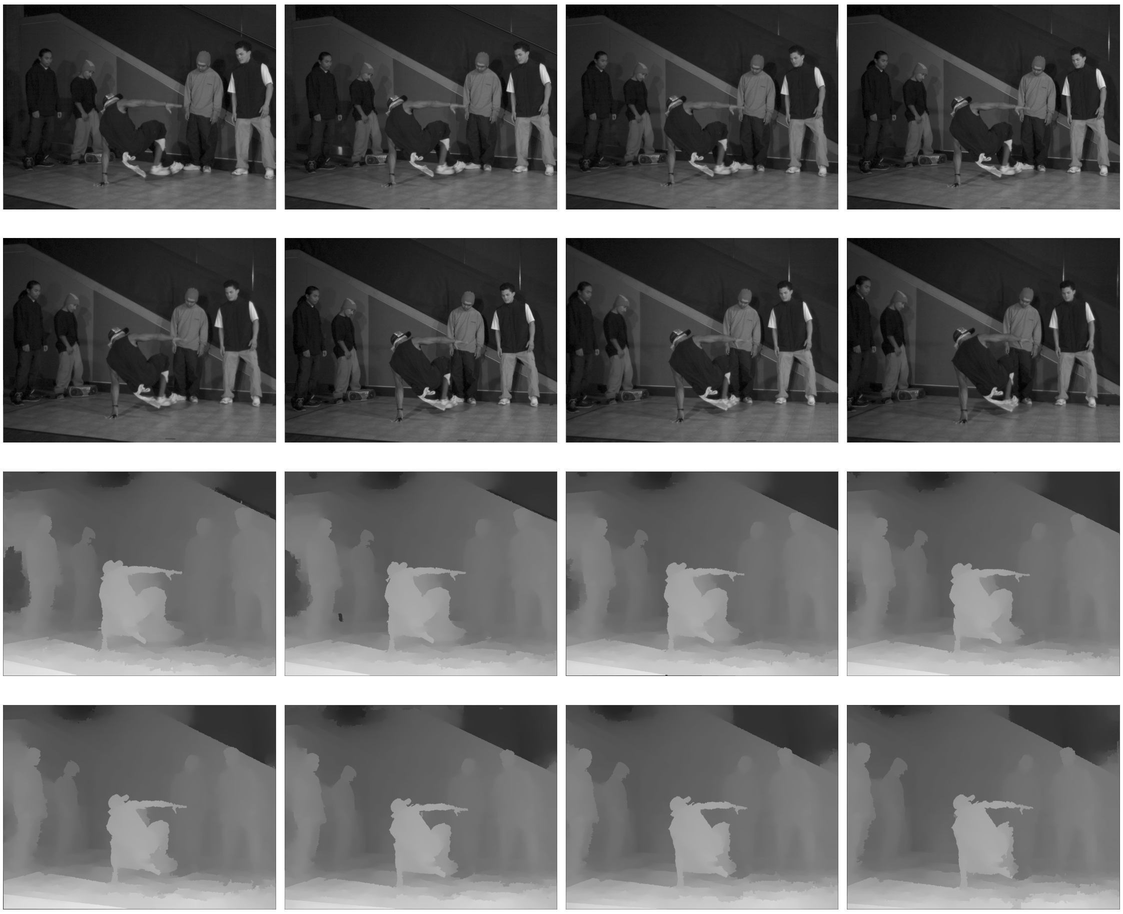
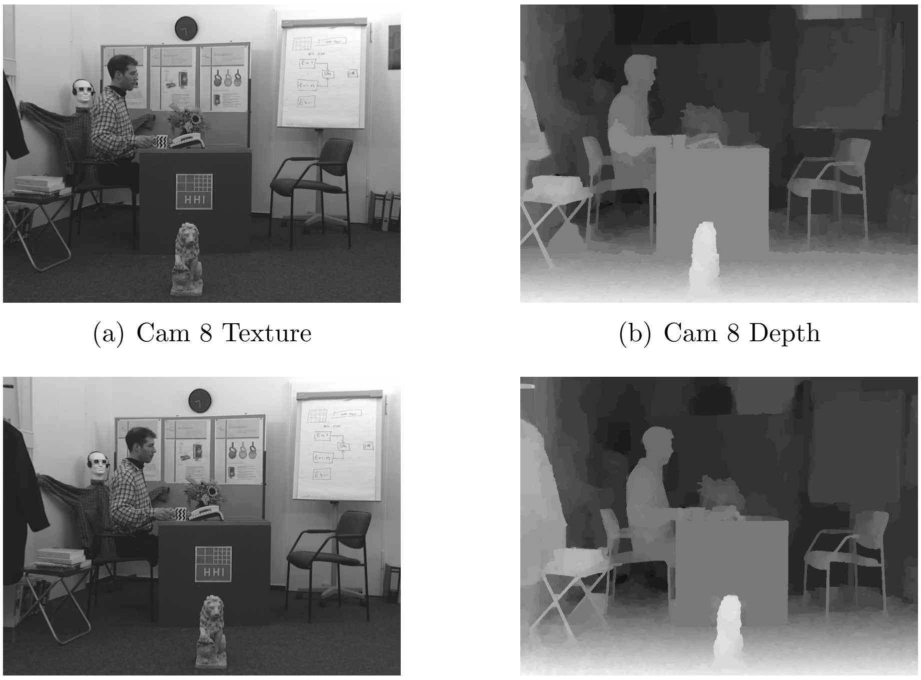


Figure K.22: First Frame of the Sequence Breakdancers (Luminance-only, 1024 × 768) and respective depth maps, from cameras 0 to 7, from left to right.



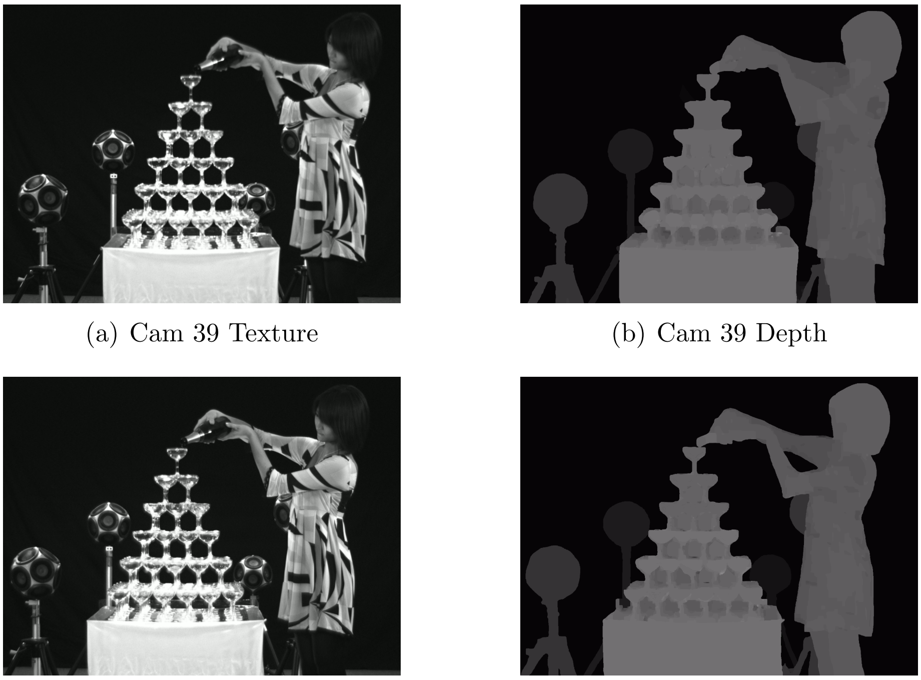
(a) Cam 8 Texture

(b) Cam 8 Depth

(c) Cam 10 Texture

(d) Cam 10 Depth

Figure K.23: First Frame of the Sequence Book Arrival (Luminance-only, 1024 × 768).



(a) Cam 39 Texture

(b) Cam 39 Depth

(c) Cam 41 Texture

(d) Cam 41 Depth

Figure K.24: First Frame of the Sequence Champagne Tower (Luminance-only, 1260 × 960).

Referências Bibliográficas

[1] SHANNON, C. E. "A mathematical theory of communication", *The Bell System Technical Journal*, v. 27, n. 3, pp. 379–423, 1948.

[2] JAIN, A. K. *Fundamentals of Digital Image Processing.* 1 ed. Upper Saddle River, NJ, Prentice-Hall, Inc., 1989.

[3] PENNEBAKER, W., MITCHEL, J. *JPEG: Still Image Data Compression Standard.* 1 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 1992.

[4] "ITU-T Recommendation T.82, Information technology - Coded representation of picture and audio information - Progressive bi-level image compression". March 1993. Disponível em: <http://www.itu.int/rec/T-REC-T.82>. Acesso em: 2 mar. 2011.

[5] "ISO/IEC JTC1/SC29/WG1 N1545, JBIG2 Final Draft International Standard". December 1999.

[6] TAUBMAN, D. S., MARCELIN, M. *JPEG2000: Image Compression Fundamentals, Standards and Practice.* 2 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 2001.

[7] "ITU-T & ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2), Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video". July 1995.

[8] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), fidelity range extensions documents JVT-L047 (non-integrated form) and JVT-L050 (integrated form)". July 2004.

[9] "ITU-T, ISO/IEC JTC 1, Advanced video coding for generic audio-visual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Version 1: May 2003, Version 2: Jan. 2004, Version 3: Sept. 2004, Version 4: July 2005." .

[10] UGUR, K., ANDERSSON, K., FULDSETH, A., et al. “High performance, low complexity video coding and the emerging HEVC standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. PP, n. 99, pp. 1, November 2010.

[11] “ISO/IEC JTC1/SC29/WG11, Draft Call for Proposal on 3D Video Coding Technology, Doc. N11679”. October 2010. Guangzhou, China.

[12] FISHER, Y. *Fractal Image Compression.* 1 ed. New York, NY, Springer Verlag, 1992.

[13] DE CARVALHO, M. B. *Compression of Multidimensional Signals based on Recurrent Multiscale Patterns.* Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2001.

[14] FILHO, E. B. L. *Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Critério de Continuidade Inter-blocos.* Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2004.

[15] RODRIGUES, N. M. M. *Multiscale Recurrent Pattern Matching Algorithms for Image and Video Coding.* Tese de Doutorado, Universidade de Coimbra, Coimbra, Portugal, 2008.

[16] FRANCISCO, N. C. *Estudo da Utilização de Recorrência de Padrões Multiescala na Codificação de Documentos Compostos.* Tese de Mestrado, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal, 2008.

[17] DE CARVALHO, M., LIMA, D. M., DA SILVA, E., et al. “Universal multi-scale matching pursuits algorithm with reduced blocking effect”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’00*, pp. 853 –856, Vancouver, BC, Canada, September 2000.

[18] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. “Improving multiscale recurrent pattern image coding with deblocking filtering”. In: *Proceedings of the IEEE International Conference on Signal Processing and Multimedia Applications, SIGMAP ’06*, pp. 118–125, Setúbal, Portugal, August 2006.

[19] DUARTE, M. H. V., DE CARVALHO, M. B., DA SILVA, E. A. B., et al. “Multiscale recurrent patterns applied to stereo image coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 11, n. 15, pp. 1434 – 1447, November 2005.

[20] FILHO, E. B. L., DA SILVA, E. A. B., JUNIOR, W., et al. “ECG compression using multiscale recurrent patterns with period normalization”. In: *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS ’06*, p. 4 pp., Island of Kos, Greece, May 2006.

[21] 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.

[22] 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, California, USA, October 2008.

[23] MARPE, D., WIEGAND, T., GORDON, S. “H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’05*, pp. 593–596, Genoa, Italy, September 2005.

[24] SCHARSTEIN, D., SZELISKI, R. “High-accuracy stereo depth maps using structured light”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR ’03*, pp. 195–202, June 2003. Madison, WI, USA.

[25] GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. “Fast implementation for multiscale recurrent pattern image coding”. In: *Proceedings of the Conference on Telecommunications - ConfTele ’09*, Santa Maria da Feira, Portugal, May 2009.

[26] GRAZIOSI, D., RODRIGUES, N., DA SILVA, E. A. B., et al. “Improving multiscale recurrent pattern image coding with least-squares prediction mode”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, pp. 2813 –2816, Cairo, Egypt, November 2009.

[27] GRAZIOSI, D., RODRIGUES, N., DA SILVA, E. A. B., et al. “Codificação de imagens com predição adaptativa baseada no critério de mínimos quadrados”. In: *Proceedings of the Simpósio Brasileiro das Telecomunicações, SBrT ’09*, Blumenau, SC, Brazil, September 2009.

[28] GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. “On the performance of lossless image compression using multiscale recurrent pattern matching”, *in process of submission*, 2011.

[29] GRAZIOSI, D., RODRIGUES, N., PAGLIARI, C., et al. "Compressing depth maps using multiscale recurrent pattern image coding", *Electronics Letters*, v. 46, n. 5, pp. 340–341, April 2010.

[30] GRAZIOSI, D., RODRIGUES, N., PAGLIARI, C., 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.

[31] ZIV, J., LEMPEL, A. "A universal algorithm for sequential data compression", *IEEE Transactions on Information Theory*, v. 23, n. 3, pp. 337–343, 1977.

[32] ZIV, J., LEMPEL, A. "Compression of individual sequences via variable-rate coding", *IEEE Transactions on Information Theory*, v. 24, n. 5, pp. 530–536, 1978.

[33] M. RODEH, V. PRATT, S. E. "Linear algorithm for data compression via string matching", *Journal of the ACM*, v. 28, n. 1, pp. 16–24, 1981.

[34] J. STORER, T. S. "Data compression via textual substitution", *Journal of the ACM*, v. 29, n. 4, pp. 928–951, 1982.

[35] BELL, T. "Better OPM/L text compression", *IEEE Transactions on Communications*, v. 34, n. 12, pp. 1176–1182, 1986.

[36] BRENT, R. "A linear algorithm for data compression", *Australian Computer Journal*, v. 19, n. 2, pp. 64–68, 1987.

[37] WELCH, T. A. "A Technique for high performance data compression", *IEEE Computer*, v. 17, n. 6, pp. 8–19, 1984.

[38] V. MILLER, M. W. "Variations on a scheme by Ziv and Lempel", *Combinatorial Algorithms on Words, NATO ASI Series*, v. F12, pp. 131–140, 1984.

[39] JAKOBSSON, M. "Compression of character strings by an adaptive dictionary", *BIT Numerical Mathematics*, v. 4, n. 25, pp. 593–603, 1985.

[40] TISCHER, P. "A modified Lempel-Ziv-Welch data compression scheme", *Australian Computer Science Communications*, v. 9, n. 1, pp. 262–272, 1987.

[41] E. FIALA, D. G. "Data compression with finite windows", *Communications of the ACM*, v. 32, n. 4, pp. 490–505, 1989.

[42] DE CARVALHO, M. B., FINAMORE, W. A. “Lossy Lempel-Ziv on subband coding of images”. In: *Proceedings of the IEEE International Symposium on Information Theory*, p. 415, Trondheim, Norway, June 1994.

[43] CHAN, C., VETTERLI, M. “Lossy compression of individual signals based on string matching and one pass codebook design”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’95*, p. 2491, Detroit, Michigan, USA, May 1995.

[44] DE CARVALHO, M. B., DA SILVA, E. A. B. “A universal multi-dimensional lossy compression algorithm”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’99*, pp. 767–771, Kobe, Japan, October 1999.

[45] 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.

[46] FILHO, E. B. L., RODRIGUES, N. M., DA SILVA, E. A. B., et al. “ECG signal compression based on DC equalization and complexity sorting”, *IEEE Transactions on Biomedical Engineering*, v. 55, n. 7, pp. 1923–1926, July 2008.

[47] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. “H.264/AVC based video coding using multiscale recurrent patterns: first results”. In: *Proceedings of the 9th International Workshop on Visual Content Processing and Representation, VLBV ’05*, v. 3893, pp. 107–114, Sardinia, Italy, September 2006.

[48] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. “An efficient H.264-based video encoder using multiscale recurrent patterns”, *SPIE - Applications of Digital Image Processing*, v. 6312, August 2006.

[49] 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.

[50] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. “On overriding H.264/AVC B-slice predicted residue coding”. In: *Pro-*

ceedings of the Picture Coding Symposium, PCS '07, Lisbon, Portugal, November 2007.

[51] "Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)". March 2005.

[52] 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.

[53] FILHO, E. B. L., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Universal image compression using multiscale recurrent patterns with adaptive probability model", *IEEE Transactions on Image Processing*, v. 17, n. 4, pp. 512–527, April 2008.

[54] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Universal image coding using multiscale recurrent patterns and prediction". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '05*, v. 2, pp. II–245–8, Genoa, Italy, September 2005.

[55] WEI-GUANG LIN, AN-CHAO TSAI, J.-F. W., YANG, J.-F. "A simple direction detection algorithm for fast H.264 intra prediction". In: *Proceedings of the 2007 IEEE Region 10 Conference, TENCON '07*, pp. 1–4, Taipei, Taiwan, October 2007.

[56] LI, X. *Edge Directed Statistical Inference and its Application to Image Processing.* Tese de Doutorado, Princeton University Graduate School, Princeton, New Jersey, USA, 2000.

[57] DINIZ, P. S. R. *Adaptive Filtering Algorithms and Practical Implementation.* 2 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 2002.

[58] ET AL, W. H. P. *Numerical Recipes in C: The Art of Scientific Computing.* 2 ed. New York, NY, Cambrigde University Press, 1992.

[59] LIU, L., LIU, Y. Z., DELP, E. J. "Enhanced intra prediction using context-adaptive linear prediction". In: *Proceedings of the Picture Coding Symposium, PCS '07*, Lisbon, Portugal, November 2007.

[60] LI, X., ORCHARD, M. T. "Edge-directed prediction for lossless compression of natural images", *IEEE Transactions on Image Processing*, v. 10, n. 6, pp. 813–817, 2001.

[61] COVER, T. M., THOMAS, J. A. *Elements of Information Theory.* 1 ed. New York, NY, Wiley-Interscience, 1991.

[62] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Basis optimisation for multiscale recurrent pattern coding". In: *Proceedings of the 3rd International Workshop on Mathematical Techniques and Problems in Telecommunications*, Leiria, Portugal, September 2006.

[63] LEE, Y. L., HAN, K.-H., SULLIVAN, G. J. "Improved lossless intra coding for H.264/MPEG-4 AVC." *IEEE Transactions on Image Processing*, v. 15, n. 9, pp. 2610–2615, 2006.

[64] SULLIVAN, G. J., YU, H., SEKIGUCHI, S., et al. "New standardized extensions of MPEG4-AVC/H.264 for professional-quality video applications". In: *Proceedings of the International Conference on Image Processing, ICIP '07*, pp. 13–16, San Antonio, Texas, USA, September 2007.

[65] WU, X., MEMON, N. "Context-based adaptive lossless image coding", *IEEE Transactions on Communications*, v. 45, n. 4, pp. 437–444, April 1997.

[66] 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, 2000.

[67] RICE, R. F. *Some practical universal noiseless coding techniques - parts I-III.* Relatório técnico, Jet Propulsion Laboratory, Pasadena, CA, 1983.

[68] VETRO, A. "Representation and Coding Formats for Stereo and Multiview Video". In: *Intelligent Multimedia Communication: Techniques and Applications*, v. 280, *Studies in Computational Intelligence*, Springer, pp. 51–73, 2010.

[69] "ISO/IEC JTC1/SC29/WG11, Vision on 3D Coding, Doc. N10357". February 2009. Lausanne, Switzerland.

[70] "ISO/IECJTC1/SC29/WG11,Text of ISO/IECFDIS23002-3 Representation of Auxiliary Video and Supplemental Information,Doc.N8768". January 2007. Marrakech, Morocco.

[71] "ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC 13818-1:2003/FDAM2 Carriage of Auxiliary Data, Doc. N8799". January 2007. Marrakech, Morocco.

[72] SMOLIC, A., MUELLER, K., STEFANOSKI, N., et al. "Coding Algorithms for 3DTV: A Survey", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 17, n. 11, pp. 1606–1621, November 2007.

[73] SCHARSTEIN, D., SZELISKI, R. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", *International Journal of Computer Vision*, v. 47(1/2/3), pp. 7–42, April-June 2002.

[74] KIM, S., HO, Y. "Mesh-Based Depth Coding for 3D Video using Hierarchical Decomposition of Depth Maps". In: *Proceedings of the International Conference on Image Processing, ICIP '07*, pp. 117–120, San Antonio, Texas,USA, September 2007.

[75] KIM, W.-S., ORTEGA, A., LAI, P., et al. "Depth map distortion analysis for view rendering and depth coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, pp. 721–724, November 2009.

[76] EKMEKCIOGLU, E., MRAK, M., WORRALL, S., et al. "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, pp. 733–736, Cairo, Egypt, November 2009.

[77] 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*, pp. 828–831, Thessaloniki, Greece, October 2001.

[78] MERKLE, P., MORVAN, Y., SMOLIC, A., et al. "The effects of multiview depth video compression on multiview rendering", *Signal Processing: Image Communication*, v. 24, n. 1-2, pp. 73–88, 2009.

[79] FARIN, D., PEERLINGS, R., DE WITH, P. "Depth-Image Representation Employing Meshes for Intermediate-View Rendering and Coding". In: *Proceedings of the 3DTV Conference*, pp. 1–4, Kos Island, Greece, May 2007.

[80] SARKIS, M., ZIA, W., DIEPOLD, K. "Fast Depth Map Compression and Meshing with Compressed Tritree". In: *Proceedings of the Ninth Asian Conference on Computer Vision, ACCV '09*, Xi'an, China, September 2009.

[81] SARKIS, M., DIEPOLD, K. "Depth map compression via compressed sensing". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '09*, November 2009.

[82] MULLER, K., MERKLE, P., WIEGAND, T. "3-D Video Representation Using Depth Maps", *Proceedings of the IEEE*, v. PP, n. 99, pp. 1–14, 2010.

[83] WANG, Z., BOVIK, A. C., SHEIKH, H. R., et al. "Image quality assessment: from error visibility to structural similarity." *IEEE Transactions on Image Processing*, v. 13, n. 4, pp. 600–612, 2004.

[84] YEA, S., VETRO, A. "Multi-layered coding of depth for virtual view synthesis". In: *Proceedings of the Picture Coding Symposium, PCS '09*, pp. 1–4, Chicago, Illinois, USA, May 2009.

[85] SANCHEZ, A., SHEN, G., ORTEGA, A. "Edge-preserving depth-map coding using graph-based wavelets". In: *Proceedings of the Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 578–582, Pacific Grove, CA, USA, November 2009.

[86] YEA, S., VETRO, A. "View synthesis prediction for multiview video coding", *Signal Processing: Image Communication*, v. 24, pp. 89–100, January 2009.

[87] KITAHARA, M., KIMATA, H., SHIMIZU, S., et al. "Multi-View Video Coding using View Interpolation and Reference Picture Selection". In: *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '06*, pp. 97–100, Toronto, Ontario, Canada, July 2006.

[88] GRAZIOSI, D. B., RODRIGUES, N. M. M., DA SILVA, E. A. B., et al. "Joint coding of texture and depth using multiscale recurrent pattern matching", *in process of submission*, 2011.

[89] GERSHO, A., GRAY, R. *Vector Quantization and Signal Compression.* 9 ed. Norwell, Massachusetts, Kluwer Academic Publishers, 1992.

[90] DE CARVALHO, M., DA SILVA, E., FINAMORE, W. "Multidimensional signal compression using multiscale recurrent patterns", *Elsevier Signal Processing*, v. 82, pp. 1559–1580, November 2002.

[91] SULLIVAN, G., BAKER, R. "Rate-distortion optimization for tree-structured source coding with multi-way node decisions". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '92*, pp. 393–396, Los Alamitos, CA, USA, 1992.

[92] PINAGÉ, F. S. *Avaliação do Desempenho de Algoritmos de Compressão de Imagens Usando Recorrência de Padrões Multiescalas.* Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2005.

[93] FILHO, E. B. L., DE CARVALHO, M., DA SILVA, E. A. B. "Multidimensional signal compression using multi-scale recurrent patterns with smooth side-match criterion". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '04*, v. 5, pp. 3201 – 3204, Singapore, Singapore, October 2004.

[94] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Image predictive coding using multiscale recurrent patterns". In: *Proceedings of the Conference on Telecommunications, ConfTele '05*, April 2005.

[95] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Efficient dictionary design for multiscale recurrent patterns image coding". In: *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '06*, p. 4 pp., Island of Kos, Greece, May 2006.

[96] FISCHER, T. "A pyramid vector quantizer", *IEEE Transactions on Information Theory*, v. IT-32, n. 4, pp. 568–583, July 1986.

[97] CHEN, F., GAO, Z., VILLASENOR, J. "Lattice vector quantization of generalized gaussian sources", *IEEE Transactions on Information Theory*, v. 43, n. 1, pp. 92–103, January 1997.

[98] PAN, Z., KOTANI, K., OHMI, T. "Improved fast search method for vector quantization using discrete Walsh transform". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '04*, v. 5, pp. 3177–3180, Singapore, Singapore, October 2004.

[99] WOODS, J. "Two-dimensional discrete Markovian fields", *IEEE Transactions on Information Theory*, v. 18, n. 2, pp. 232–240, March 1972.

[100] BOLLE, R. M., COOPER, D. B. "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-6, n. 4, pp. 418 –429, July 1984.

[101] HAVLICEK, J. P., HAVLICEK, J. W., MAMUYA, N. D., et al. "Skewed 2D Hilbert transforms and computed AM-FM models". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '98*, pp. 602–606, Chicago, Illinois, USA, October 1998.

[102] 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.

[103] WOODS, J. "Two-dimensional Kalman filtering". In: Huang, T. (Ed.), *Two-Dimensional Digital Signal Processing I*, v. 42, *Topics in Applied Physics*, Springer Berlin / Heidelberg, pp. 155–205, 1981.

[104] LI, X. "Least-square prediction for backward adaptive video coding", *EURASIP Journal on Applied Signal Processing*, v. 2006, n. 1, pp. 126–126, 2006.

[105] 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.

[106] CAMPBELL, S. L., MEYER, C. D. *Generalized Inverses of Linear Transformations.* 1 ed. Philadelphia, PA, Society for Industrial and Applied Mathematics, 2008.

[107] MITRA, S. *Digital Signal Processing.* 3 ed. New York, NY, McGraw-Hill Science/Engineering/Math, 2005.

[108] SAYOOD, K. *Introduction to Data Compression.* 2 ed. San Francisco, CA, Morgan Kaufmann Publishers Inc., 2000.

[109] WU, X. "An algorithmic study on lossless image compression". In: *Proceedings of the 1996 Data Compression Conference, DCC '96*, pp. 150–159, Snowbird, Utah, USA, 1996. IEEE Computer Society Press.

[110] WEINBERGER, M. J., RISSANEN, J., ARPS, R. "Applications of universal context modeling to lossless compression of gray-scale images." *IEEE Transactions on Image Processing*, v. 5, n. 4, pp. 575–586, 1996.

[111] MATSUDA, I., KANEKO, T., MINEZAWA, A., et al. "Lossless coding of color images using block-adaptive inter-color prediction". In: *Proceedings of the International Conference on Image Processing, ICIP '07*, pp. 329–332, San Antonio, Texas, USA, September 2007.

[112] ATALLAH, M., GENIN, Y., SZPANKOWSKI, W., et al. "Pattern matching image compression: algorithmic and empirical results", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, pp. 21–618, July 1999.

[113] YANG, E.-H., KIEFFER, J. C. "Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform - Part one: Without context models", *IEEE Transactions on Information Theory*, v. 46, n. 3, pp. 755–777, 2000.

[114] RANDERS-PEHRSON, G., ADLER, M., BOUTELL, T., et al. "PNG (Portable Network Graphics) Specification, Version 1.2". 1999. Disponível em: <`http://www.libpng.org/pub/png/spec/`>. Acesso em: 14 mar. 2011.

[115] BARNI, M. *Document and Image Compression.* 1 ed. Sound Parkway, NW, CRC Press, Inc, 2006.

[116] CARPENTIERI, B., WEINBERGER, M., SEROUSSI, G. "Lossless compression of continuous-tone image", *Proceedings of the IEEE*, v. 88, n. 11, pp. 1797–1809, November 2000.

[117] GRAZIOSI, D. B. *Survey on Lossless Coding Techniques.* Relatório técnico, Laboratório de Processamento de Sinais, 2011. Disponível em: <`http://www.lps.ufrj.br/profs/eduardo/mmp/Graziosi_Lossless_Survey.pdf`>. Acesso em: 02 mar. 2011.

[118] WU, X., MEMON, N., SAYOOD, K. "A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images". In: *ISO/IEC JTC 1/SC 29/WC 1 document No*, 1995.

[119] MEYER, B., TISCHER, P. "TMW - a new method for lossless image compression". In: *Proceedings of the Picture Coding Symposium, PCS '97*, pp. 533–8, Lisbon, Portugal, 1997.

[120] MEYER, B., TISCHER, P. "Glicbawls - grey level image compression by adaptive weighted least squares". In: *Proceedings of the Conference on Data Compression, DCC '01*, p. 503, Snowbird, Utah, USA, March 2001.

[121] MATSUDA, I., OZAKI, N., UMEZU, Y., et al. "Lossless coding using variable block-size adaptive prediction optimized for each image". In: *Proceedings of European Signal Processing Conference, EUSIPCO '05*, Antalya, Turkey, September 2005.

[122] MARTIN, G. N. N. "Range encoding: an algorithm for removing redundancy from a digitized message". In: *Proceedings of the Video and Data Recording Conference*, Southampton, UK, March 1979.

[123] SUBBOTIN, D. "Carryless Rangecoder, 1999". 2000. Disponível em: <http://search.cpan.org/src/SALVA/Compress-PPMd-0.10/Coder.hpp>. Acesso em: 14 mar. 2011.

[124] DEVROYE, L. *Non-Uniform Random Variate Generation.* 1 ed. New York, NY, Springer-Verlag, 1986.

[125] "GRAPHICS INTERCHANGE FORMAT(sm) - Version 89a". 1987,1988,1989,1990. Disponível em: <http://www.gif.org>. Acesso em: 14 mar. 2011.

[126] SCHALNAT, E. G., DILGER, A., RANDERS-PEHRSON, G. "PNG Reference Library (libpng)". 2011. Disponível em: <http://www.libpng.org/pub/png/libpng.html>. Acesso em: 14 mar. 2011.

[127] YE, Y., COSMAN, P. "Dictionary design for text image compression with JBIG2", *IEEE Transansaction on Image Processing*, v. 10, n. 6, pp. 818–828, 2001.

[128] ONO, F., RUCKLIDGE, W., ARPS, R., et al. "JBIG2 - The ultimate bi-level image coding standard". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '00*, pp. 140–143, Vancouver, BC, Canada, 2000.

[129] HOWARD, P., KOSSENTINI, F., MARTINS, B., et al. "The emerging JBIG2 standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 8, n. 7, pp. 838–848, November 1998.

[130] DEUTSCH, P. "DEFLATE compressed data format specification version 1.3". 1996. Disponível em: <http://tools.ietf.org/html/rfc1951>. Acesso em: 14 mar. 2011.

[131] BLACK DUCK SOFTWARE, I. "jj2000". 2011. Disponível em: <http://www.ohloh.net/p/jj2000>. Acesso em: 20 fev. 2011.

[132] WEINBERGER, M. J., SEROUSSI, G., SAPIRO, G. "JPEG-LS". 2011. Disponível em: <http://www.hpl.hp.com/loco>. Acesso em: 20 fev. 2011.

[133] WU, X. "CALIC". 2011. Disponível em: <ftp://ftp.csd.uwo.ca/pub/from_wu/v.huff/>. Acesso em: 20 fev. 2011.

[134] MEYER, B. "The Glicbawls page". 2011. Disponível em: <http://web.archive.org/web/20030208074120/http://byron.csse.monash.edu.au/glicbawls/>. Acesso em: 20 fev. 2011.

[135] LI, X. "EDP software code". 2011. Disponível em: <http://www.csee.wvu.edu/$\sim$xinl/code/edp.zip>. Acesso em: 20 fev. 2011.

[136] MATSUDA, I. "mrp Version 0.5". 2011. Disponível em: <http://itohws03.ee.noda.sut.ac.jp/~matsuda/mrp/mrp-05.tar.gz% textgreater . Acesso em: 20 fev. 2011.

[137] KUHN, M. "JBIG-KIT". 2011. Disponível em: <http://www.cl.cam.ac.uk/$\sim$mgk25/download/jbigkit-2.0.tar.gz>. Acesso em: 20 fev. 2011.

[138] LLC, I. S. "ImageMagick Convert Command-line Tool". 2011. Disponível em: <http://www.imagemagick.org/script/convert.php>. Acesso em: 20 fev. 2011.

[139] KIEFFER, J. C., YANG, E.-H. "Grammar-based codes: a new class of universal lossless source codes", *IEEE Transactions on Information Theory*, v. 46, n. 3, pp. 737–754, 2000.

[140] MEMON, N., SIPPY, V., WU, X. "A comparison of prediction schemes proposed for a new lossless image compression standard". In: *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '96*, v. 2, pp. 309–312, Philadelphia, PA, USA, May 1996.

[141] LI, X., ORCHARD, M. T. "Edge directed prediction for lossless compression of natural images". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '99*, pp. 58–62, Kobe, Japan, October 1999.

[142] LEHMANN, A., VAN SCHAIK, W., ROELOFS, G. "pnmtopng". 2011. Disponível em: <http://www.libpng.org/pub/png/apps/pnmtopng.html>. Acesso em: 20 fev. 2011.

[143] RANDERS-PEHRSON, G. "pngcrush". 2011. Disponível em: <http://pmt.sourceforge.net/pngcrush/>. Acesso em: 20 fev. 2011.

[144] DA SILVA, E. A. B. "MMP Project". 2011. Disponível em: <http://www.lps.ufrj.br/profs/eduardo/mmp/>. Acesso em: 22 fev. 2011.

[145] 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*, pp. 262–273, November 2005.

[146] TANIMOTO, M., TEHRANI, M., FUJII, T., et al. "Free-Viewpoint TV", *IEEE Signal Processing Magazine*, v. 28, n. 1, pp. 67–76, January 2011.

[147] BERENT, J., DRAGOTTI, P. “Plenoptic manifolds”, *IEEE Signal Processing Magazine*, v. 24, n. 6, pp. 34–44, November 2007.

[148] LEVOY, M., HANRAHAN, P. “Light Field Rendering”. In: *Proceedings of the 23rd Annual Conference on Computer Graphics*, pp. 31–42, New Orleans, LA, USA, August 1996.

[149] 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.

[150] 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.

[151] “ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC 14496-10:2009/FDAM 1 Constrained baseline profile, stereo high profile, and frame packing arrangement SEI message, Doc. N10707”. July 2009. London, UK.

[152] MERKLE, P., SMOLIC, A., MULLER, 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.

[153] SMOLIC, A., MULLER, 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, California, USA, October 2008.

[154] TANIMOTO, M. “FTV (Free-viewpoint TV)”. In: *Proceedings of the International Conference on Image Processing, ICIP ’10*, pp. 2393–2396, Hong Kong, China, September 2010.

[155] BRULS, F., GUNNEWIEK, R. K. “Options for a new efficient, compatible, flexible 3D standard”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, Cairo, Egypt, November 2009.

[156] MULLER, K., SMOLIC, A., DIX, K., et al. “Reliability-based generation and view synthesis in layered depth video”. In: *Proceedings of the IEEE 10th Workshop on Multimedia Signal Processing, MMSP ’08*, pp. 34–39, October 2008.

[157] SMOLIC, A., MUELLER, 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, Illinois, USA, May 2009.

[158] “ISO/IEC JTC1/SC29/WG11, HHI Test Material for 3D Video, Doc. M15413”. April 2008. Archamps, France.

[159] “ISO/IEC JTC1/SC29/WG11, 1D Parallel Test Sequences for MPEG-FTV , Doc. M15378”. April 2008. Archamps, France.

[160] 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, Aug 2004. Los Angeles, CA.

[161] 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.

[162] MORVAN, Y. *Acquisition, compression and rendering of depth and texture for multi-view video.* Tese de Doutorado, Eindhoven University of Technology, Eindhoven, The Netherlands, 2009.

[163] ZHANG, Z. *A Flexible New Technique for Camera Calibration.* Relatório técnico, Microsoft Research, December 1998.

[164] FLIERL, M., GIROD, B. “Multiview video compression”, *IEEE Signal Processing Magazine*, v. 24, n. 6, pp. 66–76, nov. 2007.

[165] YEA, S., VETRO, A. “Multi-layered coding of depth for virtual view synthesis”. In: *Proceedings of the Picture Coding Symposium, PCS ’09*, pp. 1–4, May 2009.

[166] KIM, W.-S., ORTEGA, A., LAI, P., et al. “Depth map coding with distortion estimation of rendered view”. In: *Proceedings of the Visual Information Processing and Communication*, v. 7543, San Jose, California, United States, January 2010.

[167] SILVA, D. V. S. X. D., FERNANDO, W. A. C., ARACHCHI, H. K. “A new mode selection technique for coding Depth maps of 3D video”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’10*, pp. 686–689, Dallas, Texas, USA, March 2010.

[168] NGUYEN, Q. H., DO, M. N., PATEL, S. J. “Depth image-based rendering with low resolution depth”. In: *Proceedings of the IEEE International Conference on Image Processing, ICIP ’09*, November 2009.

[169] OH, K.-J., YEA, S., VETRO, A., et al. “Depth Reconstruction Filter for Depth Coding”, *Electronics Letter*, v. 45, n. 6, pp. 305–306, March 2009.

[170] OH, K.-J., YEA, S., VETRO, A., et al. “Depth Reconstruction Filter and Down/Up Sampling for Depth Coding in 3-D Video”, *IEEE Signal Processing Letters*, v. 16, n. Issue 9, pp. 747–750, September 2009.

[171] ZAMARIN, M., MILANI, S., ZANUTTIGH, P., et al. “A novel multi-view image coding scheme based on view-warping and 3D-DCT”, *Journal of Visual Communication and Image Representation*, v. 21, n. 8, pp. 462–473, 2010.

[172] VETRO, A., WANG-HE, L., FLYNN, M. “TV Architecture Supporting Multiple 3D Services”. In: *Proceedings of the IEEE International Conference on Consumer Electronics, ICCE ’10*, pp. 1135–136, Shanghai, China, January 2010.

[173] KONRAD, J., HALLE, M. “3-D Displays and Signal Processing”, *IEEE Signal Processing Magazine*, v. 24, n. 6, pp. 97–111, November 2007.

[174] VETRO, A., MATUSIK, W., PFISTER, H., et al. “Coding Approaches for End-to-End 3D TV Systems”. In: *Proceedings of the Picture Coding Symposium, PCS ’04*, San Francisco, CA, USA, December 2004.

[175] “ISO/IEC JTC1/SC29/WG11, 3D TV System , Doc. m10624”. March 2004. Munich, Germany.

[176] MOLLER, C., TRAVIS, A. “Correcting interperspective aliasing in autostereoscopic displays”, *IEEE Transactions on Visualization and Computer Graphics*, v. 11, n. 2, pp. 228–236, March 2005.

[177] ZINGER, S., DO, L., DE WITH, P. “Free-viewpoint depth image based rendering”, *Journal of Visual Communication and Image Representation*, v. 21, n. 8, pp. 533–541, 2009.

[178] “ISO/IEC JTC1/SC29/WG11, Report on Experimental Framework for 3D Video Coding, Doc. N11631”. October 2010. Guangzhou, China.

[179] MORI, Y., FUKUSHIMA, N., YENDO, T., et al. “View generation with 3D warping using depth information for FTV”, *Signal Processing: Image Communication*, v. 24, n. 8, pp. 65–72, 2009.

[180] DOMANSKI, M., GOTFRYD, M., WEGNER, P. “View Synthesis for Multi-view Video Transmission”. In: *Proceedings of the 2009 International Conference on Image Processing, Computer Vision, & Pattern Recognition, IPCV ’09*, pp. 433–439, Las Vegas, Nevada, USA, July 2009.

[181] TELEA, A. “An Image Inpainting Technique Based on the Fast Marching Method”, *Journal of Graphics, GPU, and Game Tools*, v. 9, n. 1, pp. 23–34, 2004.

[182] LEE, C., HO, Y. “View synthesis using depth map for 3D video”. In: Sapporo, J. (Ed.), *Proceedings of the Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, APSIPA ’09*, October 2009.

[183] FU, D., ZHAO, Y., YU, L. “Temporal Consistency Enhancement on Depth Sequences”. In: *Proceedings of the Picture Coding Symposium, PCS ’10*, pp. 342–345, Nagoya, Japan, December 2010.

[184] ZHAO, Y., CHEN, Z., TIAN, D., et al. “Suppressing Texture-Depth MIs-alignment for boundary noise removal in view synthesis”. In: *Proceedings of the Picture Coding Symposium, PCS ’10*, pp. 30–33, Nagoya, Japan, December 2010.

[185] DO, L., ZINGER, S., DE WITH, P. H. N. “Objective quality analysis for free-viewpoint DIBR”. In: *Proceedings of the International Conference on Image Processing, ICIP ’10*, pp. 2629–2632, Hong Kong, China, September 2010.

[186] OH, K.-J., YEA, S., HO, Y.-S. “Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3-D video”. In: *Proceedings of the Picture Coding Symposium, PCS ’09*, pp. 1–4, Chicago, Illinois, USA, May 2009.

[187] “ISO/IEC JTC1/SC29/WG11, Applications and Requirements on 3D Video Coding Technology, Doc. N11678”. October 2010.

[188] “ISO/IEC JTC1/SC29/WG11, Description of Exploration Experiments in 3D Video Coding, Doc. N11630”. October 2010. Guangzhou, China.

[189] SUHRING, K. “H.264/AVC Reference Software”. 2011. Disponível em: <`http://iphome.hhi.de/suehring/tml/download/`>. Acesso em: 23 fev. 2011.

[190] ASPERT, N., SANTA-CRUZ, D., EBRAHIMI, T. "MESH: Measuring Errors between Surfaces using the Hausdorff Distance". In: *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '02*, v. I, pp. 705–708, Lusanne, Switzerland, August 2002.

[191] "ISO/IEC JTC1/SC29/WG11, 3DV/FTV EE1 and EE4 report on Champagne Tower sequence, Doc. M18354". October 2010. Guangzhou, China.

[192] "ISO/IEC JTC1/SC29/WG11, 3DV EE4 Report on Book_arrival Sequence, Doc. M18342". October 2010. Guangzhou, China.

[193] "ISO/IEC JTC1/SC29/WG11, Preliminary FTV Model and Requirements, Doc. N9168". July 2007. Lausanne, Switzerland.

[194] MERKLE, P., SMOLIC, A., MULLER, K., et al. "Multi-View Video Plus Depth Representation and Coding". In: *Proceedings of the IEEE International Conference on Image Processing, ICIP '07*, pp. 201–204, San Antonio, Texas, USA, October 2007.

[195] DE OLIVEIRA, A. V. C. *Codificação de Textura e Profundidade de Imagens 3DUtilizando Recorrência de Padrões Multiescalas.* Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2011.

[196] MORVAN, Y., FARIN, D., DE WITH, P. H. N. "Joint depth/texture bit-allocation for multi-view video compression". In: *Proceedings of the Picture Coding Symposium, PCS '07*, Lisbon, Portugal, November 2007.

[197] BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., et al. "PatchMatch: a randomized correspondence algorithm for structural image editing". In: *Proceedings of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '09*, pp. 24:1–24:11, New Orleans, Louisiana, USA, August 2009.